# Why is it so hard to make self-driving cars?
## (Trustworthy Autonomous Systems)

Robert Stewart Distinguished Lecture
ISU-Department of Computer Science
April 1st , 2022

Joseph Sifakis
Verimag Laboratory

# Trustworthy Autonomous Systems – Main Characteristics

❑ Autonomous systems are distributed systems involving dynamically changing sets of agents, each pursuing specific goals, and coordinating so that their collective behavior satisfies given global properties.

- They are essential for achieving the Industrial IoT vision as they emerge from the needs to further automate existing organizations by gradually replacing human operators with autonomous agents.
- They are very different from game-playing robots or intelligent personal assistants.

❑ Autonomous systems are often critical and should exhibit "broad intelligence" by handling knowledge in order to

- Manage dynamically changing sets of possibly conflicting goals;
- Cope with uncertainty of complex, unpredictable cyber physical environments;
- Harmoniously collaborate with human agents e.g. "symbiotic" autonomy.

❑ Two different technical avenues both falling short of the autonomy challenge:

- traditional **model-based** critical systems engineering, successfully applied to aircraft and production systems, proves to be inadequate.
- industrial end-to-end **AI-enabled solutions** that fail to provide the required strong trustworthiness guarantees.

❑ The development of trustworthy autonomous systems is considered a bold step toward closing the gap between human and artificial intelligence

# Trustworthy Autonomous Systems – The Emblematic Case of Self-driving Cars

- Self-driving cars are a topical case emblematically illustrating the blockages on the path from automation to autonomy. Currently, new trends and practices emerge in autonomous systems engineering.

- In contrast to well-established critical systems engineering practice, self-driving car manufacturers

  - do not follow a “safety by design” concept adopting end-to-end ML-enabled solutions to overcome the technical difficulties implied by traditional model-based approaches,
  - in the absence of standards, they are allowed to apply self-certification by following guidelines (that are not in themselves legal requirements).
  - are allowed regular over the air updates critical software

- These trends are the subject of lively discussions.

  Many believe that it is necessary to break with traditional development techniques that are a barrier to the acceptance of new technologies.

  - Some show blunt realism that we should charge ahead and accept the risks as the benefits will be so great!
  - Others reject rigorous approaches as inherently inadequate for such complex systems and show blind faith in ad hoc solutions.
  - Finally, there are those who are wildly optimistic that we have the right tools and that full autonomy is only a matter of time.

# Autonomous systems – For a New Scientific and Engineering Foundation

| | Critical Systems Engineering is facing a huge gap, moving | | | | |
|---|---|---|---|---|---|
| FROM | Small size | Centralized | Automated | Predictable Envt | Elicitable Specs |
| TO | Complex | Decentralized | Autonomous | Unpredictable Envt | Non-elicitable Specs |

We need a new scientific and engineering foundation that cannot be obtained by simply combining existing results from ML, Autonomic computing, Adaptive systems, Autonomous Agent Systems and brings answers to the following problems:

1. Realizing the magnitude of the undertaking
   - Bridging the gap between Automation and Autonomy
   - What are the technical solutions for enhancing a system's autonomy and the implied difficulties?

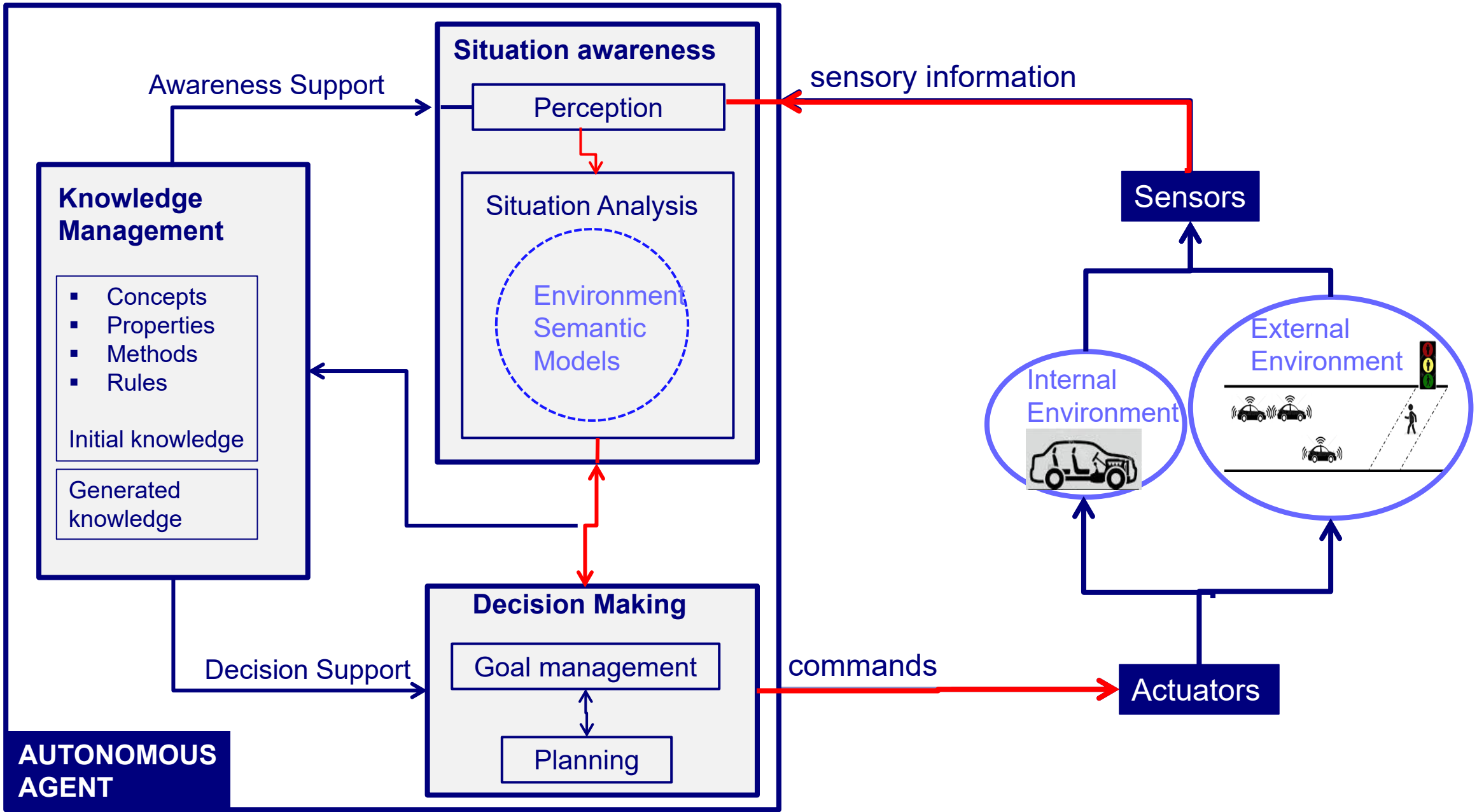
2. Trustworthy autonomous agent design
   - Investigate "hybrid approaches" seeking trade-offs between the rigor of model-based approaches and the efficiency of AI-based techniques.
   - Adequately address systems engineering issues for risk analysis and mitigation.

3. New theory for the global validation of autonomous systems based on simulation and testing.
   - Realistic and semantically aware simulation requires new modelling techniques
   - Develop new theory of testing relying an adequate coverage criteria

OVERVIEW

- The magnitude of the undertaking
  - Autonomy and autonomic complexity
  - Systems engineering issues
- Trustworthy agent design
  - Hierarchical control architecture
  - Design for dependability
- Global system validation
- The way ahead

# The Magnitude of the Undertaking – Autonomous Agent Architecture



# The Magnitude of the Undertaking – From Automation to Autonomy

| | SAE AUTONOMY LEVELS |
|---|---|
| Level 0 | No automation |
| Level 1 | Driver assistance required |
| | The driver still needs to maintain full situational awareness and control of the vehicle e.g. cruise control. |
| Level 2 | Partial automation options available |
| | Autopilot manages both speed and steering under certain conditions, e.g. highway driving. |
| Level 3 | Supervised Autonomy |
| | The car, rather than the driver, takes over actively monitoring the environment when the system is engaged. However, human drivers must be prepared to respond to a "request to intervene" |
| Level 4 | Geofence autonomy |
| | Self driving is supported only in limited areas or under special circumstances, like traffic jams |
| Level 5 | Full autonomy |
| | No human intervention is required e.g. a robotic taxi |

AUTOMATION (ADAS)

AUTONOMY

# The Magnitude of the Undertaking – From Automation to Autonomy (2)

The SAE autonomy hierarchy may lead to misinterpretations and confusion.

