



Highest availability needs in-field debugging

Albrecht Mayer, Distinguished Engineer
Infineon automotive microcontroller
MPSoC 2025, Megeve

Supported by:



Bundesministerium
für Bildung
und Forschung

Mannheim **EMDRIVE**

In-field debugging

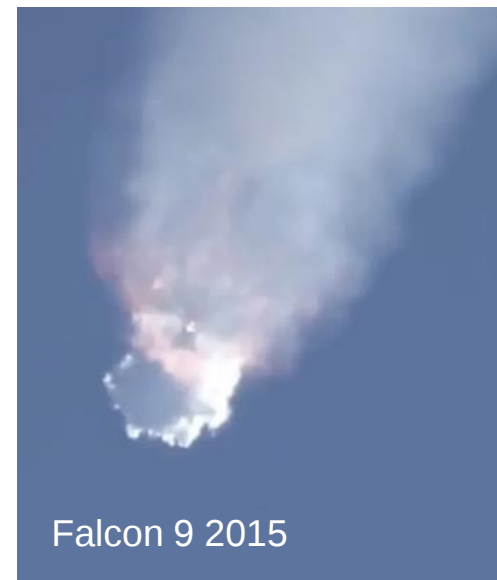
ESA: First time right

SpaceX: Let's do it

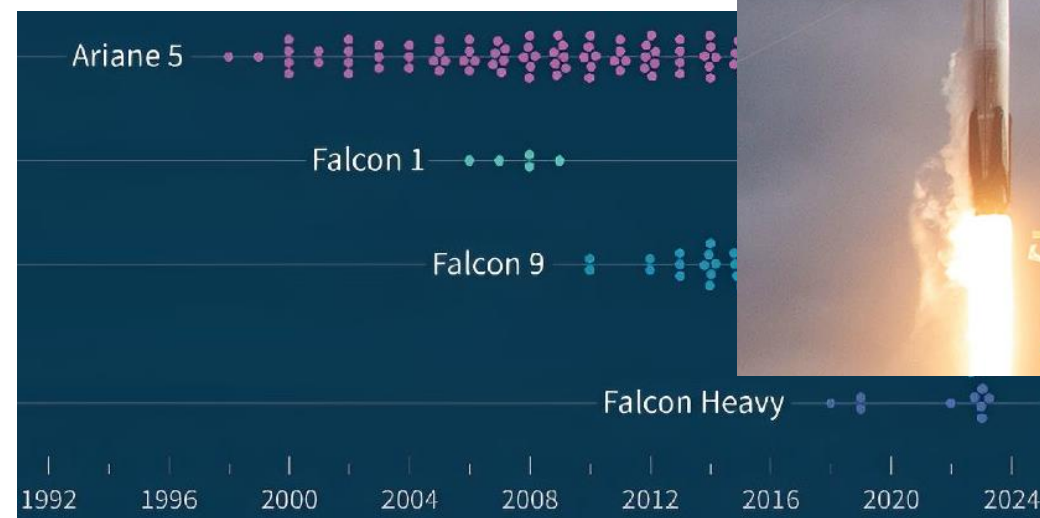
Complex systems:

You can reduce the number of failures in the field by testing, but some effects will not be foreseeable

How to learn best from field failures?



Falcon 9 2015



<https://spacenews.com/nasa-investigation-linked-2015-falcon-9-failure-to-design-error/>

Table of contents

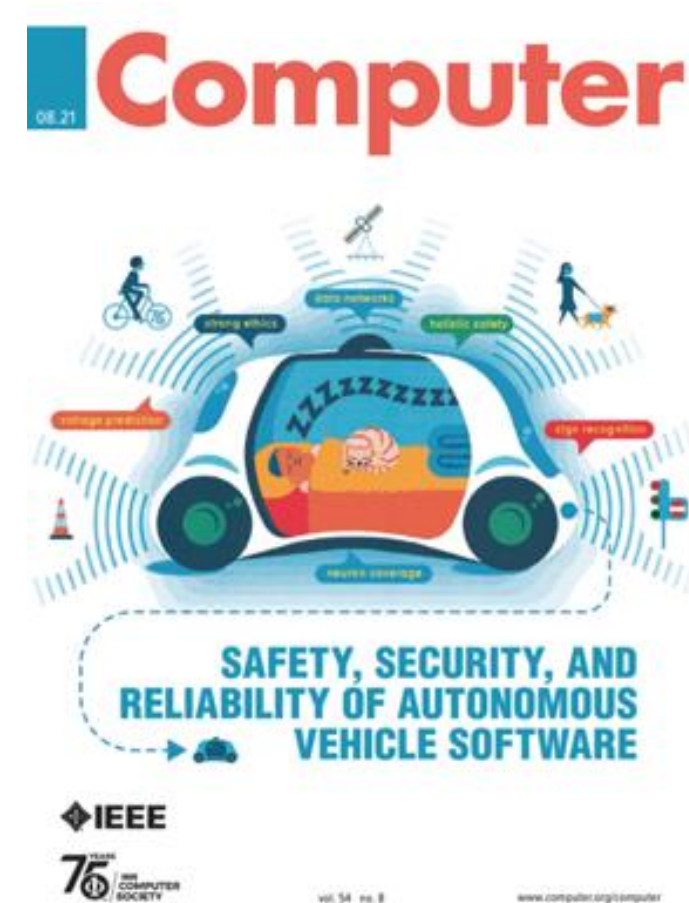
1	Availability	4
2	In field debugging	13
3	Summary	21

