



A FLEXIBLE, VERIFIABLE, AND VALIDATEABLE APPROACH TO AUTONOMOUS VEHICLE SAFETY

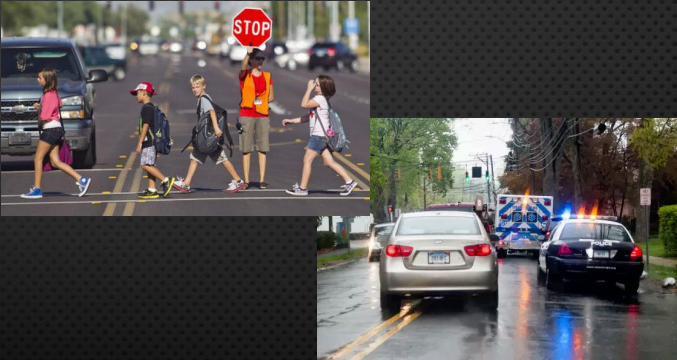
PAUL J. PERRONE



JANUARY 30, 2021

SAFETY VS CERTIFICATION AXIOMS

SAFETY



SOMETHING CERTIFIED MAY NOT BE SAFER

CERTIFICATION

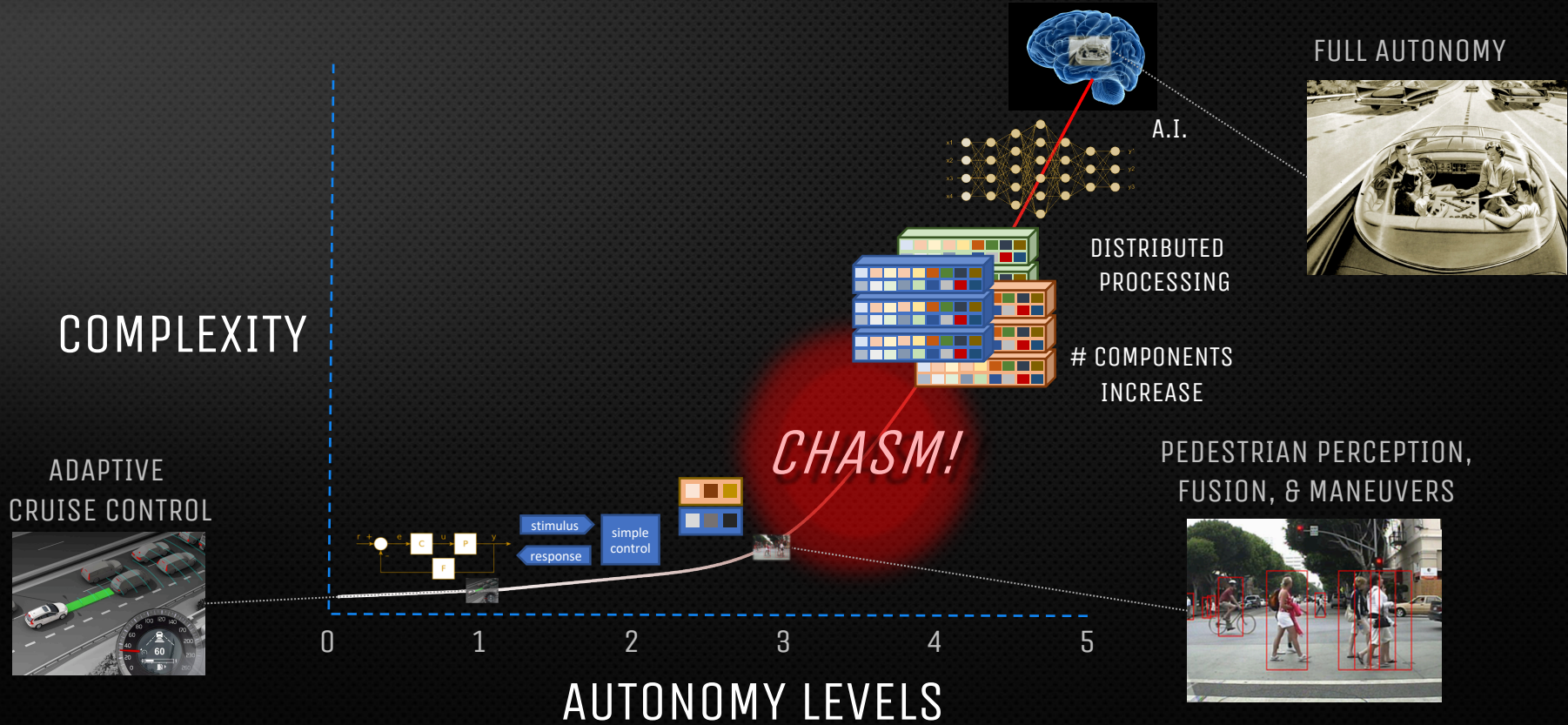


SOMETHING SAFER MAY NOT BE CERTIFIED

ESPECIALLY TRUE WITH SOFTWARE SYSTEMS

