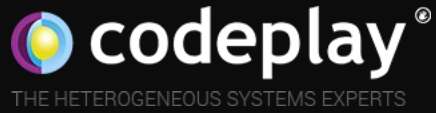


Why advances in Automotive Driving Aids Systems (ADAS) + efficient C++ + meet safety concerns

Illya Rudkin Principal Engineer Safety Critical Software

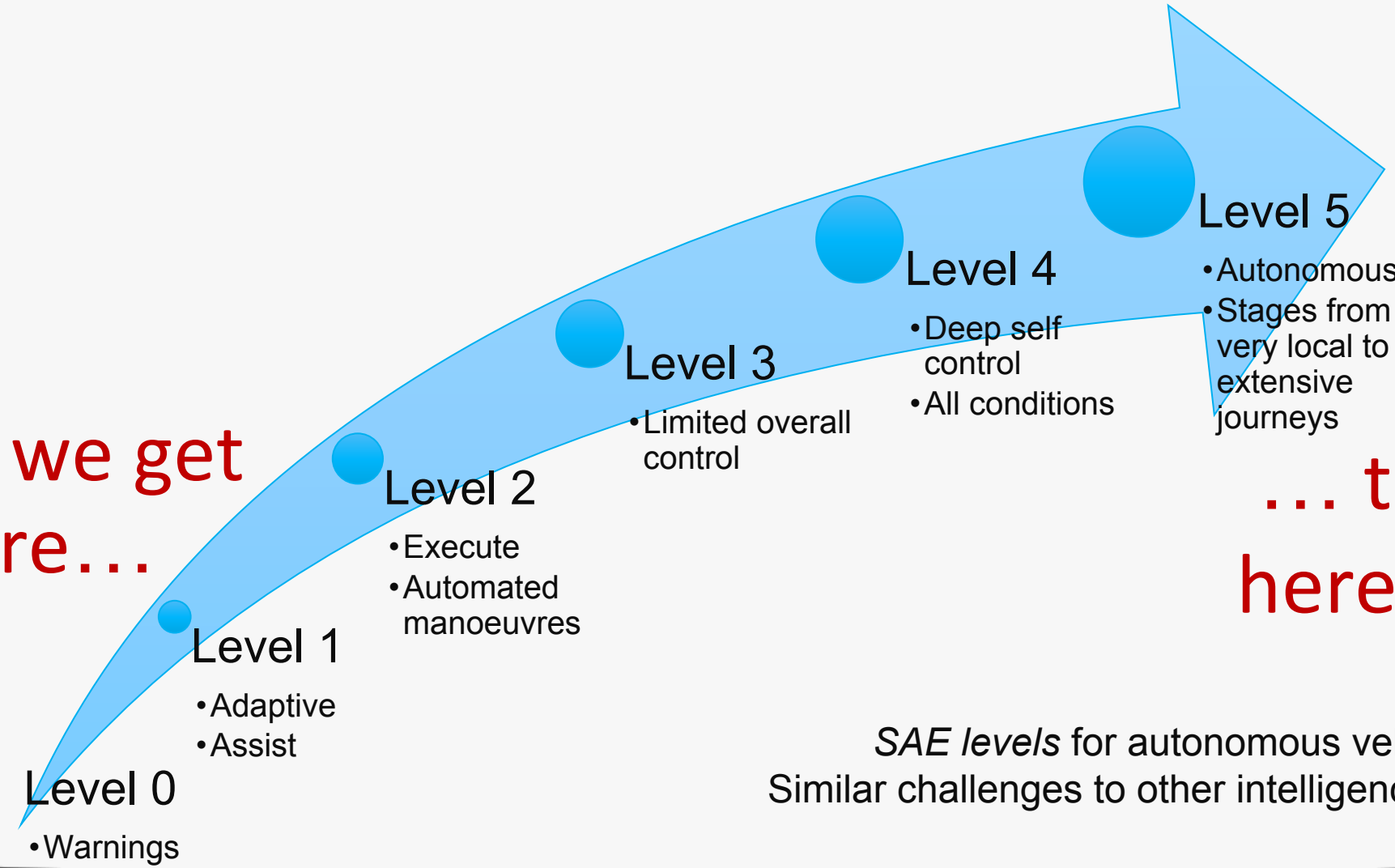
Khronos Safety Critical Advisory Forum chairman