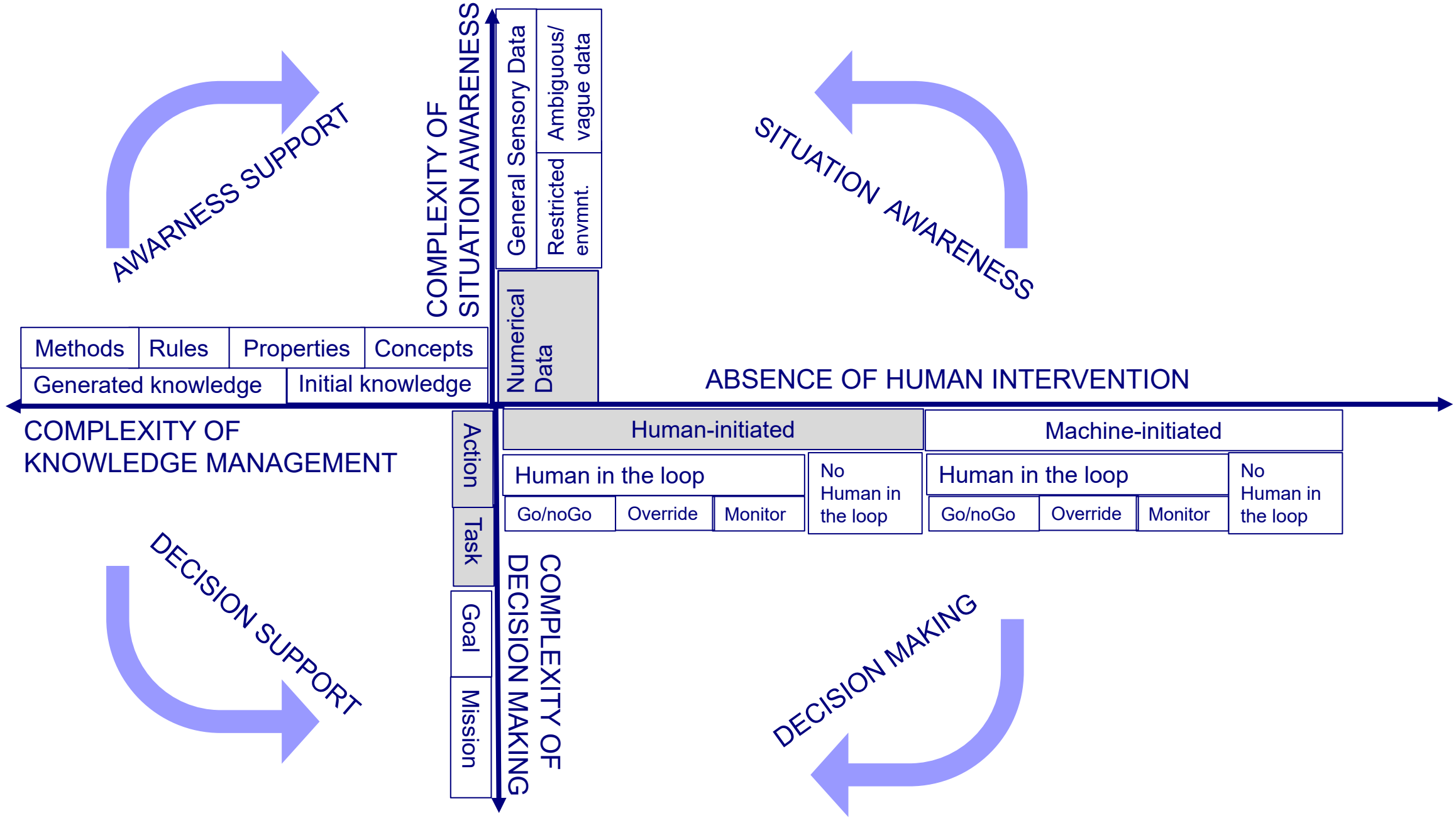
- It suggests that the transition from automated ADAS systems to self-driving systems can be progressive by climbing up levels – There is a big gap between levels 2 and 3

- Supervision of autonomous cars on autopilot prescribed by Level 3, turns out to be a hazardous idea - safe collaboration between autonomous systems and human agents goes much deeper than classical HMI.
  - When the autopilot proactively solicits human agent's intervention, the latter should have the adequate information and time to understand the situation and act adequately.
  - When the supervisor realizes that something goes wrong, disengaging the autopilot or overriding machine's decisions should result into safe situations controllable by humans.

We need protocols guaranteeing awareness of responsibility transfer from human to machines and vice versa.

- There is a big gap between level 4 and level 5
  - as most autonomic complexity factors result from non-reliability of the perception function and non-predictability of the external environment.
  - in geofenced environments the external environment is much more predictable because the number of possible interactions and configurations drastically decreases. Furthermore, for such environments it is possible to use extensive instrumentation to enhance perception quality.
  - For instance, truck platooning in freeways seems feasible in some near future.

# The Magnitude of the Undertaking – From Automation to Autonomy (3)



# The Magnitude of the Undertaking – Autonomic Complexity

- Complexity of situation awareness
  - Complexity of perception characterizes the difficulty to interpret stimuli (cope with ambiguity, vagueness) and to timely generate corresponding inputs for the agent environment model.
  - Complexity of uncertainty due to
    - situations involving imperfect or unknown information implying lack of predictability about the environment such as dynamic change caused by physical or human processes, rare events, critical events such as failures and attacks.
    - entirely new situations not anticipated at design time – requires self-learning and generation of goals
- Complexity of decision reflected in the complexity of the agent's decision process (goal management and planning) and impacted by the following factors:
  - type of goals e.g. safety, reachability, security, optimization of resources
  - multiplicity of goals, especially long/mid/short goals, potentially conflicting
  - complexity of the space of solutions to be explored for plan generation and lack of controllability

In addition to autonomic complexity, the construction of autonomous systems involves engineering problems not related to the intelligence of agents, characterized by their system complexity.

system complexity = component interactive complexity $\times$ architecture complexity

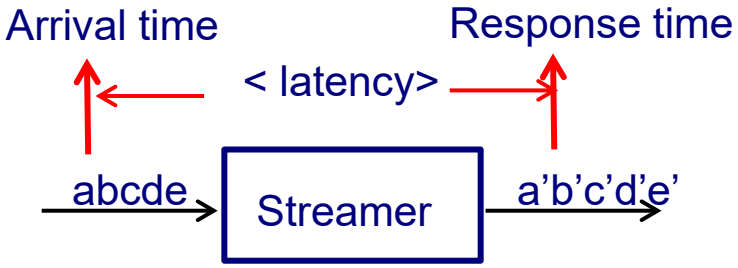
- The magnitude of the undertaking
  - Autonomy and autonomic complexity
  - Systems engineering issues
- Trustworthy agent design
  - Hierarchical control architecture
  - Design for dependability
- Global system validation
- The way ahead

OVERVIEW

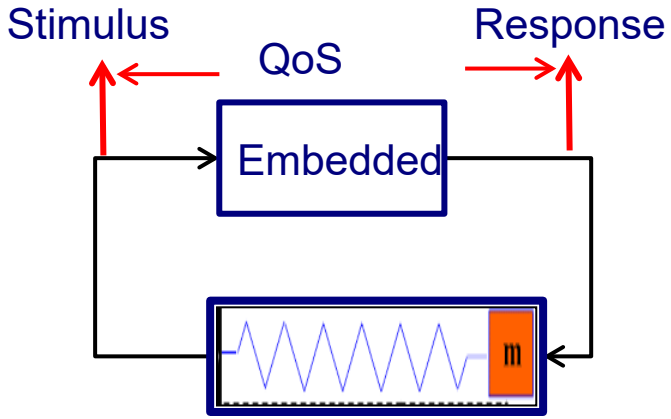
# The Magnitude of the Undertaking – Reactive Complexity