Table of contents

1	Availability	4
2	In field debugging	13
3	Summary	21

Motivation for in-field debugging

- Availability is affected more by software issues than by hardware faults: factor 10x
- Understanding very rare issues
- Traditional diagnosis paradox:
Writing error routines for unknown errors
- Non-intrusive hardware on-chip trace has no fundamental conflict with safety

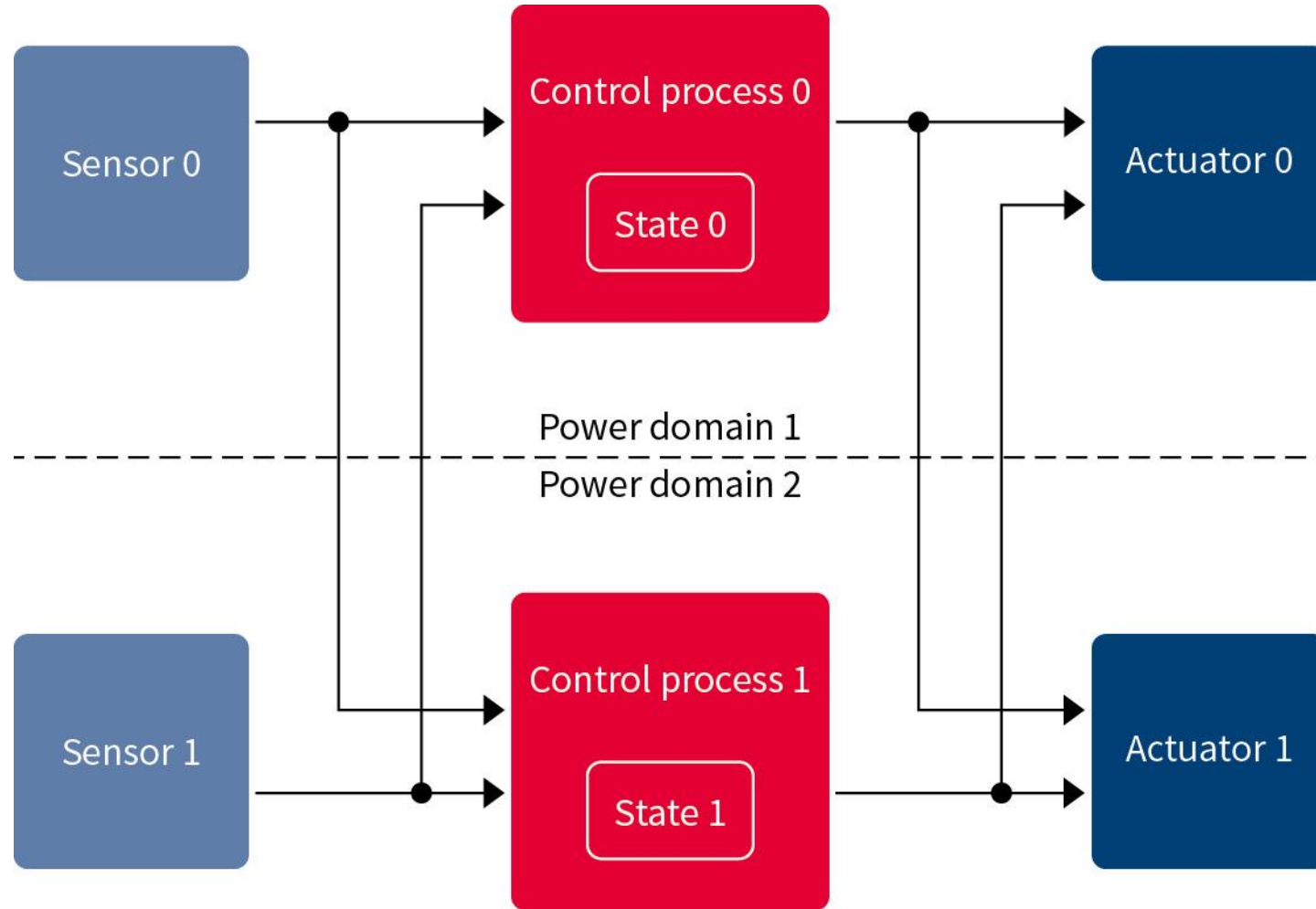


Source: Computer, vol. 54, no. 08, <https://www.computer.org/csdl/magazine/co/2021/08>

