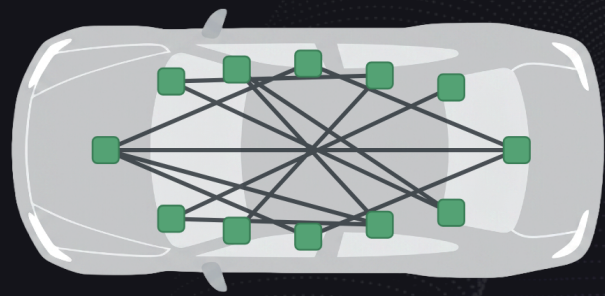


Three generations of moving the compute, and the wiring with it.

01 DISTRIBUTED

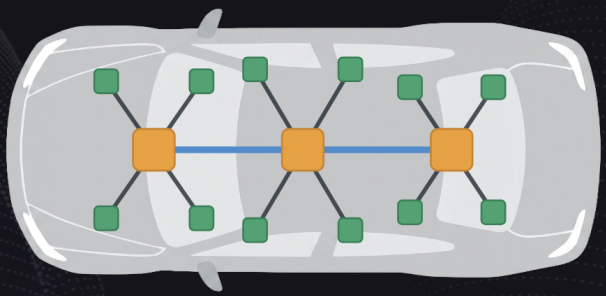


COMPUTE
One microcontroller per function, dozens of them, none of them in charge.

WIRING
Point-to-point copper, with a bespoke harness for every model and trim.

Limit: every new feature means new hardware, and there is no way to update the car after it ships.

02 DOMAIN

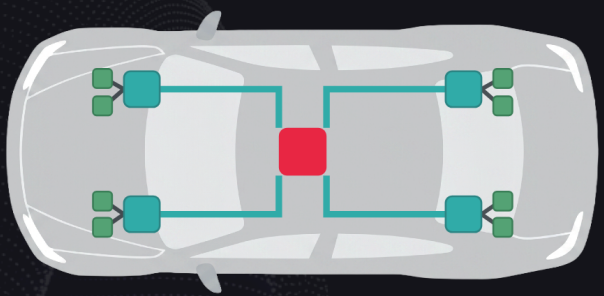


COMPUTE
A few domain controllers group compute by function — powertrain, body, infotainment.

WIRING
CAN and LIN still run the length of the car to reach scattered ECUs.

Limit: compute is grouped, no centralized and each controller still runs its own software stack.

03 ZONAL



COMPUTE
Controllers move to physical zones and a central computer starts to take over.

WIRING
Shorter local runs, but automotive Ethernet between every zone and the center.

Limit: still several computers, each thinking for itself. The wiring got centralized, the brains didn't.

ECU / device

Domain controller

Zone controller

Central computer

Local wire

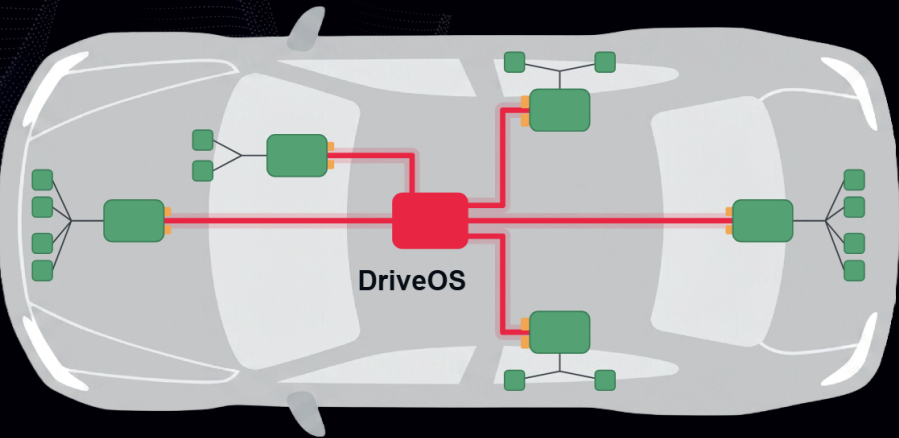
CAN

Automotive Ethernet

DRAKO ARCHITECTURE

One brain. One Backbone. Simple I/O

One single ECU with simple I/O controllers on a single unified backbone: centralised compute and simplified, extensible wiring.



DriveOS PC
One USB cable per DrakIO
Local sub-harnesses only

DrakIO: a very low cost I/O controller that gives the PC the interfaces it lacks — CAN, LIN, GPIO, PWM and analog I/O — and acts on DriveOS instructions in both directions. Add another wherever you need I/O — plug and play

Buses go virtual. DriveOS bridges multiple protocols in software (CAN, LIN, Ethernet, HDMI and more), so two devices behave as if they share a bus with no physical wires connecting them — the differentiation moves off the harness and into software.

COMPUTE

Commodity PC
A multicore PC with hardware virtualisation runs what used to be 100+ separate ECUs, as software tasks.

Every domain, one platform
Powertrain, chassis, body, HVAC, cluster, infotainment and ADAS — no microcontroller per function.

RTOS and Linux, side by side
A partitioning hypervisor splits cores, memory and I/O between them, then leaves the runtime path — so the trusted code base is hard real time and very compact.

Bounded latency
Critical I/O is consolidated across USB — 108 µs guaranteed end to end, sensor to actuator, under load.

ONE CABLE, THREE JOBS

1

Device control (I/O)
Sensor and actuator data, bounded latency

2

Power delivery
48 V native over USB-PD

3

Data and networking
High-bandwidth device I/O and host-to-host transfers

A traditional harness needs three separate runs each on its own medium **USB already does all three** — in every phone, laptop and dock on earth. The one thing it lacked for vehicles was determinism. So we invented **real-time USB**. Standard protocol, off-the-shelf controllers, no dedicated Ethernet PHY per node.

THE DRAKO ARCHITECTURE ENABLES OTA · ADAS natively · Off-the-shelf hardware Supplier freedom · Data ownership · AI flexibility

BRING YOUR SITUATION AND YOUR QUESTIONS

Book a 1:1 meeting with our president