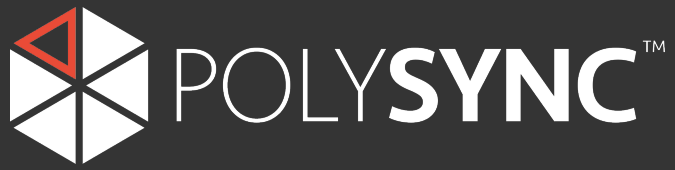
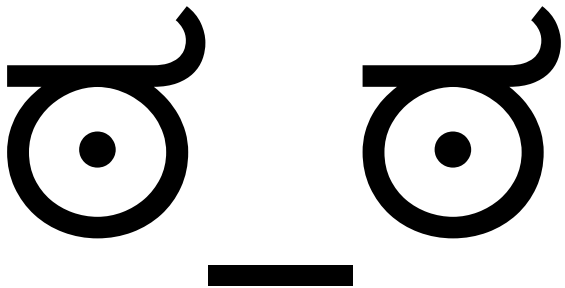


Properties of Chaos

Nathan Aschbacher

@gen_nja



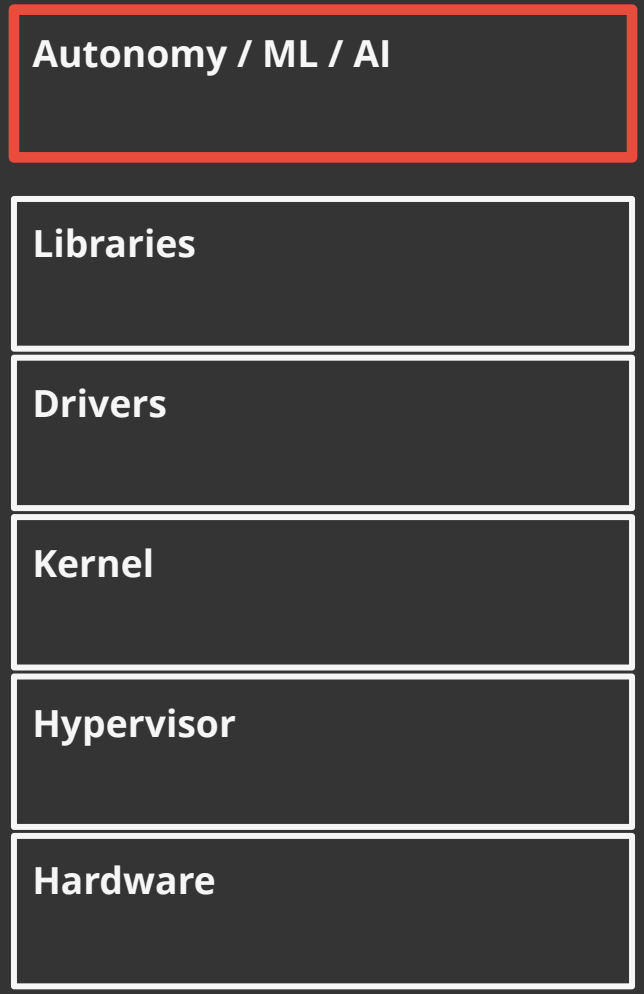
“...we suspect most users are not working on these kinds of safety-critical systems.”

-- Chaos Engineering the Book

Must Be at Least this 26262 to Ride

“Functional Safety in AI Controlled Vehicles: If Not ISO 26262, Then What?”

Joseph Dailey
Global Functional Safety Manager
Mentor


Autonomy / ML / AI
Libraries
Drivers
Kernel
Hypervisor
Hardware

“227 issues”