CERTIFICATION HARD TO KEEP UP WITH EXPONENTIAL TECHNOLOGY GROWTH

COMPLEXITY INCREASE WITH AUTONOMY



WORLDS COLLIDING

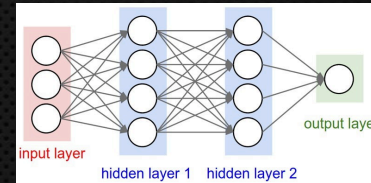
AUTOMOTIVE CONTROLS



LOW-LEVEL
SIMPLER
EASIER TO V&V

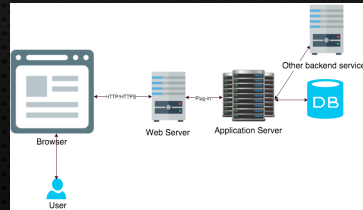
FULL
AUTONOMY

ARTIFICIAL INTELLIGENCE



ABSTRACT/PROBABILISTIC-LEVEL
HIGHEST COMPLEXITY
IMPOSSIBLE TO V&V

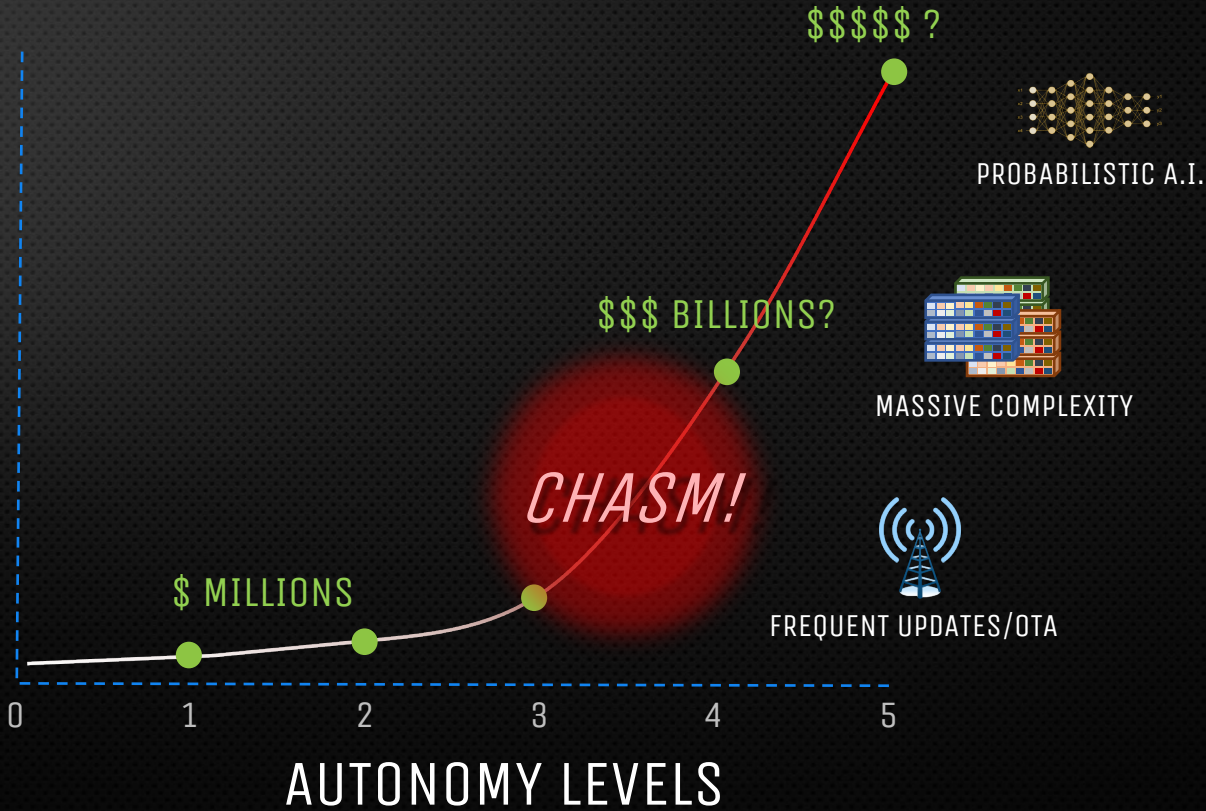