Khronos OpenCL Safety Critical WG

MISRA C++ WG

Or a whistle stop tour of safety critical development

Levels of vehicles' aids & autonomy

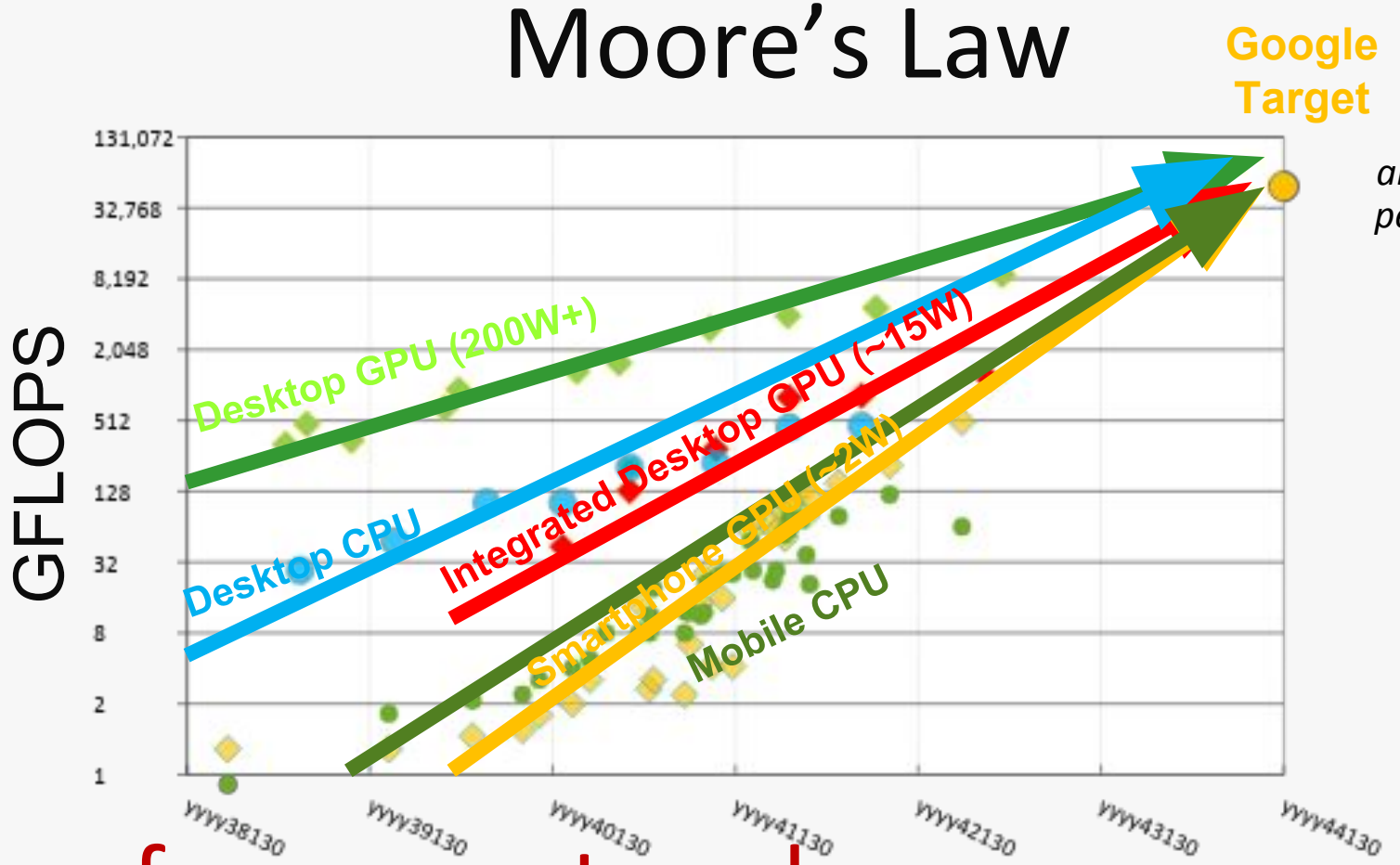


How do we get from here…

… to here?

SAE levels for autonomous vehicles.
Similar challenges to other intelligence industries