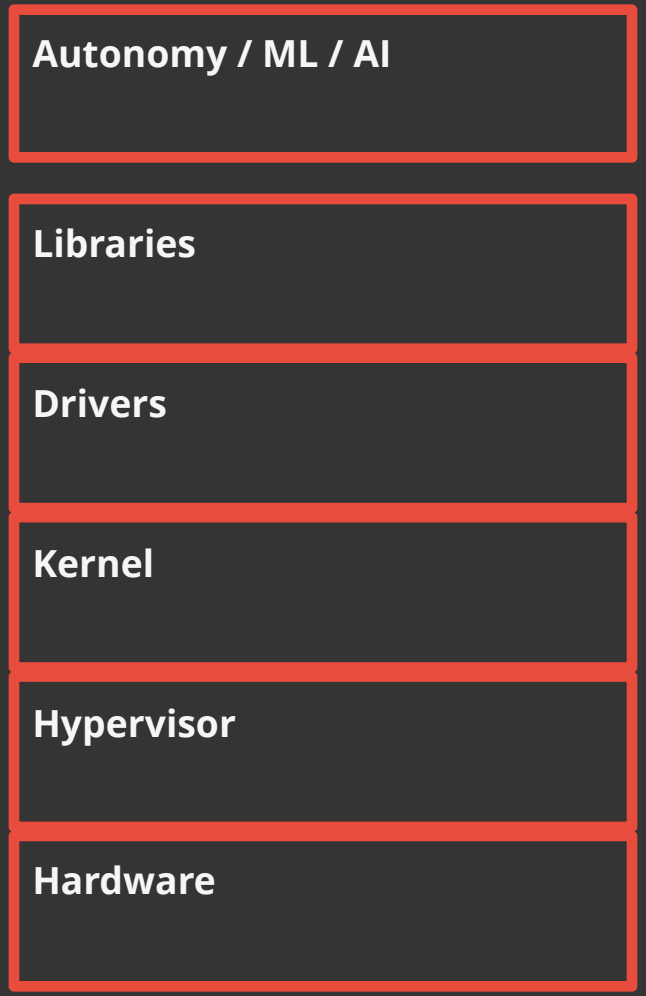
Testing AUTOSAR software with QuickCheck

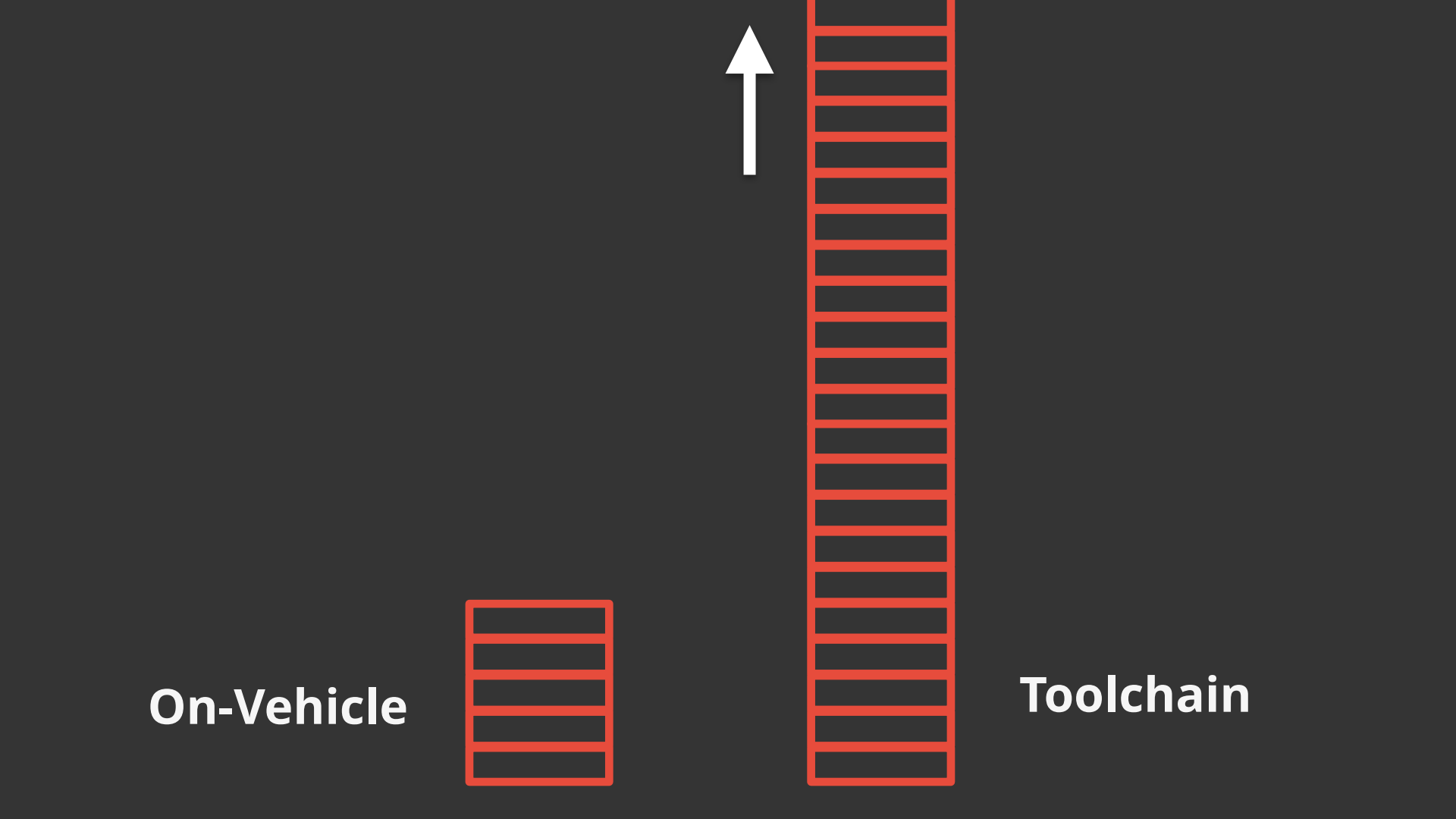
Thomas Arts, John Hughes, Ulf Norell, Hans Svensson
Quviq AB
Gothenburg, Sweden
thomas.arts@quviq.com