LARGE SCALEABLE IT SOFTWARE



HIGHER-LEVEL
SCALABLE COMPLEXITY
DIFFICULT TO V&V

NON-SCALING LEGACY CERTIFICATION

CERTIFICATION
COSTS



SCALING FOR FULL AUTONOMY

ARTIFICIAL INTELLIGENCE

MACHINE LEARNING METHODS
(NEURAL NETWORKS, MARKOV MODELS, RULES)

LARGE SCALEABLE IT SOFTWARE

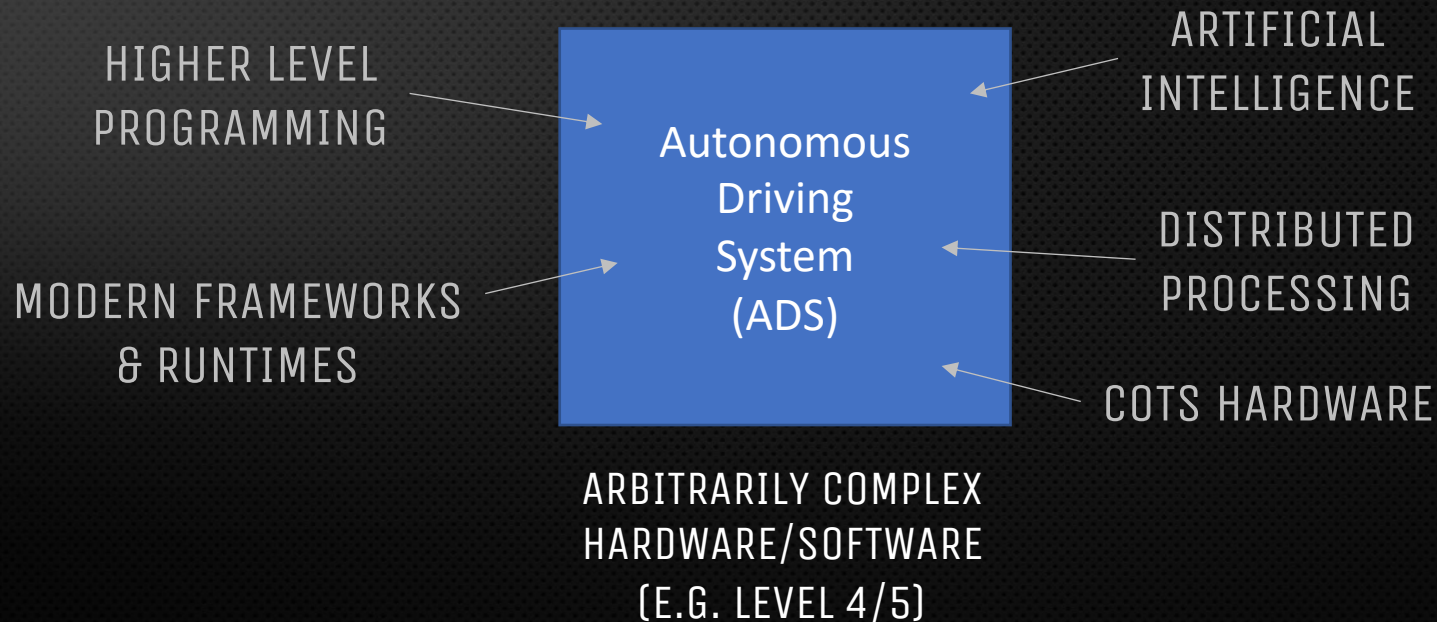
HIGHER LEVEL PROGRAMMING LANGUAGES
(JAVA, C#, PYTHON, RUBY)