Moore's Law

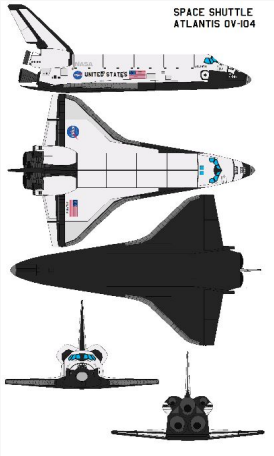


GPUs or similar architectures give the performance required for AI

Not CPUs

…performance trends are increasing…

Software in a car is now the major component



Space Shuttle
~500K lines of code

Boeing 777
~3M lines of code

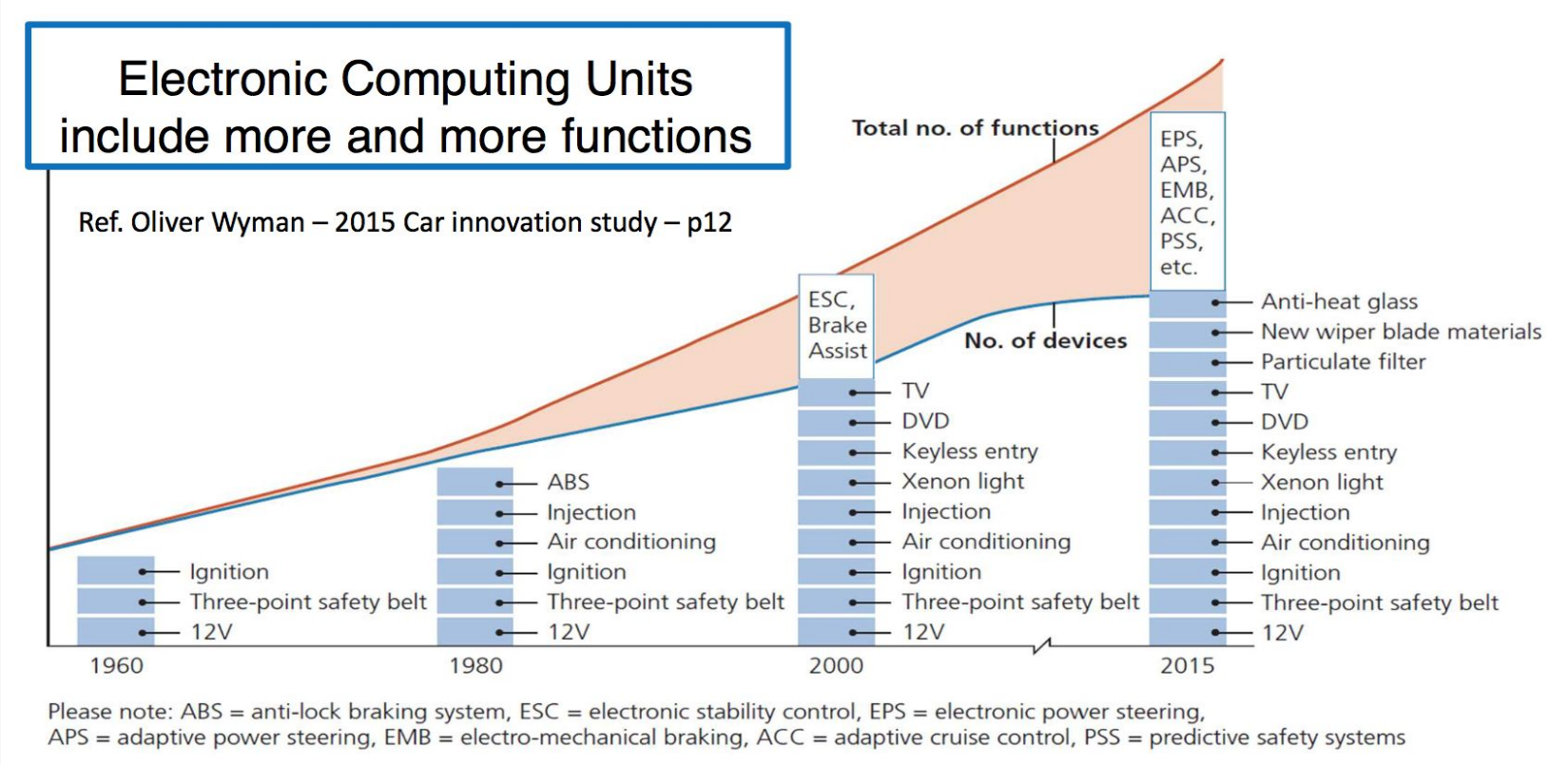
Modern Automobile
~100M lines of code
Up to 100 ECUs

And growing…..

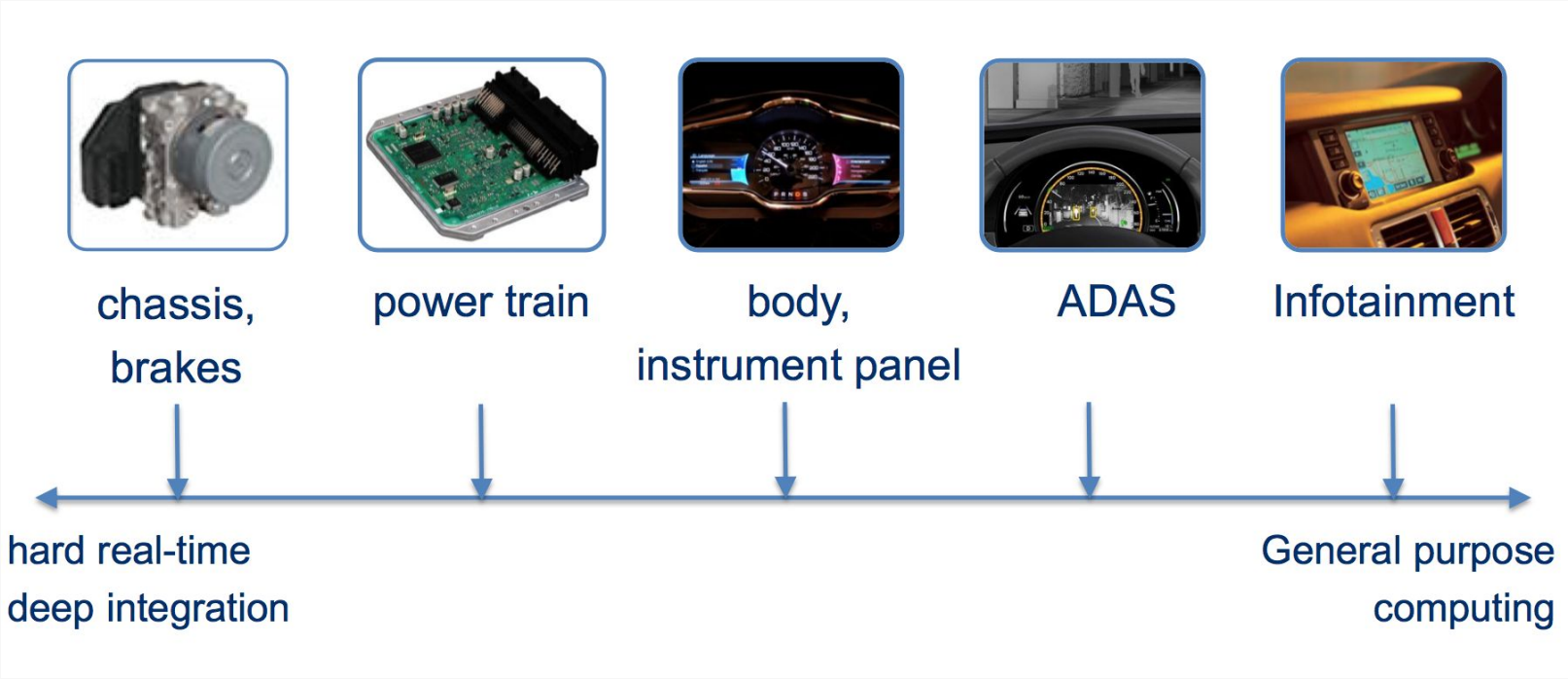
But reduce ECUs!