Detection of faults for HW

- FIT – Failures per 10E9 operating hours
- 1 FIT ~ 1 failure for 100000 cars with 10000h operation during lifetime
- Safety FIT rate != availability FIT rate
- Safety FIT := undetected dangerous faults
- Availability FIT := undetected + detected dangerous faults (safe state)
- AURIX™ class chip has
 - < 1 FIT safety FIT rate
 - 100x higher availability relevant FIT rate**

Fail-operational on one slide: Random HW faults are no problem



The dark zone of the availability FIT rate

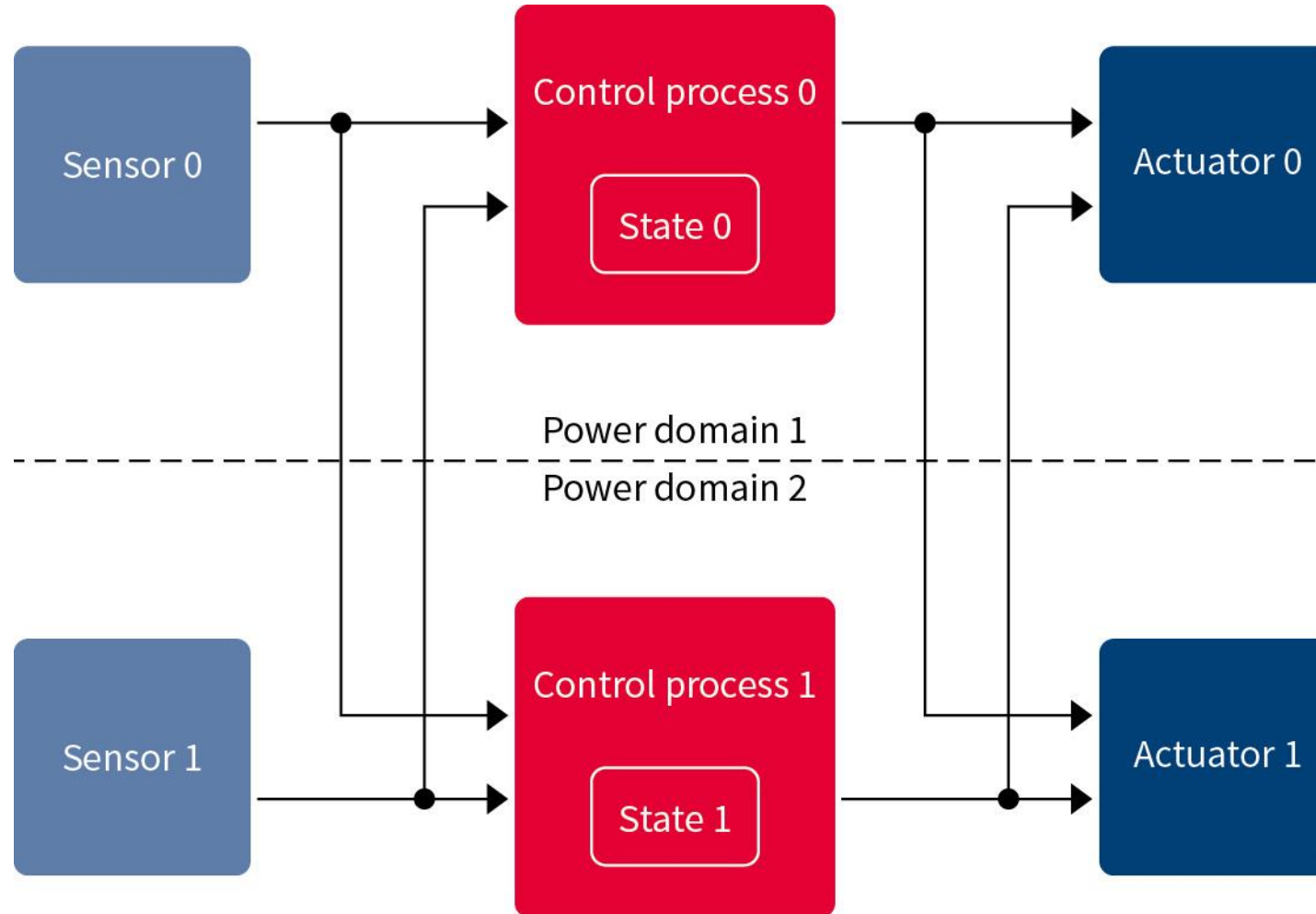


Availability FIT rate: ~1000 FIT for HW + SW



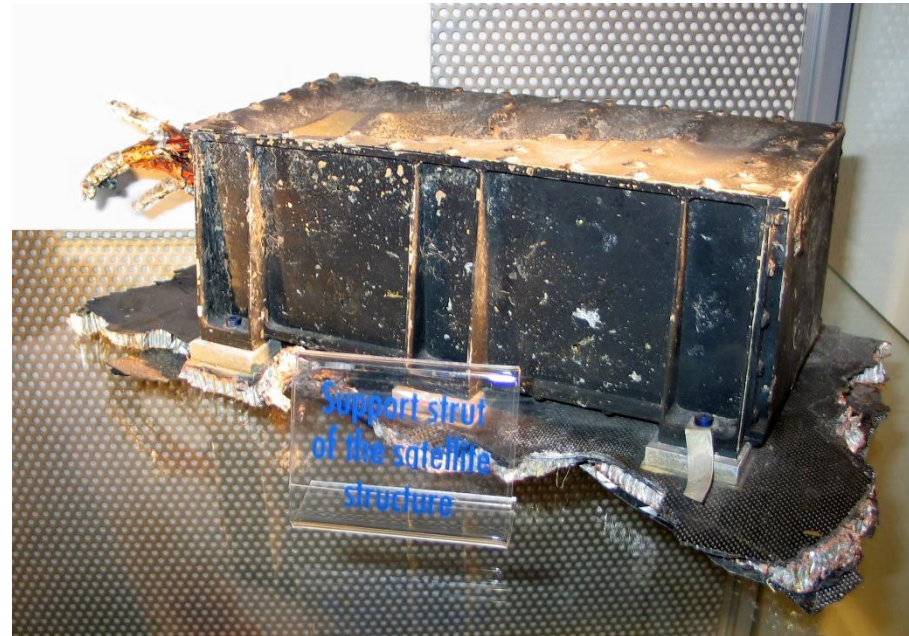
OEMs: > 90% of availability issues by software