HIGHER LEVEL PROGRAMMING METHODS
(OBJECT-ORIENTED, FRAMEWORKS, MANAGED RUNTIMES)

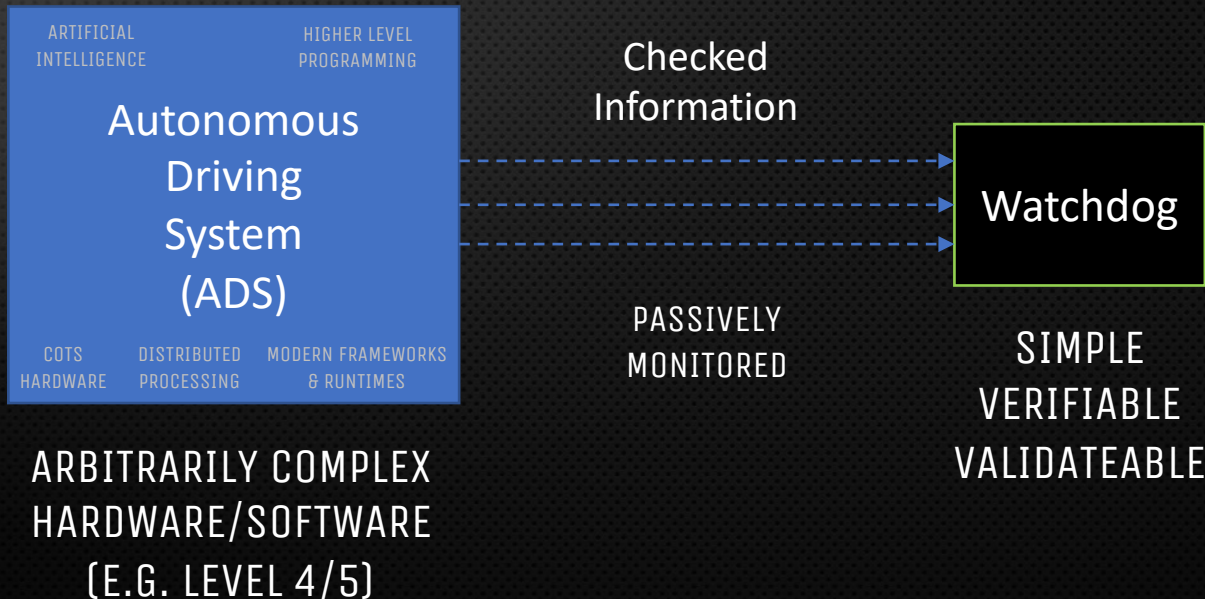
EMBEDDED-IT INTERSECTIONS

HIGH-LEVEL REAL-TIME PROGRAMMING
(REAL-TIME OS, REAL-TIME JAVA)

HOW DO WE MAKE THIS SAFE?