And power requirements

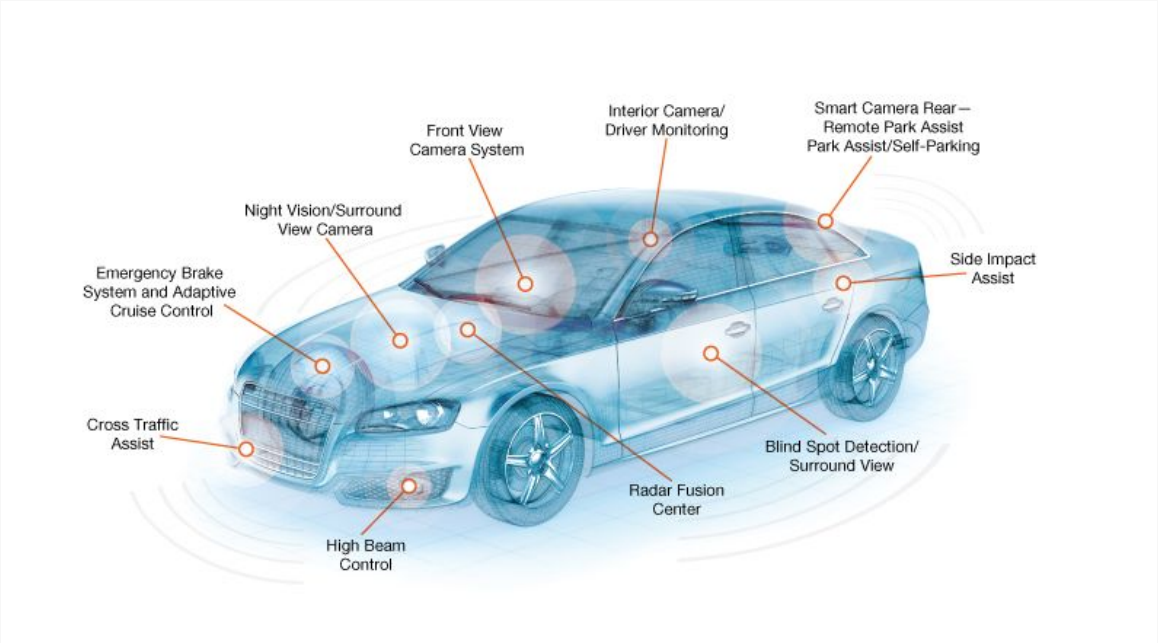
The reason you car is getting bloated



Critical and less critical vehicle items



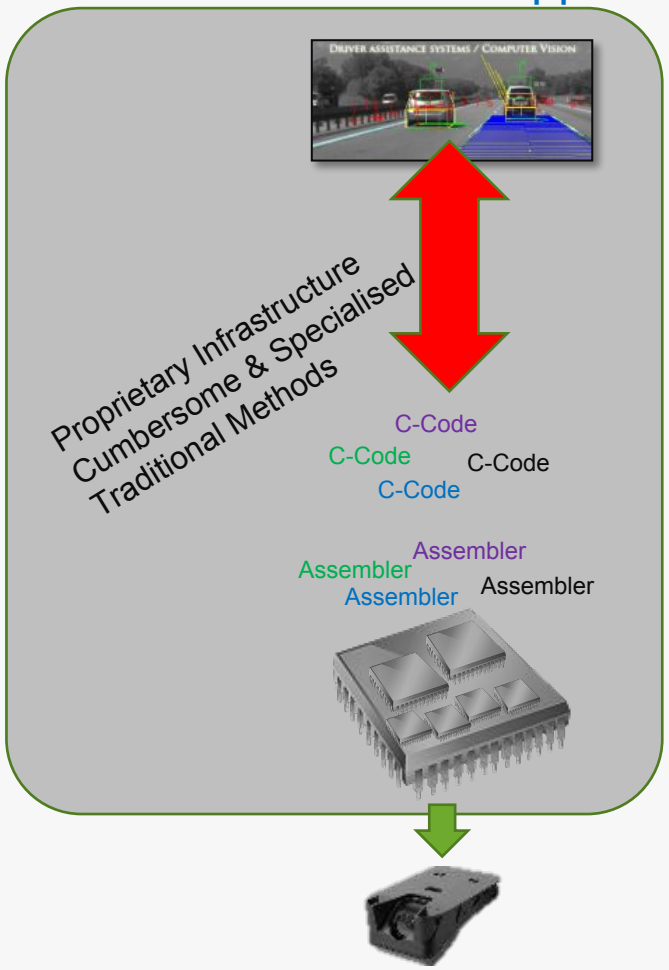
Scale Down – Affordable and Reliable



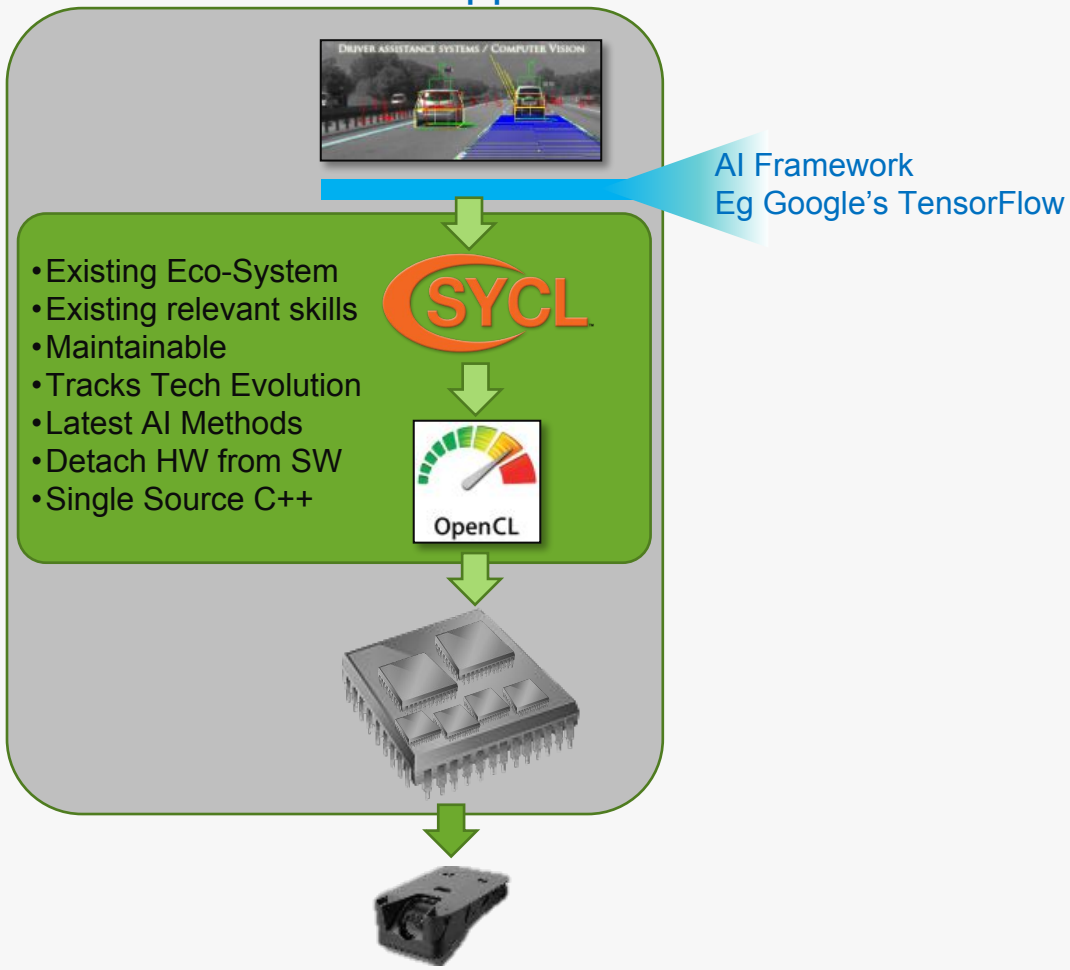
…shrink systems to fit

Evolving Software Infrastructure

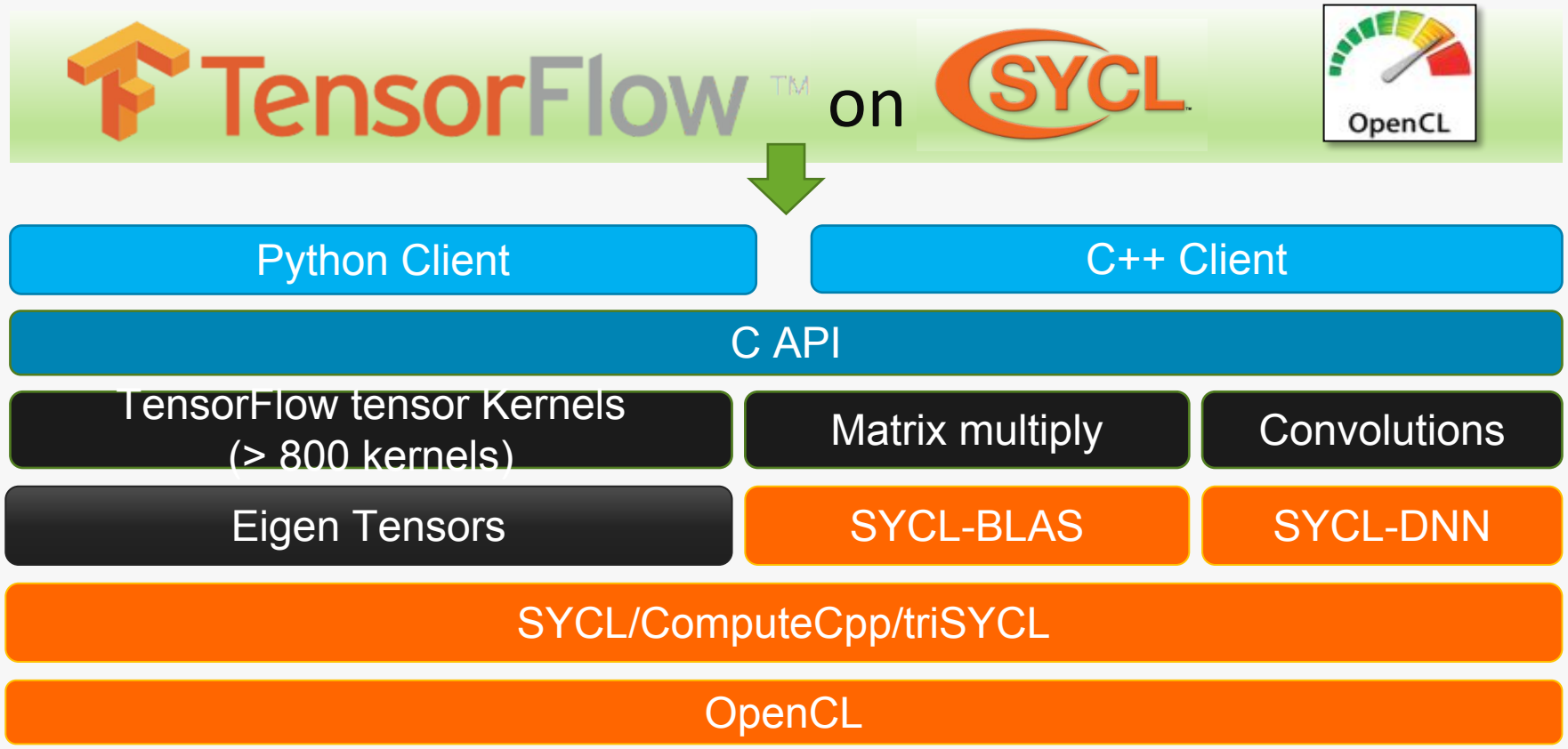
Traditional Automotive Approach



Future Automotive Approach



Typical compute software stack



Functional safety standards

Various industries have safety standards for items used in safety critical domains like:

- DO-178C Avionics
- ISO 26262 Automotive
- IEC 62304 Medicine
- IEC 62279 Railway
- IEC 61511 Process
- IEC 61513 Nuclear

Functional safety standards

Various industries have safety standards for items used in safety critical domains like:

- DO-178C Avionics
- ISO 26262 Automotive
- IEC 62304 Medicine
- IEC 62279 Railway
- IEC 61511 Process
- IEC 61513 Nuclear

+

Cybersecurity standards to mitigate program and system vulnerabilities

- CERT recommendations
- New standards emerging

Functional safety standards

Various industries have safety standards for items used in safety critical domains like:

- DO-178C Avionics
- ISO 26262 Automotive
- IEC 62304 Medicine
- IEC 62279 Railway
- IEC 61511 Process
- IEC 61513 Nuclear

+ **Cybersecurity** standards to mitigate program and system vulnerabilities
  - CERT recommendations
  - New standards emerging

+ **Green programming**
  - Embedded high compute low power devices
  - Deterministic and timely
  - Use of efficient algorithms
  - Use the correct algorithms or library functions