Fail-operational on one slide: Systematic software faults are a big problem

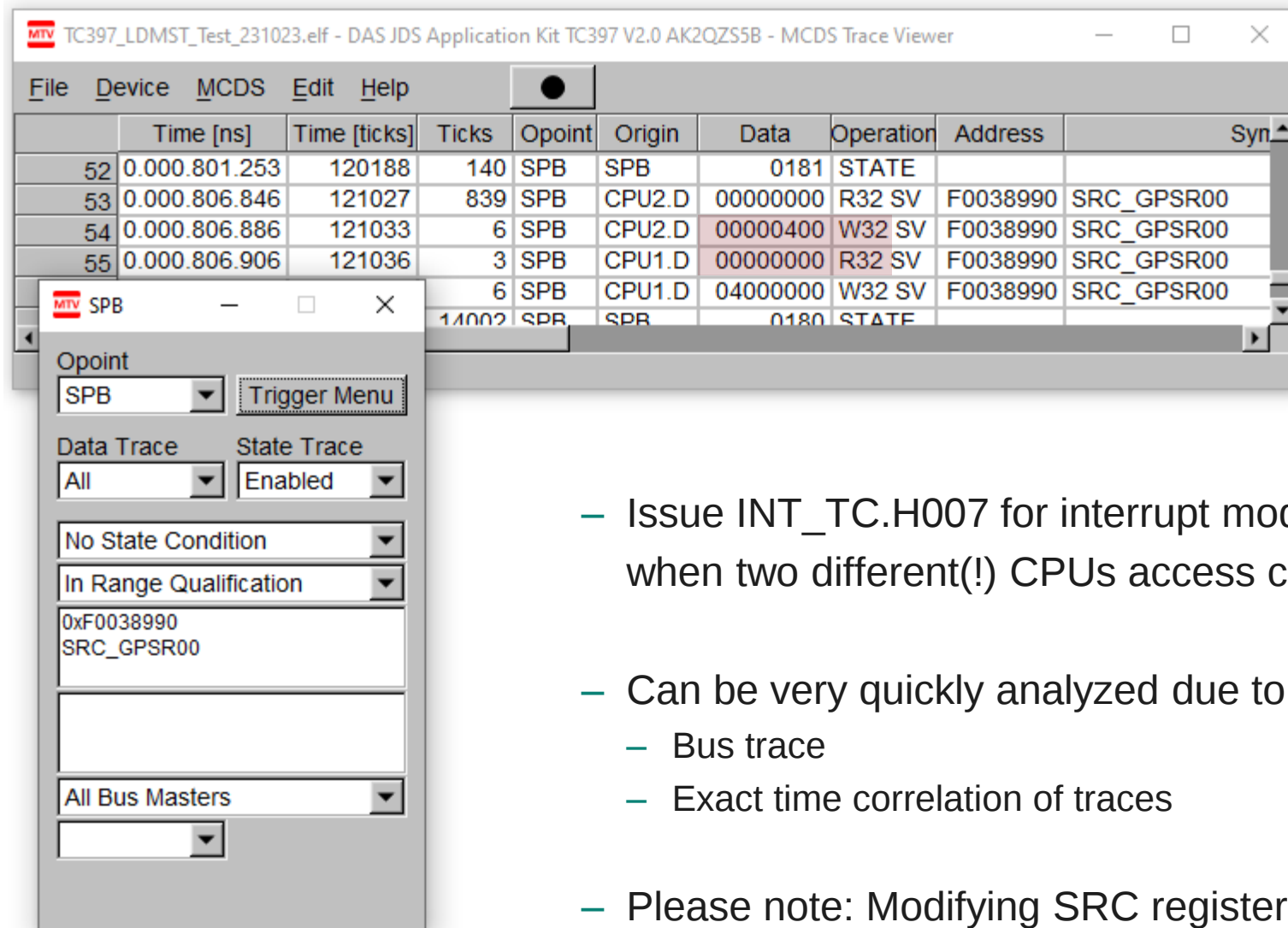


Ariane 5 software bug

You can be sure they tried hard to avoid this



Old story and they didn't use AURIX there can be always surprises



	Time [ns]	Time [ticks]	Ticks	Opoint	Origin	Data	Operation	Address	Syn
52	0.000.801.253	120188	140	SPB	SPB	0181	STATE		
53	0.000.806.846	121027	839	SPB	CPU2.D	00000000	R32 SV	F0038990	SRC_GPSR00
54	0.000.806.886	121033	6	SPB	CPU2.D	00000400	W32 SV	F0038990	SRC_GPSR00
55	0.000.806.906	121036	3	SPB	CPU1.D	00000000	R32 SV	F0038990	SRC_GPSR00
			6	SPB	CPU1.D	04000000	W32 SV	F0038990	SRC_GPSR00
14002				SPB	SPB	0180	STATE		

SPB

Opoint: SPB

Trigger Menu

Data Trace: All

State Trace: Enabled