Reactive complexity of components
- characterizes the intricacy of the interaction between a component and its environment.
- Is independent from memory space complexity or time complexity (related to resources needed)

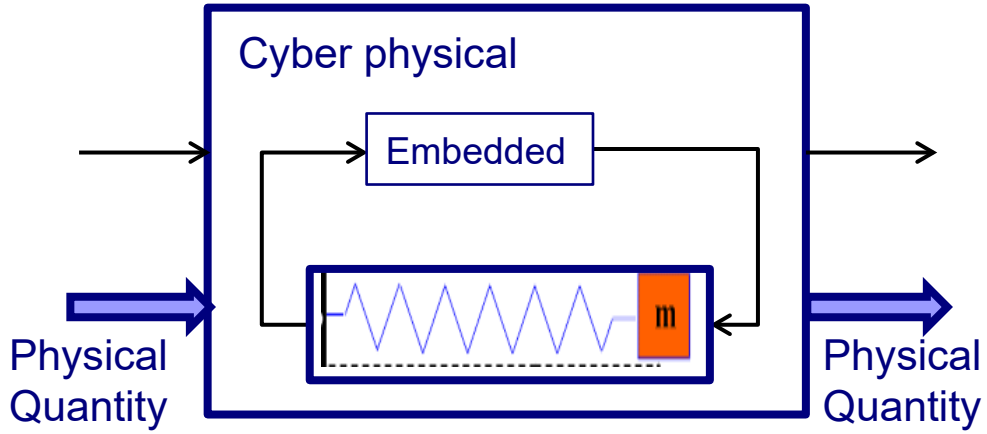

Transformational component
e.g. Intelligent Personal Assistant




Streaming component
e.g. Encoder, Signal processor



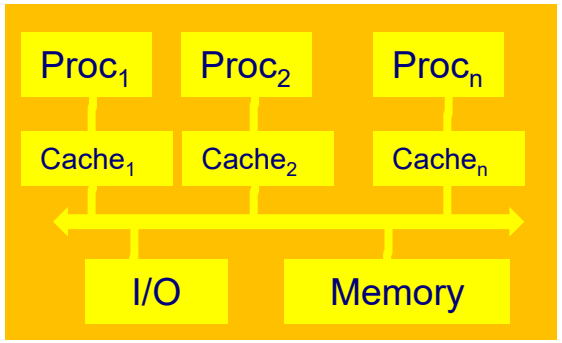

Embedded component
e.g. Flight controller



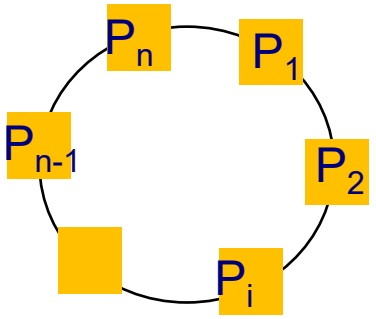

Cyber physical component e.g.
for building self-driving cars

# The Magnitude of the Undertaking – Architectural Complexity

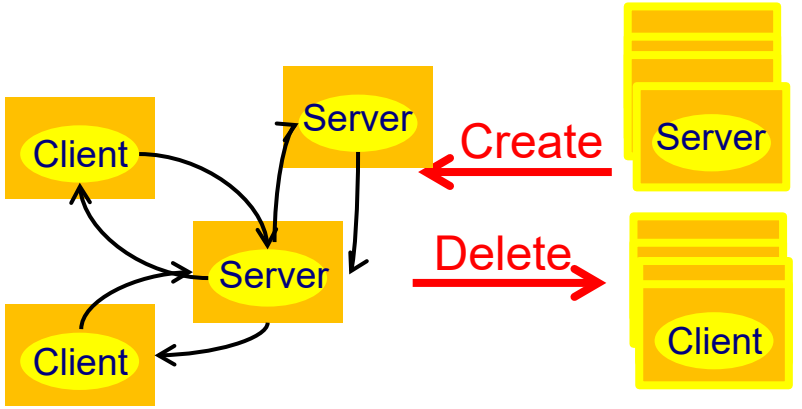
How much involved is the coordination between agents?