Perform hazard risk analysis

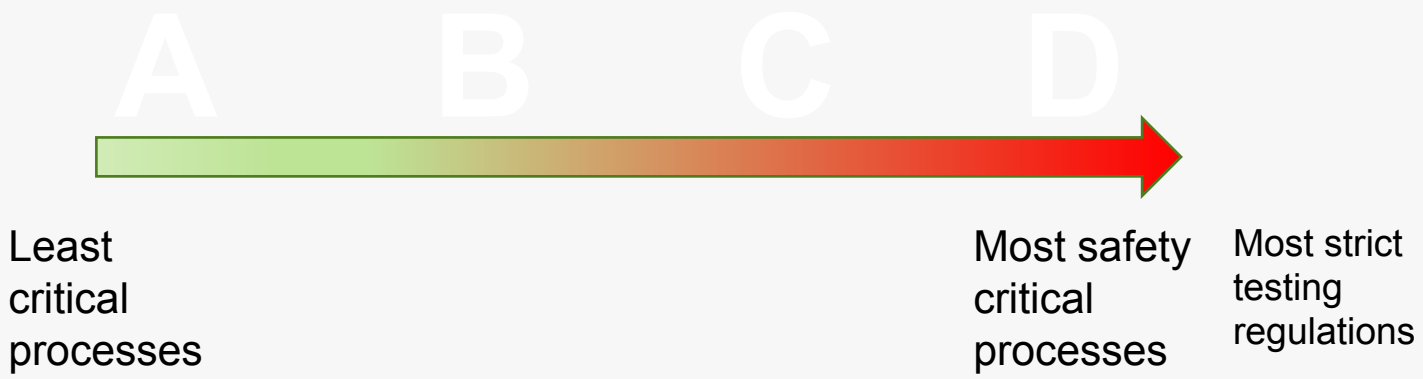
What is the use case for the safety critical item?

1. Explorer and record all the scenarios where injury or death can occur due to item failure
2. For each hazard derive the mitigations action to take to get to a safe state – safety goal
3. Verify the safety goals are met for all operational conditions

What is ISO 26262?

What is ASIL?

Automotive Safety Integrity Level (**ASIL**) is a risk classification scheme **defined** by the **ISO 26262** - Functional Safety for Road Vehicles standard. This is an adaptation of the Safety Integrity Level used in IEC 61508 for the automotive industry.



What is ISO26262?

At the beginning of the safety life cycle, hazard analysis and risk assessment is performed, resulting in assessment of ASIL to all identified hazardous events and safety goals.

Severity Classifications (S)	Severity of failure
S0	No (<10%)
S1	Light moderate injuries (>=10%)
S2	Severe to life threatening
S3	Life threatening

Exposure Classifications (E)	Probability of an issue
E0	Incredibly unlikely/Never/0%
E1	Very low/less than once a year
E2	Low/A few times a year/ <1% average operating time
E3	Medium/Once a month or more/ 1 – 10% average operation time
E4	High/Almost every drive/ >10%

Controllable Classifications (C)	Is Controllable
C0	In general
C1	Simply / 99% of drivers
C2	Normally / 90% of drivers
C3	With difficulty / <90% of drivers

What is ISO26262?

Severity Classifications (S)	Severity of failure
S0	
S1	
S2	
S3	

Exposure Classifications (E)	Probability of an issue
E0	
E1	
E2	
E3	
E4	

Controllable Classifications (C)	Is Controllable
C0	In general
C1	Simply / 99% of drivers
C2	Normally / 90% of drivers
C3	With difficulty / <90% of drivers

Controllability	Exposure	Severity			
		S0	S1	S2	S3
C1	E1	QM	QM	QM	QM
	E2	QM	QM	QM	QM
	E3	QM	QM	QM	A
	E4	QM	QM	A	B
C2	E1	QM	QM	QM	QM
	E2	QM	QM	QM	A
	E3	QM	QM	A	B
	E4	QM	A	B	C
C3	E1	QM	QM	QM	A
	E2	QM	QM	A	B
	E3	QM	A	B	C
	E4	QM	B	C	D

The probability and risk acceptability criteria produce an ASIL level

What is ISO26262?

For software product development ISO26262 supports the correctness of the design and implementation by stipulating guidelines for programming languages addressing the following topics:

Software implementation design:

- Correct execution order
- Interface consistency
- Unnecessary complexity
- Correct data/control flow
- Simplicity
- Readability and comprehensibility
- Robustness
- Change suitability
- Testability
- Maintainability

	Methods	ASIL A	ASIL B	ASIL C	ASIL D
1a	One entry and one exit point in subprograms and functions[a]	++	++	++	++
1b	No dynamic objects or variables, or else online test during their creation[a, b]	+	++	++	++
1c	Initialisation of variables	++	++	++	++
1d	No multiple use of variable names[a]	+	++	++	++
1e	Avoid global variables or else justify their usage[a]	+	+	++	++
1f	Limited use of pointers[a]	o	+	+	++
1g	No implicit type conversions[a, c]	+	++	++	++
1h	No hidden data flow or control flow[b, d]	+	++	++	++
1i	No unconditional jumps[a, c, d]	++	++	++	++
1j	No recursions	+	+	++	++

[a] Methods 1a, 1b, 1c, 1d, 1e, 1f, 1g and 1i may not be applicable for graphical modelling notations used in model-based development.

[b] If these compiler features are "tool qualified" in accordance with ISO 26262-8:—, Clause 10, Method 1b need not be applied if a compiler is used which ensures that there will be enough program storage allocated for all dynamic variables and objects before run-time or which inserts run-time tests for correct online-allocation of program storage, i.e. stack bounds checking.