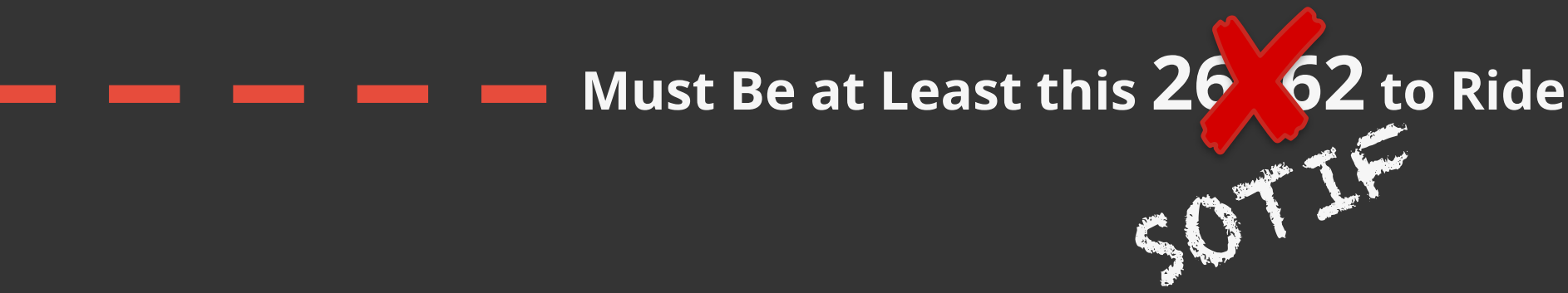
***Abstract*—AUTOSAR (AUTomotive Open System ARchitecture) is an evolving standard for embedded software in vehicles, defined by the automotive industry, and implemented by many different vendors. A modern car may contain over 100 processors, running AUTOSAR software from a variety of different suppliers, which *must* work together for the car to function. On behalf of Volvo Cars, we have developed model-based acceptance tests for some critical AUTOSAR components, to guarantee that the implementations from different vendors are compatible. We translated over 3000 pages of textual specifications into QuickCheck models, and tested many different implementations using large volumes of random test cases generated from these models. This resulted in over 200 issues which we raised with Volvo and the software vendors. We compare our approach with an earlier manual approach, and argue that ours is more efficient, more effective, and more correct.**

C programming language) and hundreds of requirements. Typically such a document is a few hundred pages long.

The first OEM to produce a car model completely based upon a new version of this software standard faces a risk of incompatibility between components from different vendors. The risk can be mitigated by buying all software from the *same* vendor, but most OEMs like to choose between different solutions. Of course, all components are already tested by their vendors, but only against the vendor's own interpretation of the standard. Therefore, Volvo Car Corporation (VCC) collaborated with SP and Quviq to define acceptance tests[1] for vendor specific implementations of the Basic Software Modules—in particular, for 20 modules including the communication part, the communication stacks, and diagnostics.