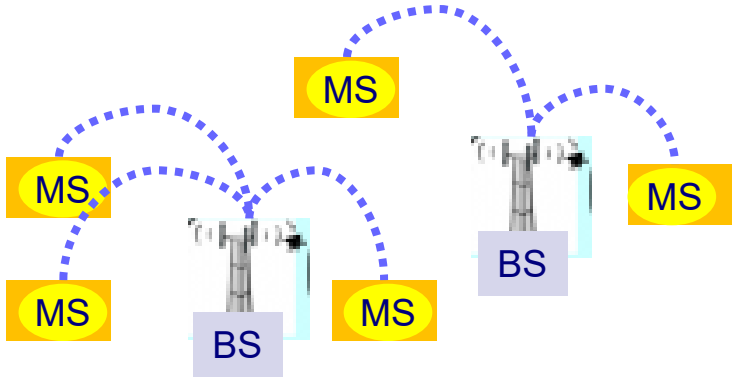

Static Architecture:
Multiprocessor system



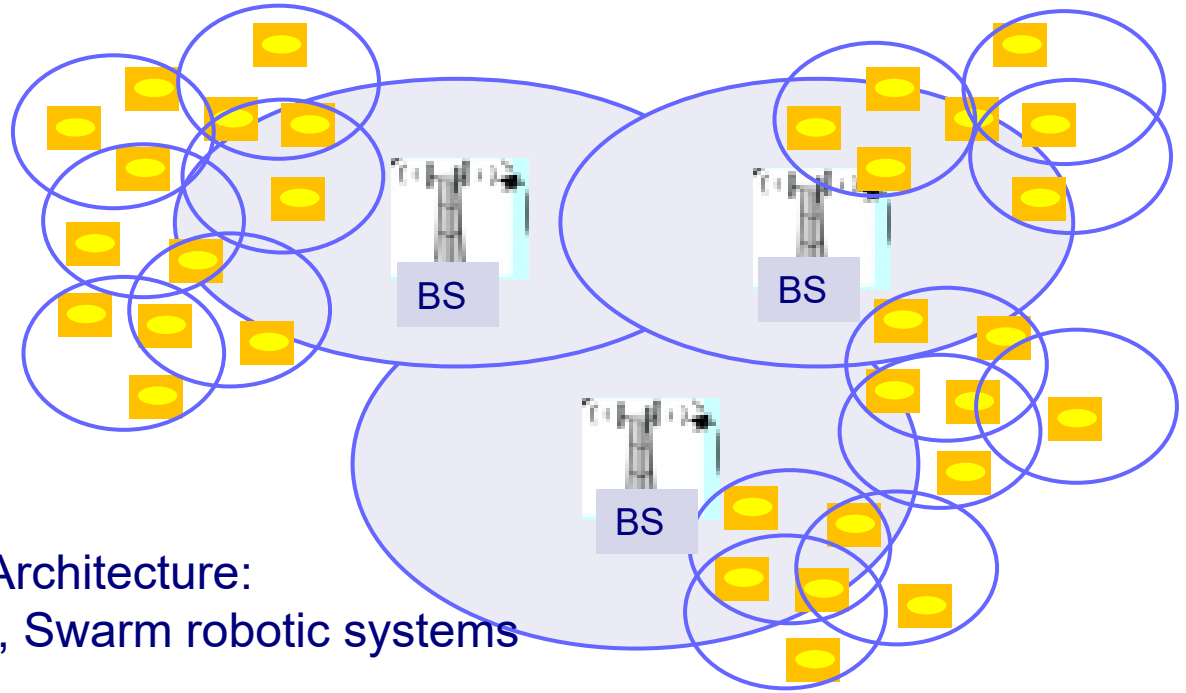

Parametric Architecture:
Ring architecture




Dynamic Architecture:
Distributed system

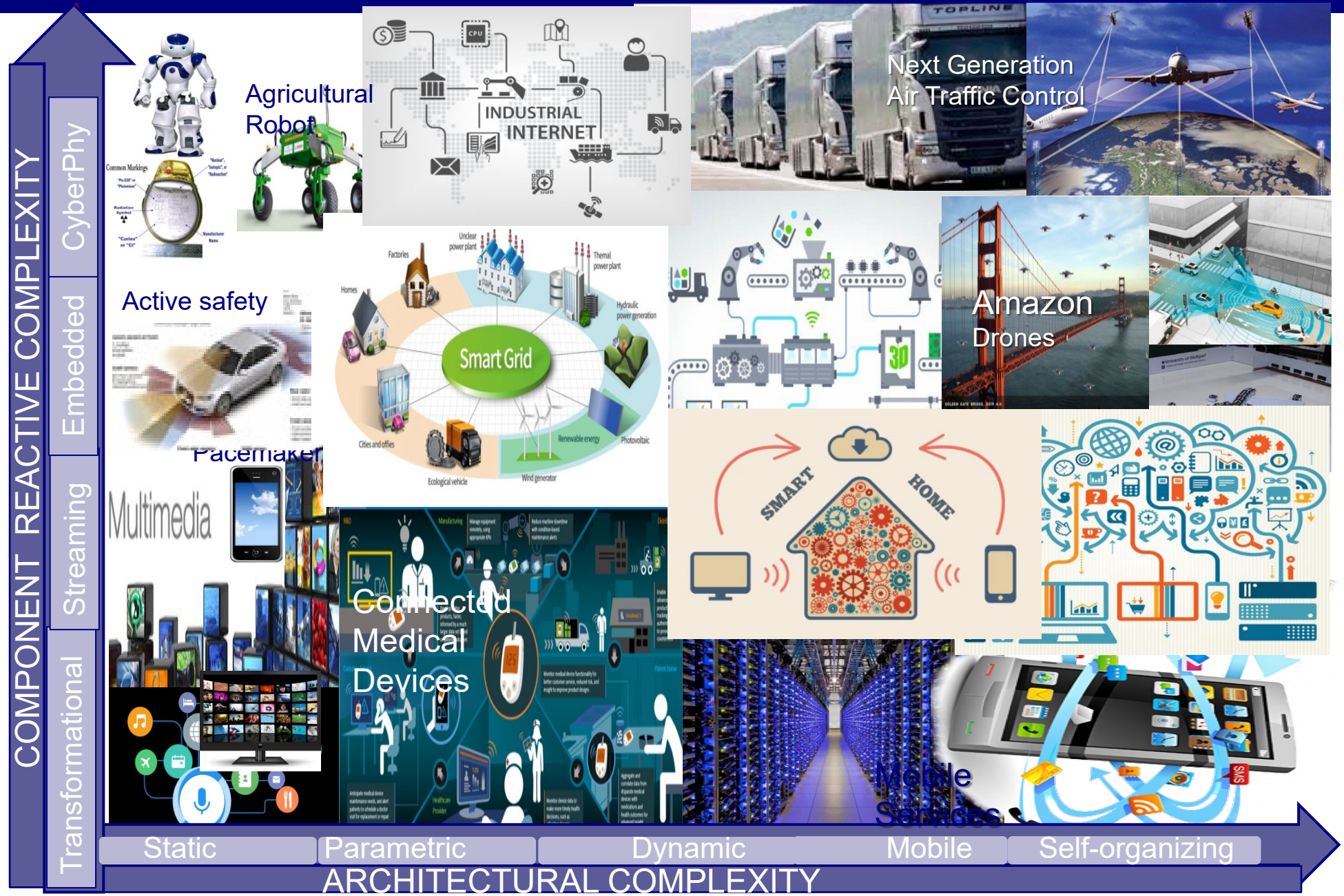



Mobile Architecture:
Mobile telecommunication system

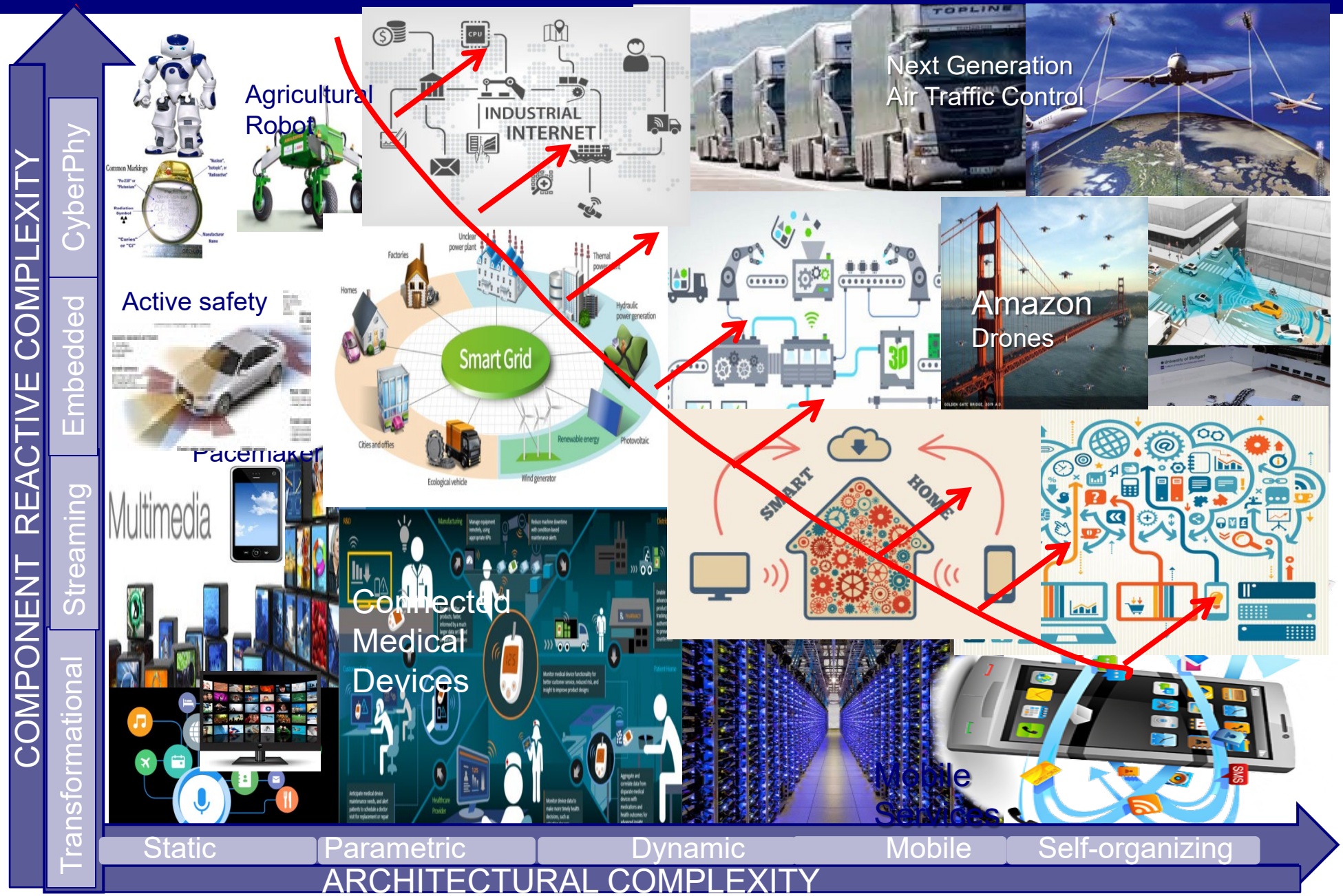



Self-organizing Architecture:
Self-driving cars, Swarm robotic systems

# The Magnitude of the Undertaking – System Complexity: Reactive × Architectural



# The Magnitude of the Undertaking – System Complexity: Reactive × Architectural



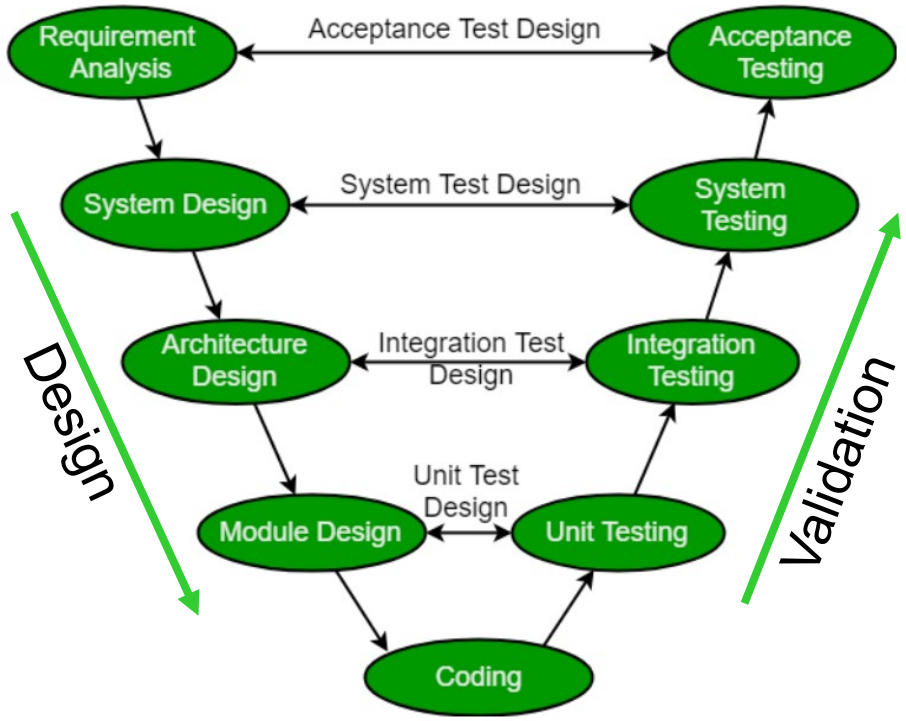
- The magnitude of the undertaking
  - Autonomy and autonomic complexity
  - Systems engineering issues