SAFETY WATCHDOG



WATCHDOG SPECIFICATION PROCESS

SYSTEM SPECIFICATION

SAFETY-CRITICAL SPEC SUBSET

MAPPED V&V'D SAFETY CHECKING FUNCTIONS

MAPPED V&V'D WATCHDOG RULES

```
1. The system shall do this.
1.1 The system shall do this and that.
1.1.1 The system shall do this and that.
1.2 The system shall do this and that.
1.2.1 The system shall do this and that.
1.2.2 The system shall do this and that.
1.3 The system shall do this and that.
2. The system shall do this and that.
2.1 The system shall do this and that.
2.2 The system shall do this and that.
2.3 The system shall do this and that.
2.3.1 The system shall do this and that.
2.3.2 The system shall do this and that.
2.3.3 The system shall do this and that.
2.3.4 The system shall do this and that.
2.3.4.1 The system shall do this and that.
2.3.4.2 The system shall do this and that.
2.4 The system shall do this and that.
3. The system shall do this and that.
3.1 The system shall do this and that.
3.1.1 The system shall do this and that.
3.1.2 The system shall do this and that.
```

SAFETY ANALYSIS

```
1. The system shall do this.
1.1 The system shall do this and that.
1.1.1 The system shall do this and that.
1.2 The system shall do this and that.
1.2.1 The system shall do this and that.
1.2.2 The system shall do this and that.
1.3 The system shall do this and that.
2. The system shall do this and that.
2.1 The system shall do this and that.
2.2 The system shall do this and that.
2.3 The system shall do this and that.
2.3.1 The system shall do this and that.
2.3.2 The system shall do this and that.
2.3.3 The system shall do this and that.
2.3.4 The system shall do this and that.
2.3.4.1 The system shall do this and that.
2.3.4.2 The system shall do this and that.
2.4 The system shall do this and that.
3. The system shall do this and that.
3.1 The system shall do this and that.
3.1.1 The system shall do this and that.
3.1.2 The system shall do this and that.
```

SAFETY FUNCTIONS

```
1. Safety Functions - AECIT will avoid wrong
derivations.
This safety function is to address the hazard of
a person or machine operating the system
and to ensure that the system is not
operating in a state that is not intended
for normal operation. The system shall
be able to detect and respond to any
condition that could lead to a safety
critical failure. The function shall
be able to detect and respond to any
condition that could lead to a safety
critical failure. The function shall
be able to detect and respond to any
condition that could lead to a safety
critical failure.
```

HAZARD ANALYSES,
FORMAL METHODS,
OR V&V'D METHODS

DERIVATION PROCESS

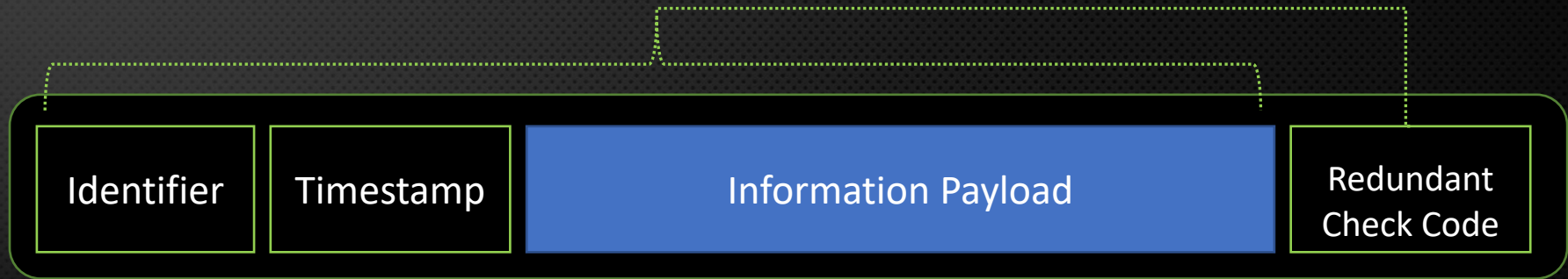
```
if X > Y, then Z = off
assert beta = 20
SBD status on
if Rtheta > TBD, then speed.limit = 20
if X > Y, then Z = off
assert beta = 20
SBD status on
if Rtheta > TBD, then speed.limit = 20
```