No State Condition

In Range Qualification

0xF0038990
SRC_GPSR00

All Bus Masters

- Issue INT_TC.H007 for interrupt module: Write value of LDMST can get lost when two different(!) CPUs access close to each other
- Can be very quickly analyzed due to AURIX trace features:
 - Bus trace
 - Exact time correlation of traces
- Please note: Modifying SRC registers with different CPUs is strange

Table of contents

1	Availability	4
2	In field debugging	13
3	Summary	21

In-field debugging

Restrictions

- Safety → trace only
- Security → access protection
- Cost → limited trace resources

Opportunities

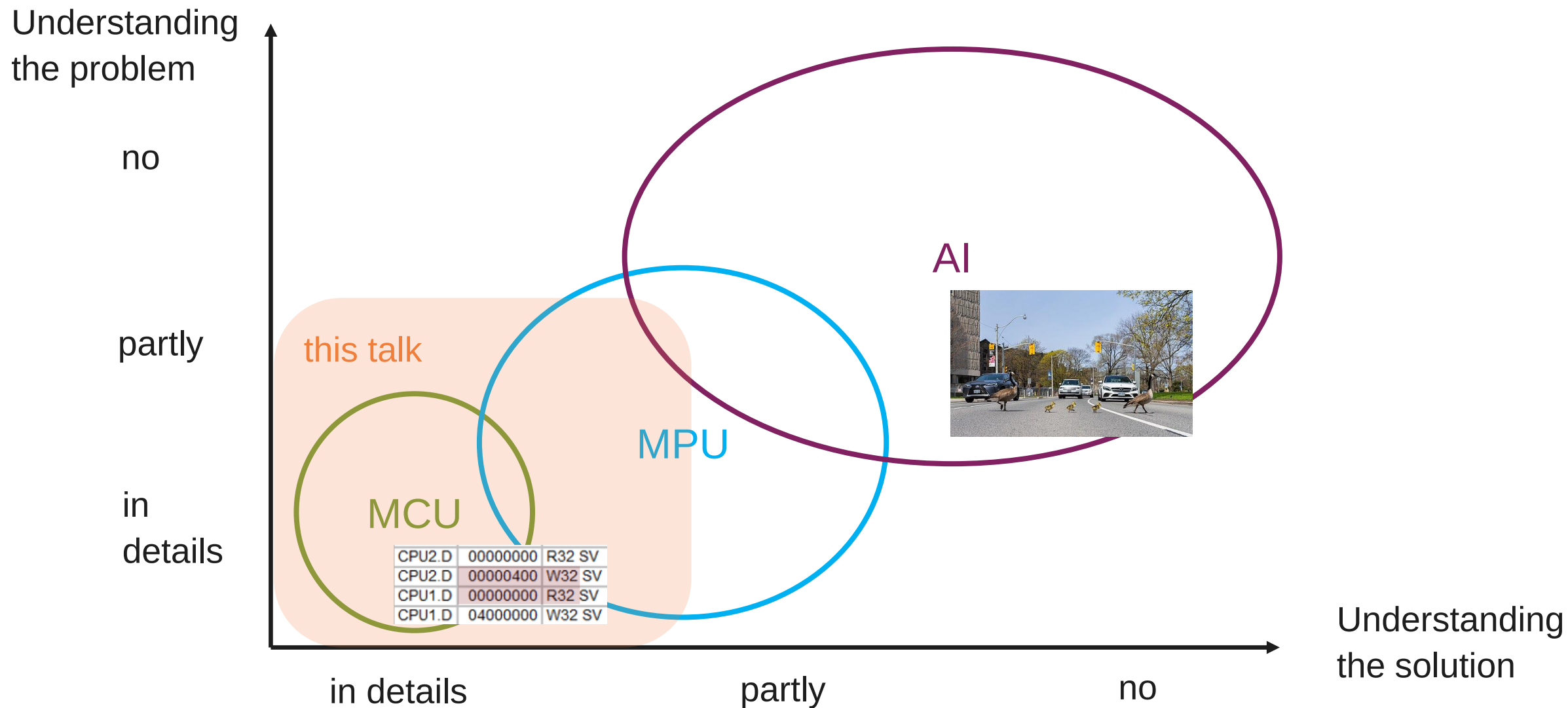
- “Test stand” with thousands of units
- “Text stand” with unexpected test vectors
- Find bugs before there is an accident