- Trustworthy agent design
  - Hierarchical control architecture
  - Design for dependability
- Global system validation
- The way ahead

OVERVIEW

# Agent Design – Critical Systems Engineering Limitations

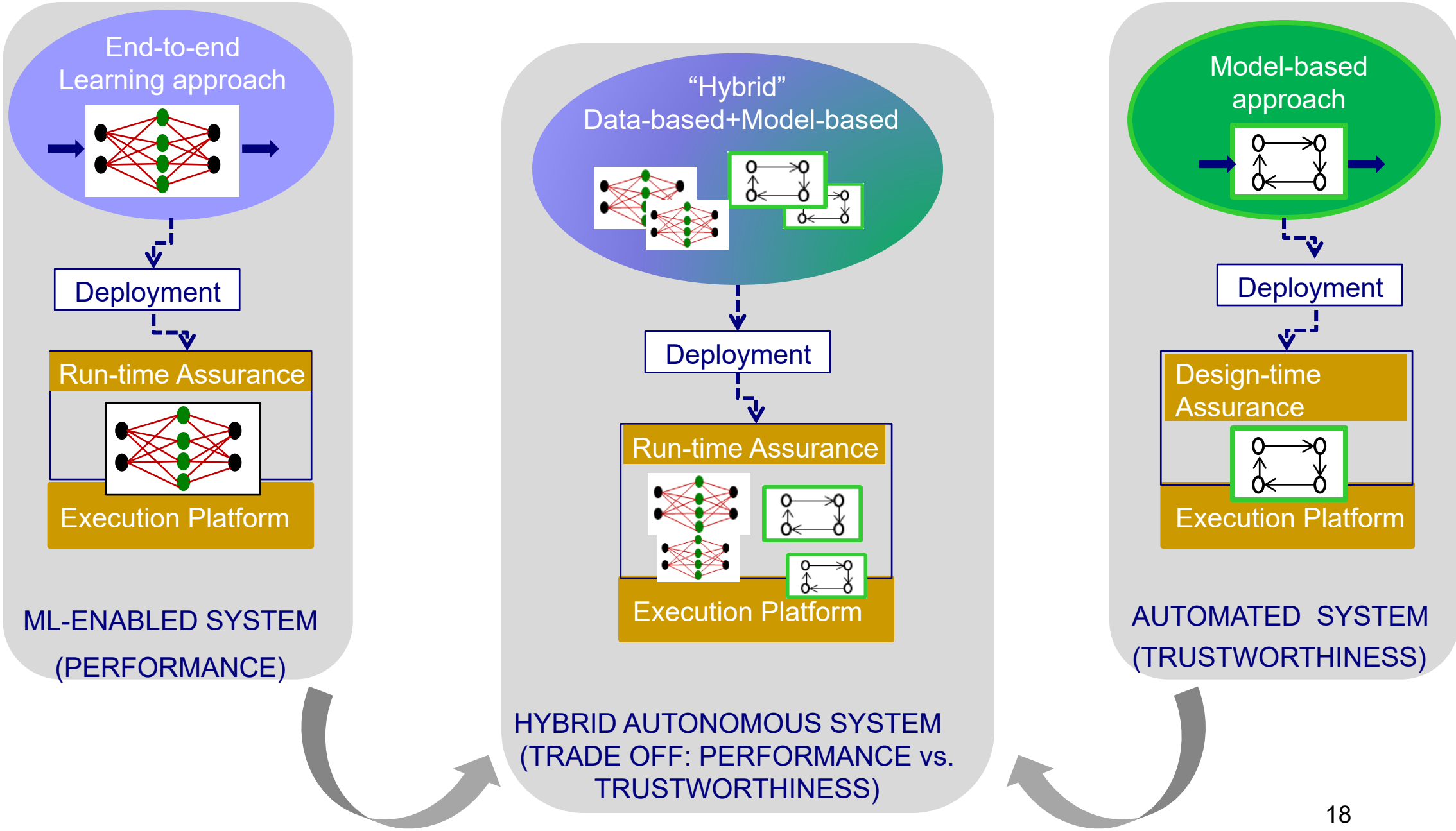
Critical systems design flows follow model-based prescriptive frameworks recommended by standards e.g. ISO26262

- Assume that system development is top-down and validation is bottom-up.
- Assume that all requirements are initially known, can be clearly formulated and understood.
- Consider that global system requirements can be broken down into requirements satisfied by system components.
- Focus on providing model-based conclusive evidence that the system is safe e.g.
  $10^{-9}$ failures per hour of flight