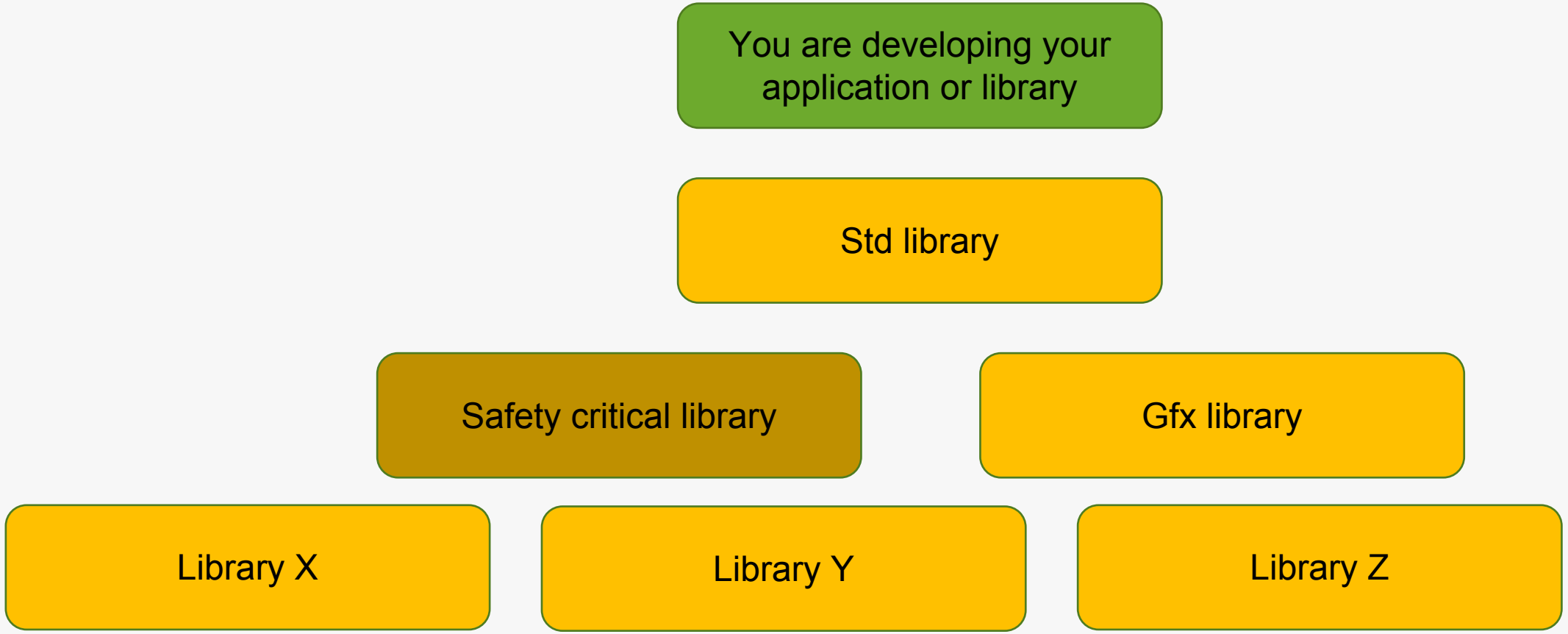
[c] Methods 1g and 1i are not applicable in assembler programming.

[d] Methods 1h and 1i reduce the potential for modelling data flow and control flow through jumps or global variables.

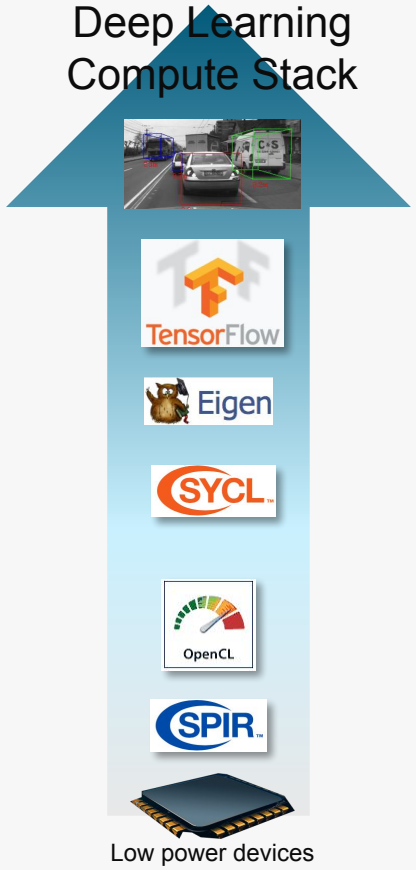
How deep is your love for C++?

You are developing your application or library

How deep is your love for C++?



Conclusion



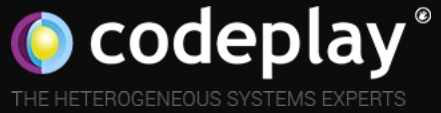
Requires validation & verification
requires tools

Not so fast … we are not finished yet

Tools

For most of the functional safety standards they ask you shall also validate your tools.

This means everything you did to create you program running on your devices needs to be applied to the tools that helped you create your program.



Thank you