Strategy

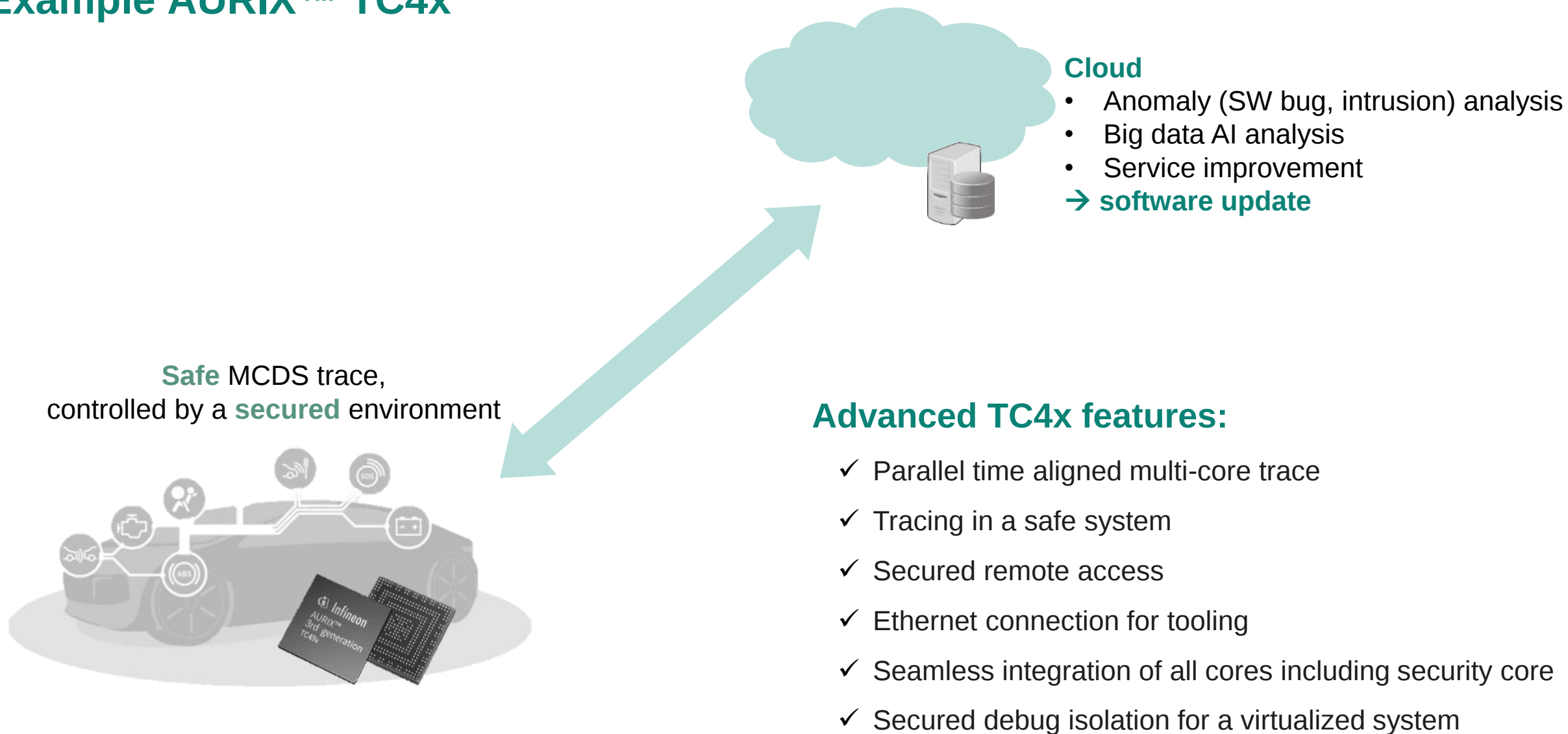
- Enable existing tools where possible
- Create new methods and tools where needed