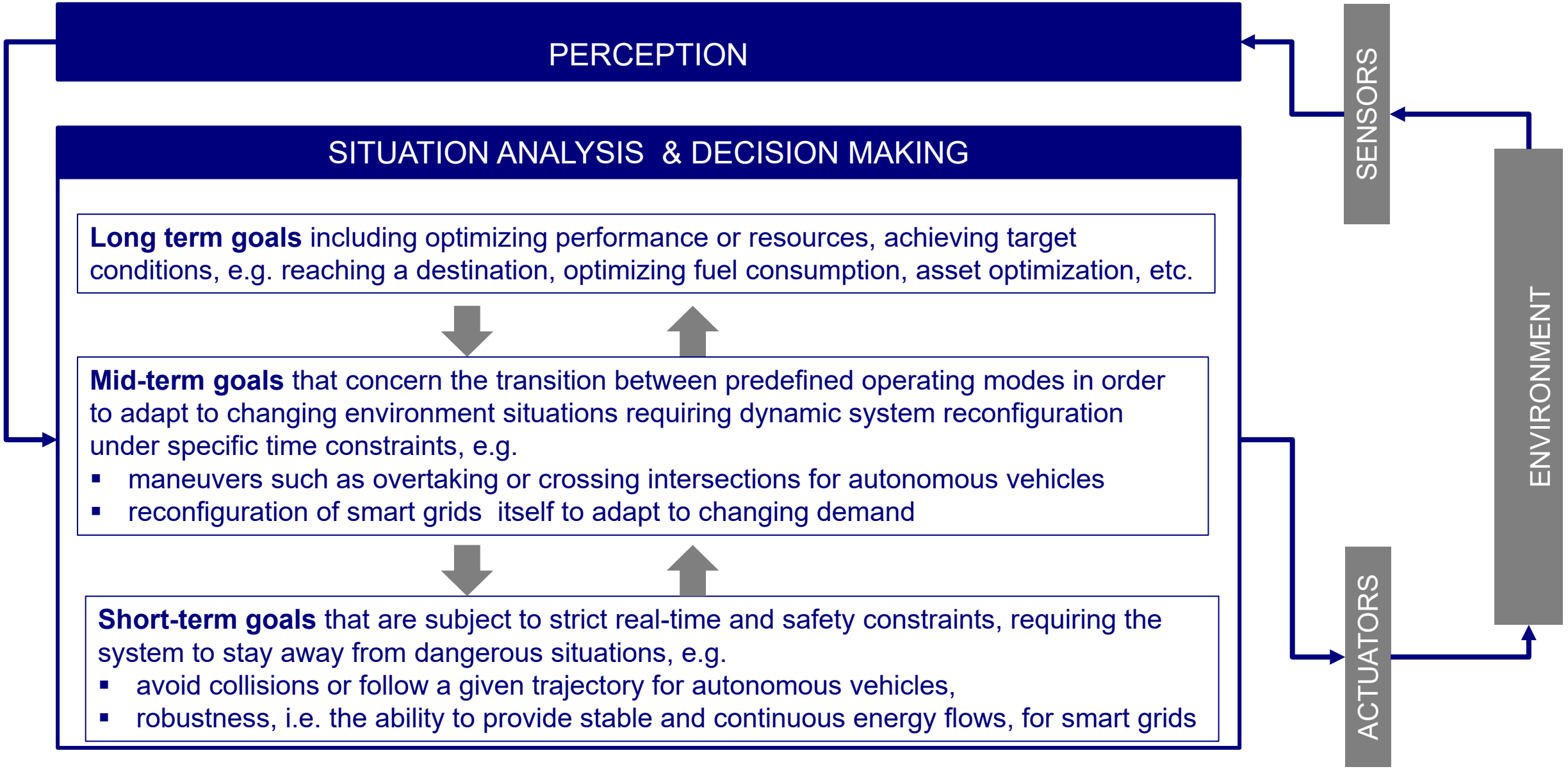



- The model-based paradigm is defeated by the overwhelming complexity and diversity of autonomous systems
- This explains the adoption by industry of end-to-end machine-learning-enabled techniques which however preclude conclusive safety guarantees

# Hybrid Agent Design –Taking the Best from Each



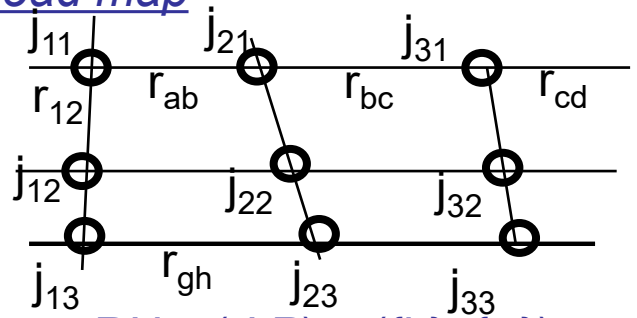
# Hybrid Agent Design – Hierarchical Control Principle



# Hybrid Agent Design – Hierarchical Semantic Model

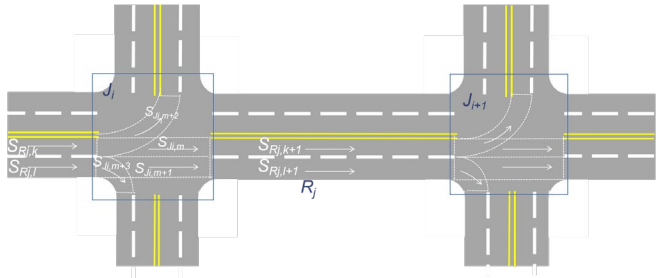
## Maps

### Road map



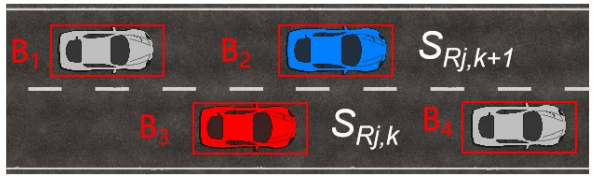

$RN = (J,R) = (\{j_k\}, \{r_s\})$

### Lane map



$SN = (S,E)$

### Synthesized local map

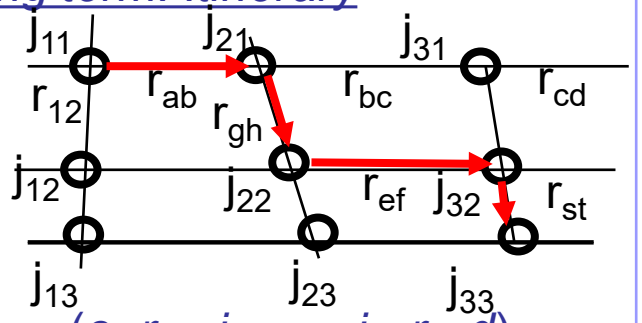



## Hierarchical Control

- **Mission Planning - Level 4**
- **Path Planning - Level 3**
- **Maneuver Planning & protocols - Level 2**
- **Trajectory Planning - Level 1**

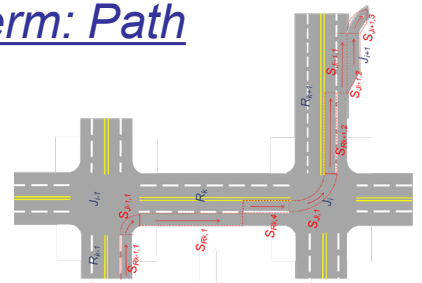
## Goals

### Long term: Itinerary




$(o, r_{ab}, j_{21}, \ldots j_k, r_k, d)$

### Mid-term: Path

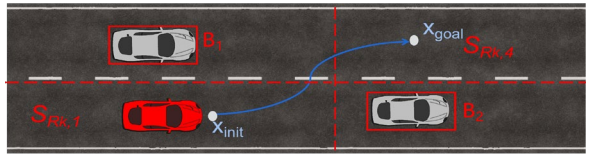


$(s_{Rk-1,1}, s_{Ji-1,1}, s_{Rk,1}, s_{Rk,4}, s_{Ji,1}, s_{Rk+1,2})$

### Maneuver sequences

$(lane\_change, s_{Rk,1}, s_{Rk,4})$

### Short-term: Trajectory



- The magnitude of the undertaking
  - Autonomy and autonomic complexity
  - Systems engineering issues
- Trustworthy agent design
  - Hierarchical control architecture
  - Design for dependability
- Global system validation
- The way ahead

OVERVIEW

# Agent Design for Dependability – Trends

- Dependability Engineering

  - has historically emerged as a sub-discipline of Physical Systems Engineering is often presented as something apart from the other system design activities.
  - has been the object of a huge number of works – nonetheless only very few address the issue of their integration into rigorous design flows.
  - initially has focused on building small centralized safety critical systems with static architectures of guaranteed dependability, characterized by a set of quantitative attributes such as reliability, availability, maintainability, , serviceability, manageability, etc.

  - Later, specific techniques have been developed for security critical systems e.g. smart cards, Scada systems, networks.

- New trends with the advent of IIoT and autonomous systems, in particular,

  - both safety and security should be considered together as they can impact each other (S&S co-design) , e.g. a security vulnerability in a connected car could be used to disable braking while driving, resulting in potential loss of control and crash;
  - traditional techniques are defeated by complexity and extensive use of non-explainable AI components;
  - design for dependability should be better integrated into system development methodologies – in particular to determine global tradeoffs between functional safety/security and safety/security risks.

# Agent Design for Dependability – The Flow

RISK ANALYSIS

Identify system hazards and estimate their likelihood in terms scenarios involving risk causes and effects.