DERIVATION PROCESS

```
Gps.qual < X
if speed > X, steering < Y
Assert SBD < X
Lidar.1.status = 1
```

VERIFIABLE & VALIDATEABLE PROCESS

CODED INFORMATION

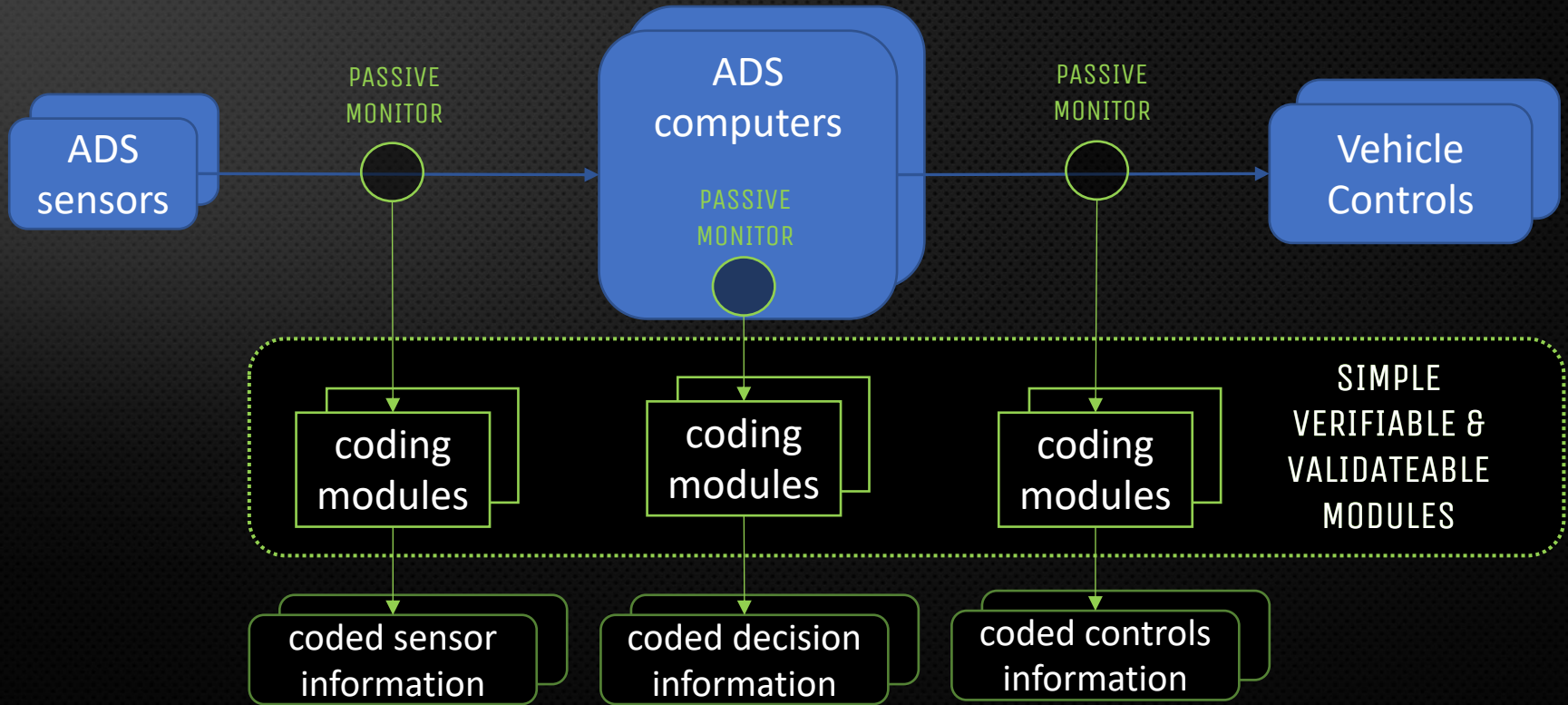


UNIQUELY
IDENTIFY
INFO

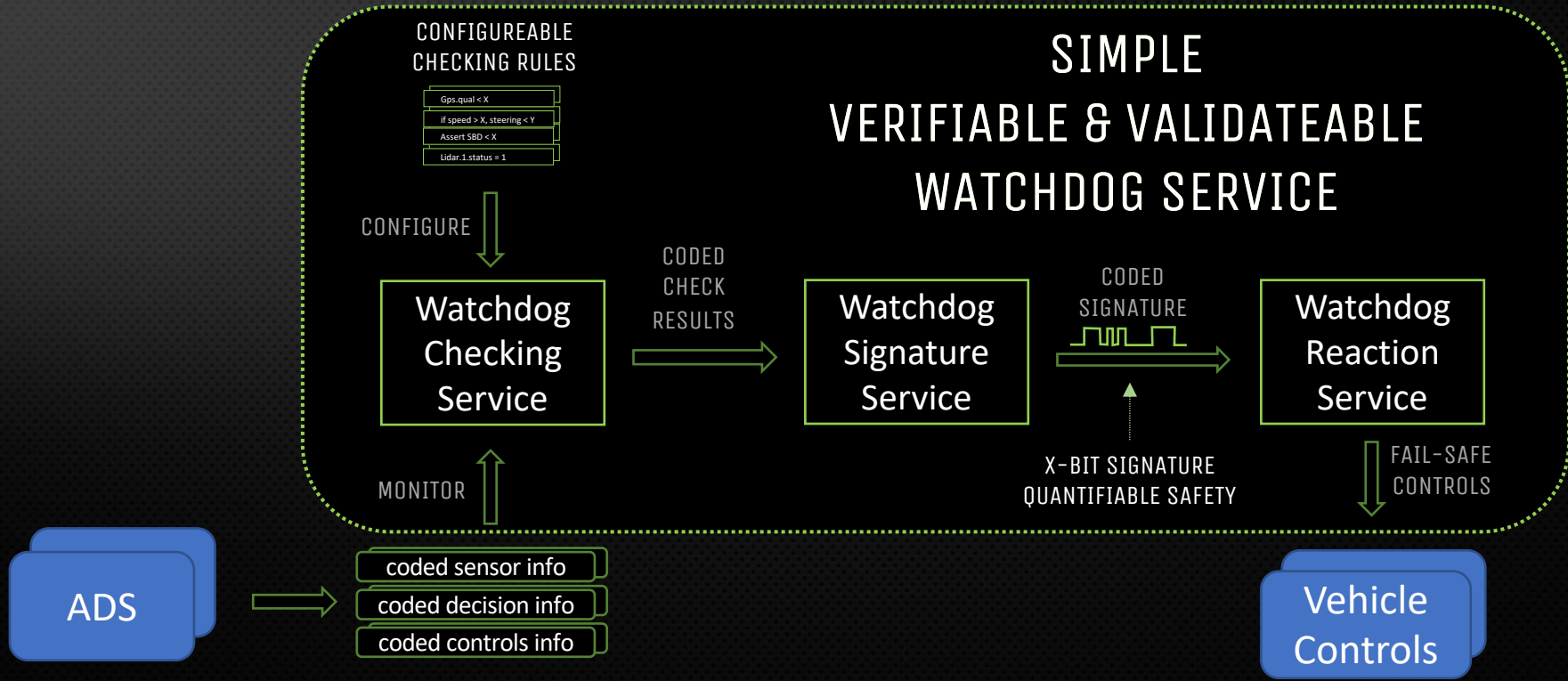
ENSURE
INFO
"FRESHNESS"