Autonomy / ML / AI
Libraries
Drivers
Kernel
Hypervisor
Hardware


On-Vehicle
Toolchain



2019

Fail-Operational

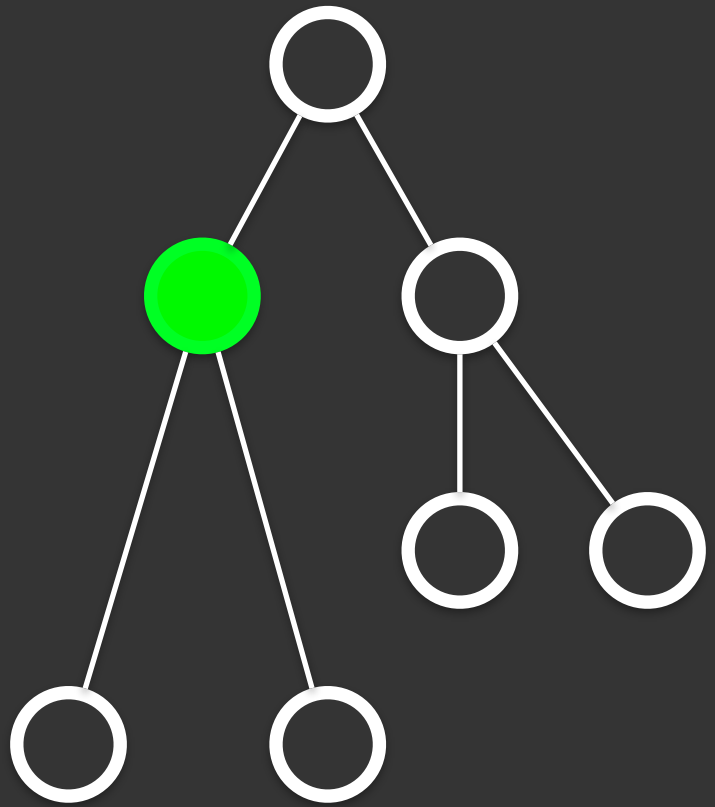
ϕ Temporal Logic

⊢ Higher Order Logic

Π Dependent Types

π Join Calculus

Mathematical Proof



“Chaos Engineering is about engineering practices that help us surface those systemic effects.”

-- Casey Rosenthal

Co-author of Chaos Engineering Book

“Chaos strongly prefers to experiment directly on production traffic.”

-- Principles of Chaos

“We don’t expect engineers to inject noise into the sensors of self-driving cars containing unsuspecting passengers!”

-- Chaos Engineering the Book

Property Chaosed Testing

```
Background:
  Given a process Alice
    And a process Bob
    And an arbitrary vector of processes Carls
    And a message capability from Bob to Alice
    And a message capability from Alice to Bob
    And the Carls continuously send arbitrary messages to Alice

Scenario:  Message delay from unauthorized IPC storm
  When Alice sends an arbitrary message to Bob
    And Bob replies to message from Alice
  Then the message delay from Bob to Alice must be < 1 ms
```

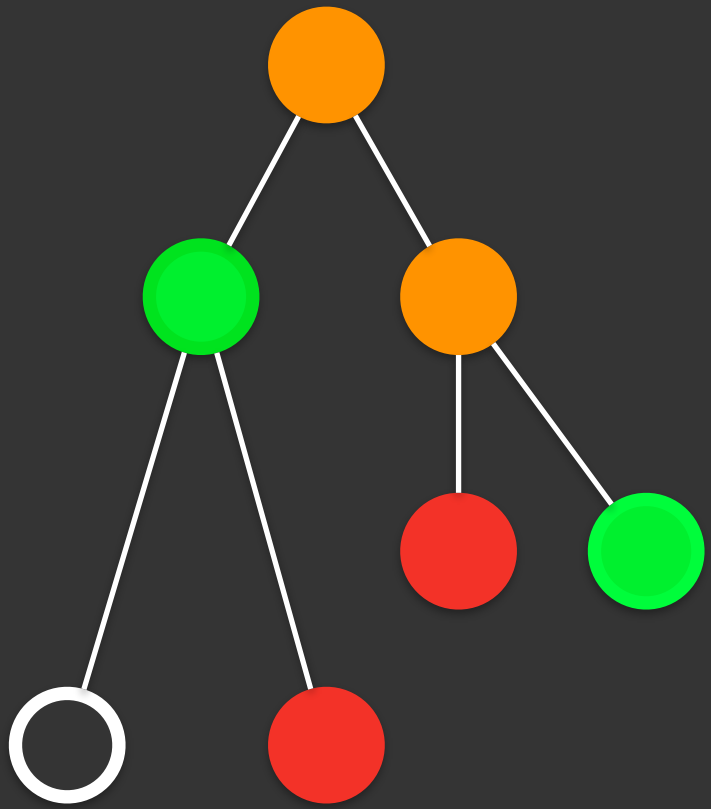
```rust
impl Arbitrary for message {
    fn arbitrary<G: Gen>(g: &mut G) -> message {
        message {
            id: u32::arbitrary(g),
            dlc: u8::arbitrary(g),
            timestamp: u32::arbitrary(g),
            data: [u8::arbitrary(g); 8],
        }
    }
}
```