RISK MITIGATION

Based on Risk Analysis, design and implement for each hazard corresponding mechanisms for

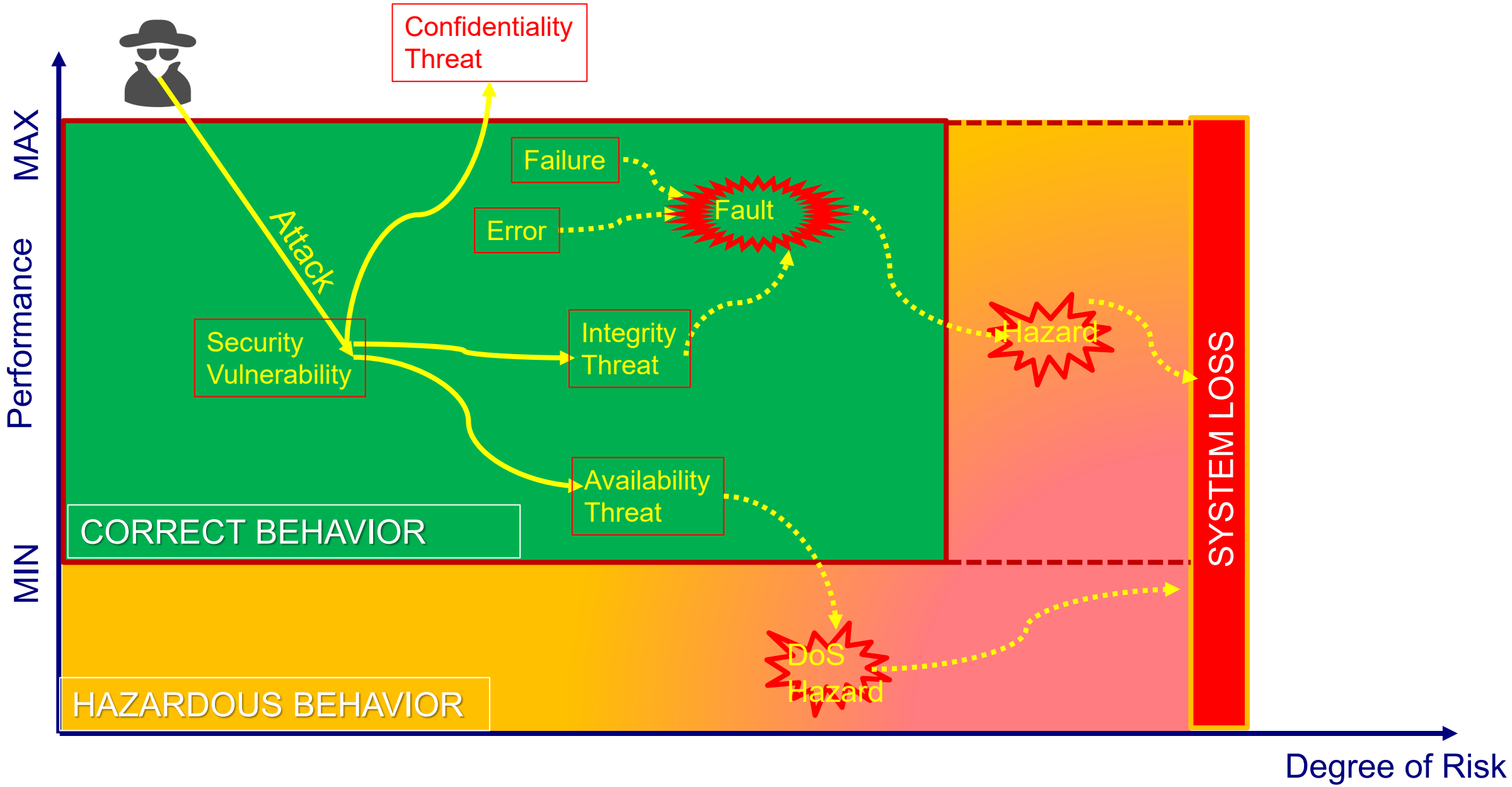
- Detection
- Isolation
- Recovery

RISK EVALUATION

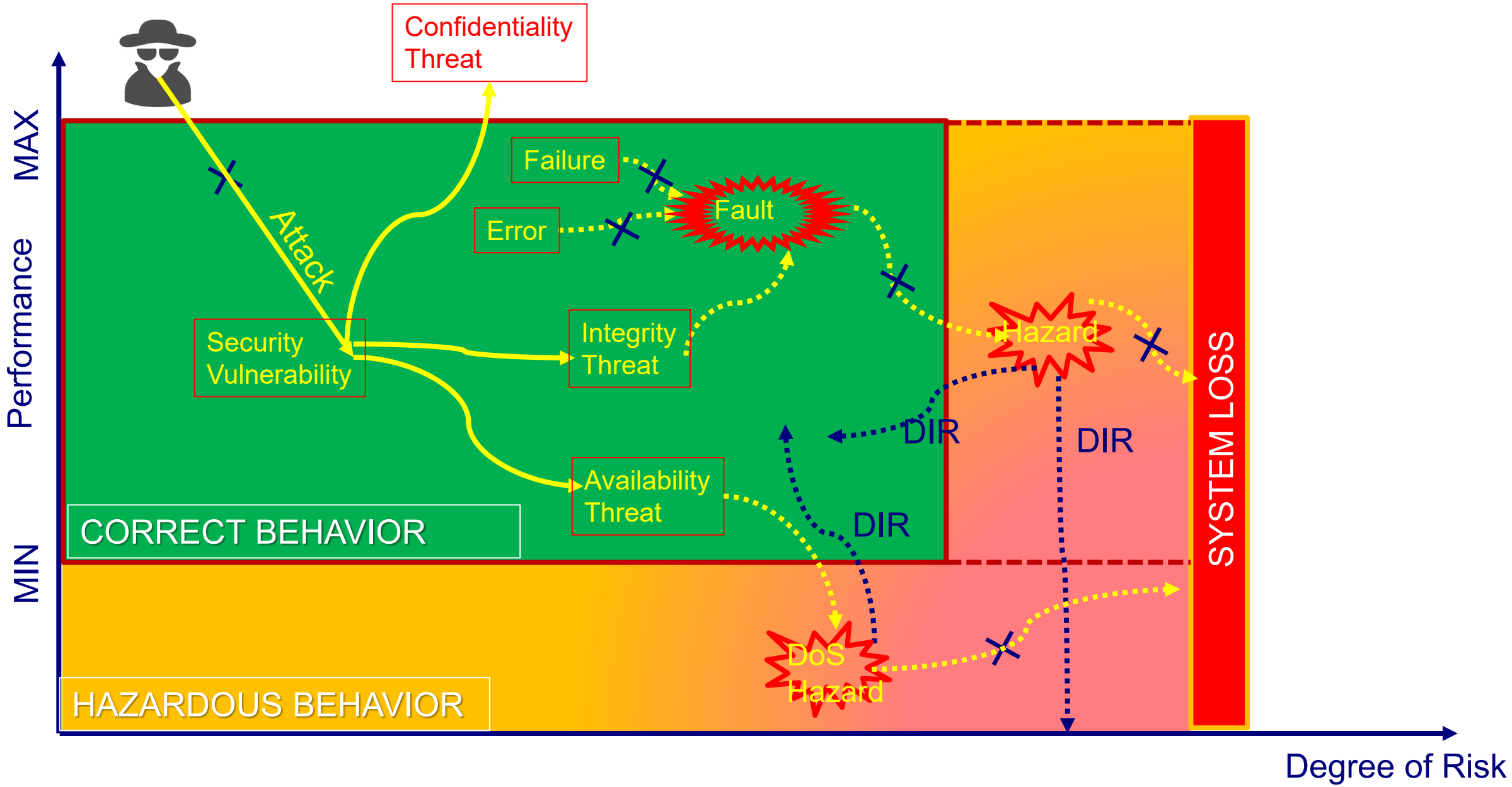
Evaluate the resilience of the system to hazards and in particular the likelihood that it meets a set of properties characterizing its correctness.

- Design for dependability is probably the hardest task in system development.
  - it requires a deep and global knowledge of the system's functional behavior, its implementation and above all its interaction with the external environment;
  - It requires good common sense engineering skills to estimate 1) what can go wrong? and 2) how frequently it can happen using all available evidence, primarily past experience and expert judgment.

# Agent Design for Dependability – Risk Analysis



# Agent Design for Dependability – Fault/Attack Detection, Isolation, Recovery (DIR)



# Agent Design for Dependability – Fault/Attack Detection, Isolation, Recovery (DIR)

Hazard Detection: achieved
- on line without disturbing service delivery e.g. security log analysis, redundancy, error detecting codes;
- off line by applying testing and diagnostics techniques, batch security log analysis.