Problem and solution space from SW programmer perspective



Example AURIX™ TC4x

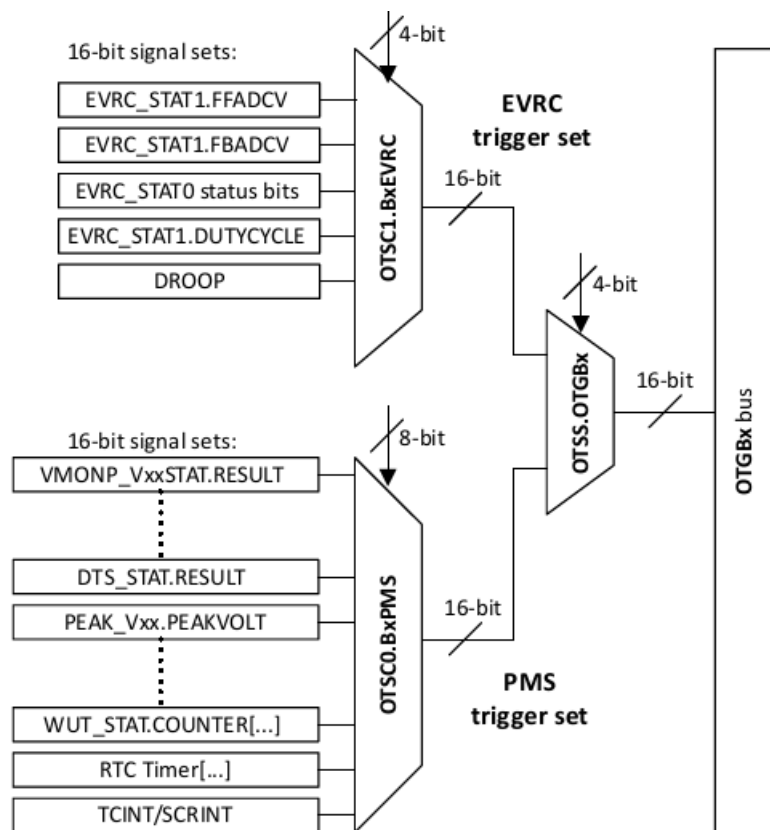


Nothing has happened (yet)



- › Debugging needs a symptom (car falls off cliff)
- › Safe system design: Check if close to the cliff

Demo example: Triggering on low supply voltage



AURIX™ OTGB trace source

- On-chip TriGger Bus
- 2x 16 bit logic analyzer
- Mux structure → any two
- Timer, ADC, PMS, etc.

Demo example: Triggering on low supply voltage

MTV MCAL-TC4x-ADC-PWM-ICU-VIRTUALIZED-CORE-LIB.elf - TriBoard TC4X9 COM V1.0 TB8REYFO - MCDS Trace Viewer

File Device MCDS Edit Help

	Time [ns]	Time [ticks]	Ticks	Opoint	Origin	Data	Operation	Address	Symbol	SO	Comment
-12	0.000.500.104	-19	1	CPU0	CPU0 V2		IP CALL	80400BCC	loop_core1_vm2	0	NOP
-11	0.000.500.188	-18	1	CPU0	CPU0 V3		IP CALL	800153CC 8001552C	core0_vm3_main loop_core0_vm3	44 0	CALL 0x160 NOP
-10	0.000.500.208	-13	5	CPU1	CPU1 V2		IP RET	80400BD2 80400A1C	loop_core1_vm2 core1_vm2_main	6 48	RET LD.HU D15, [A10], 0x0
-9	0.000.500.212	-12	1	CPU0	CPU0 V3		IP RET	80015532 800153D0	loop_core0_vm3 core0_vm3_main	6 48	RET LD.HU D15, [A10], 0x0
-8	0.000.500.216	-11	1	CPU1	CPU1 V2		IP CALL	80400A18 80400BCC	core1_vm2_main loop_core1_vm2	44 0	CALL 0x1b4 NOP
-7	0.000.500.220	-10	1	CPU0	CPU0 V3		IP CALL	800153CC 8001552C	core0_vm3_main loop_core0_vm3	44 0	CALL 0x160 NOP
-6	0.000.500.240	-5	5	CPU0	CPU0 V3		IP RET	80015532 800153D0	loop_core0_vm3 core0_vm3_main	6 48	RET LD.HU D15, [A10], 0x0
-5	0.000.500.240	-5		CPU1	CPU1 V2		IP RET	80400BD2 80400A1C	loop_core1_vm2 core1_vm2_main	6 48	RET LD.HU D15, [A10], 0x0
-4	0.000.500.248	-3	2	CPU0	CPU0 V3		IP CALL	800153CC 8001552C	core0_vm3_main loop_core0_vm3	44 0	CALL 0x160 NOP
-3	0.000.500.252	-2	1	CPU1	CPU1 V2		IP CALL	80400A18 80400BCC	core1_vm2_main loop_core1_vm2	44 0	CALL 0x1b4 NOP
-2	0.000.500.256	-1	1	CPU2	CPU2 V0	00006001	STATE		ISR_END_NESTED2		IEN=0 NESTED=1 VM=0 PRS=0
-1	0.000.500.260	0	1	OTGB	noO1 PMS	0000015D	O10				PMS_TS16_PMS
0	0.000.500.260	0		OTGB	OTGB		Trig OTGB				u16 <= 0x015D
1	0.000.500.264	1	1	CPU2	CPU2 V0	00006011	STATE				IEN=1 NESTED=1 VM=0 PRS=0