Common Cause

Cascading

Interference

Common Mode

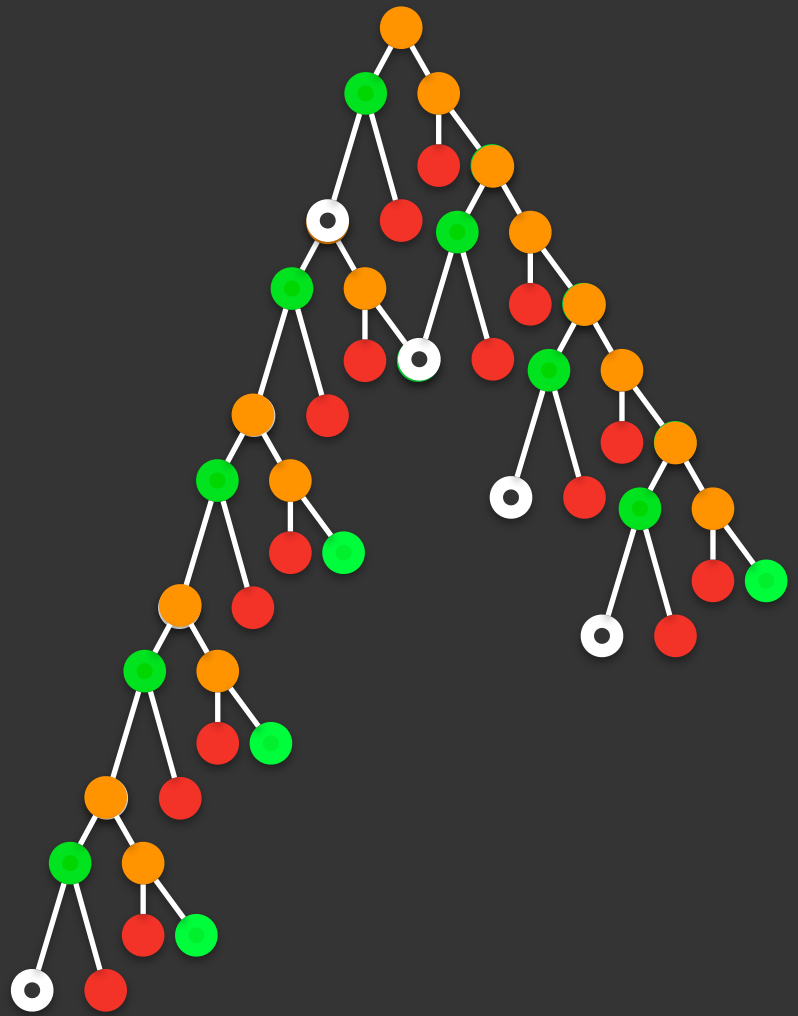


Perceiving

Planning

Performing

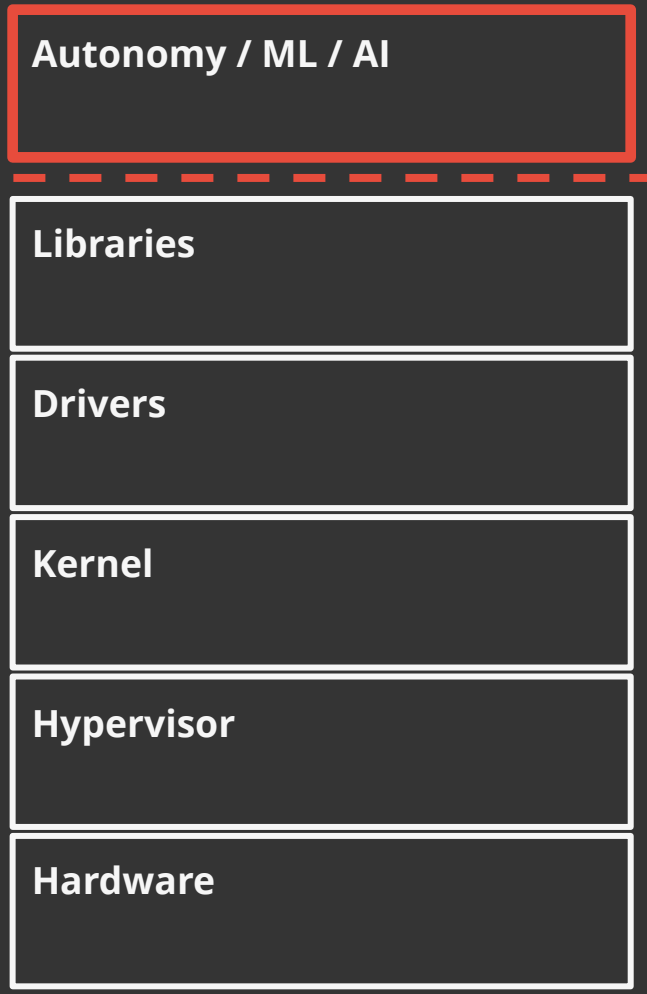
Time < ∞



Before > After

Necessarily Probabilistic

Terrifyingly Probabilistic



The robots are coming.

?