PROTECT
AGAINST
CORRUPTION

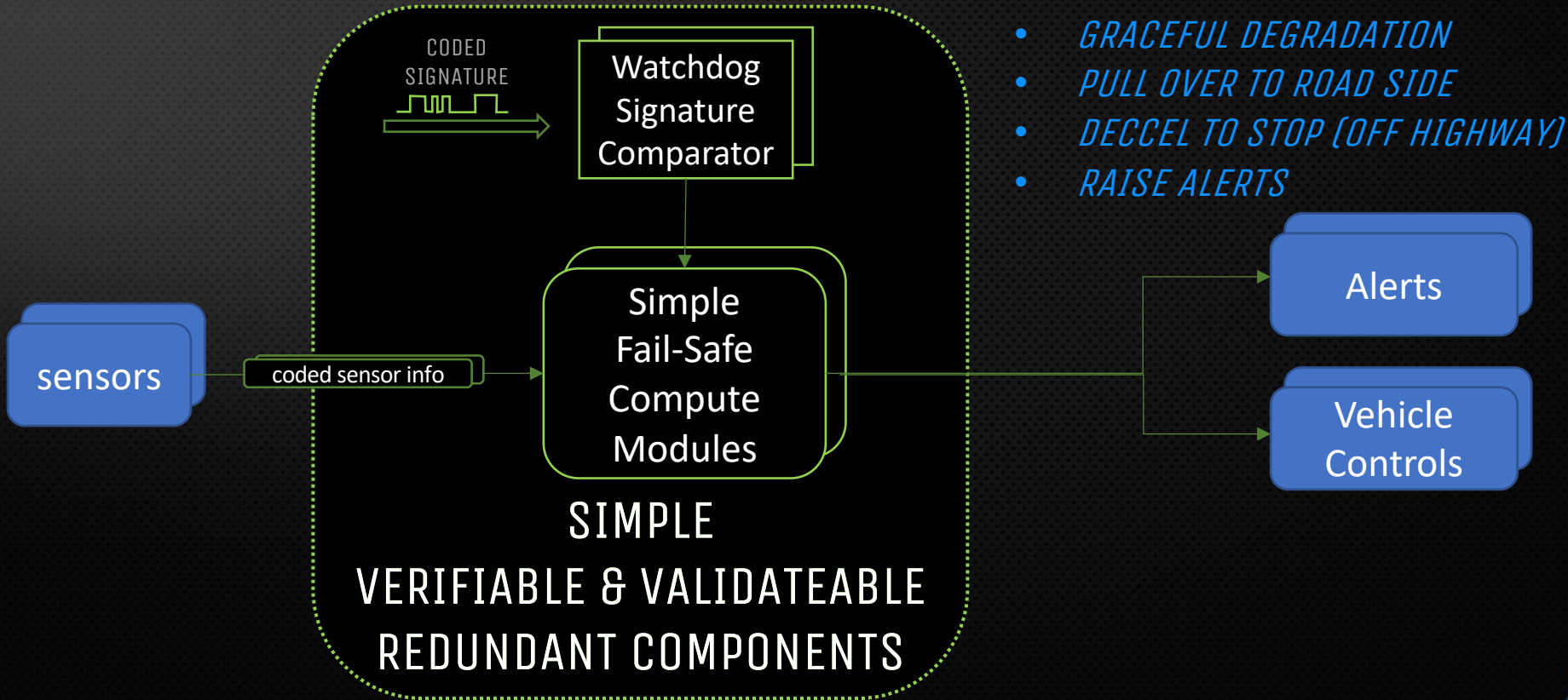
CODING SAFETY-CRITICAL INFORMATION



WATCHDOG SERVICE APPROACH



FAIL-SAFE WATCHDOG REACTION SERVICE



SUMMARY

- CURRENT CERTIFICATION METHODS “DATED” TO DEAL WITH FULL AUTONOMY COMPLEXITY
- HIGHER LEVEL PROGRAMMING LANGUAGES/METHODS (FROM IT) + MODERN FRAMEWORKS/RUNTIMES + AI/ML REQUIRED TO ADDRESS FULL AUTONOMY COMPLEXITY
- NEW APPROACHES NEEDED FOR SAFETY WITH FULL AUTONOMY
- SIMPLE VERIFIABLE/VALIDATEABLE WATCHDOG APPROACH
 - Provable and quantifiable safety
 - Bonus: Same approach to safety applies to security