MTV Signals (OTGB)

OTGB1 Source: Disabled
OTGB0 Source: PMS
OTGB10 Trigger: Trigger Trace Recording

OTGB10 Trigger Condition: No State Condition

OTGB10 Trigger Expression: u16 <= 0x015D

OTGB10 Trigger Source: OTSC0.B0PMS

In-field debugging with on-chip trace and standard debugger

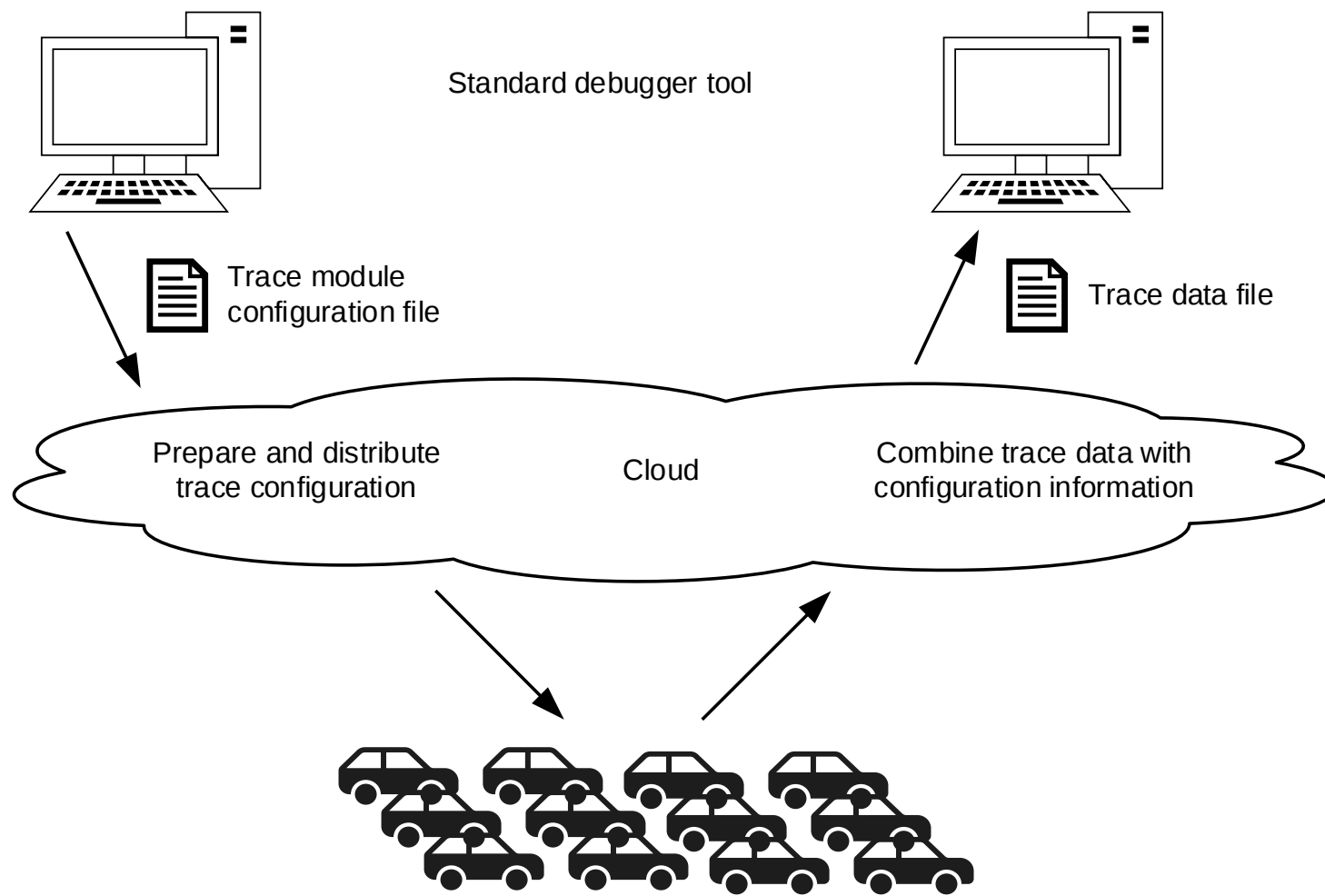


Table of contents

1	Availability	4
2	In field debugging	13
3	Summary	21

Summary

- In-field debugging is needed for very low failure rates
... but can also “just” improves customer experience
- On-chip hardware trace is very flexible, safe and secured (for TC4x)
- Can reuse proven debug tooling
- New AI based debug tooling needed for in-field 100k units parallel “testing”