Hazard Isolation: various techniques such as
- use of partitioned architectures such that the memory and processing time of a partition is not affected by another faulty partition;
- firewalls, cryptography, privileged access management.

Hazard Recovery:
- using fault-tolerance mechanisms that mask the detected fault e.g. TMR;
- online using roll-back techniques to a trustworthy saved state that existed prior to the occurrence of the hazard (caused by SW error or attack);
- online using roll-forward techniques to a to a trustworthy new state (caused by SW error or attack);
- online through reconfiguration that preserves some minimal service leaving out the affected components ;
- offline with system re-initialization after checking that the detected risk causes are not present anymore.

**Traditional DIR techniques rely on a detailed case by case mitigation of risks that cannot be applied to autonomous systems with unpredictable environments.**

# Agent Design for Dependability – Failure Typology for Light Vehicles

Existing techniques successfully applied to conventional critical systems e.g. aircraft, are challenged by the perspective that self-driving cars should cope with the overwhelming complexity of these risk factors !!

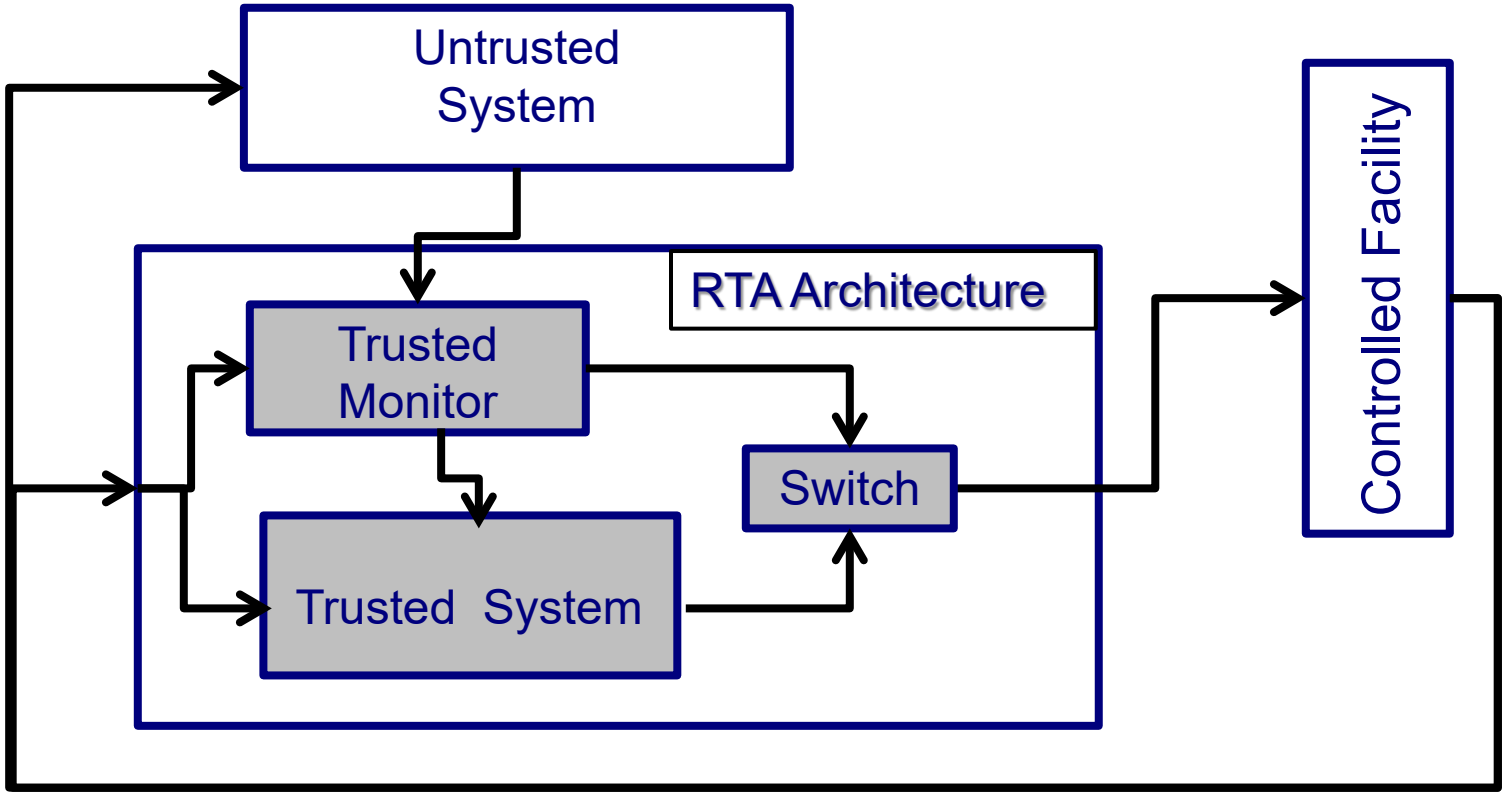
| | |
|---|---|
| 1 | Vehicle Failure |
| 2 | Control Loss With Prior Vehicle Action |
| 3 | Control Loss Without Prior Vehicle Action |
| 4 | Running Red Light |
| 5 | Running Stop Sign |
| 6 | Road Edge Departure With Prior Vehicle Maneuver |
| 7 | Road Edge Departure Without Prior Vehicle Maneuver |
| 8 | Road Edge Departure While Backing Up |
| 9 | Animal Crash With Prior Vehicle Maneuver |
| 10 | Animal Crash Without Prior Vehicle Maneuver |
| 11 | Pedestrian Crash With Prior Vehicle Maneuver |
| 12 | Pedestrian Crash Without Prior Vehicle Maneuver |
| 13 | Pedalcyclist Crash With Prior Vehicle Maneuver |
| 14 | Pedalcyclist Crash Without Prior Vehicle Maneuver |
| 15 | Backing Up Into Another Vehicle |
| 16 | Vehicle(s) Turning – Same Direction |
| 17 | Vehicle(s) Parking – Same Direction |
| 18 | Vehicle(s) Changing Lanes – Same Direction |
| 19 | Vehicle(s) Drifting – Same Direction |
| 20 | Vehicle(s) Making a Maneuver – Opposite Direction |
| 23 | Lead Vehicle Accelerating |
| 24 | Lead Vehicle Moving at Lower Constant Speed |
| 25 | Lead Vehicle Decelerating |
| 26 | Lead Vehicle Stopped |
| 27 | Left Turn Across Path From Opposite Directions at Signalized Junctions |
| 28 | Vehicle Turning Right at Signalized Junctions |
| 29 | Left Turn Across Path From Opposite Directions at Non-Signalized Junctions |
| 30 | Straight Crossing Paths at Non-Signalized Junctions |
| 31 | Vehicle(s) Turning at Non-Signalized Junctions |
| 32 | Evasive Action With Prior Vehicle Maneuver |
| 33 | Evasive Action Without Prior Vehicle Maneuver |
| 34 | Non-Collision Incident |
| 35 | Object Crash With Prior Vehicle Maneuver |
| 36 | Object Crash Without Prior Vehicle Maneuver |
| 37 | Other |

Pre-crash failure typology covering 99.4% of light-vehicle crashes for 5,942,000 cases.
Source: Pre-Crash Scenario Typology for Crash Avoidance Research, DOT HS 810 767, April 2017.

# Agent Design for Dependability – Run-time Assurance Techniques (RTA)

- Allow risk mitigation based on a set of critical properties to be preserved without making a detailed risk analysis.
- Mark the shift from correctness at design time to correctness at runtime.

- Consist in running in parallel the **Untrusted System** and a Run-time Assurance System composed of a **Trusted Monitor**, a **Trusted System**, and a **Switch:**
  - The **Trusted Monitor** detects hazards - discrepancies from the nominal behavior (set of critical properties);
  - The **Trusted System** can cope with hazards detected by the **Trusted Monitor** with possible performance degradation;
  - The **Switch** provides the output of the **Untrusted System** as long as no violation of critical properties is detected..

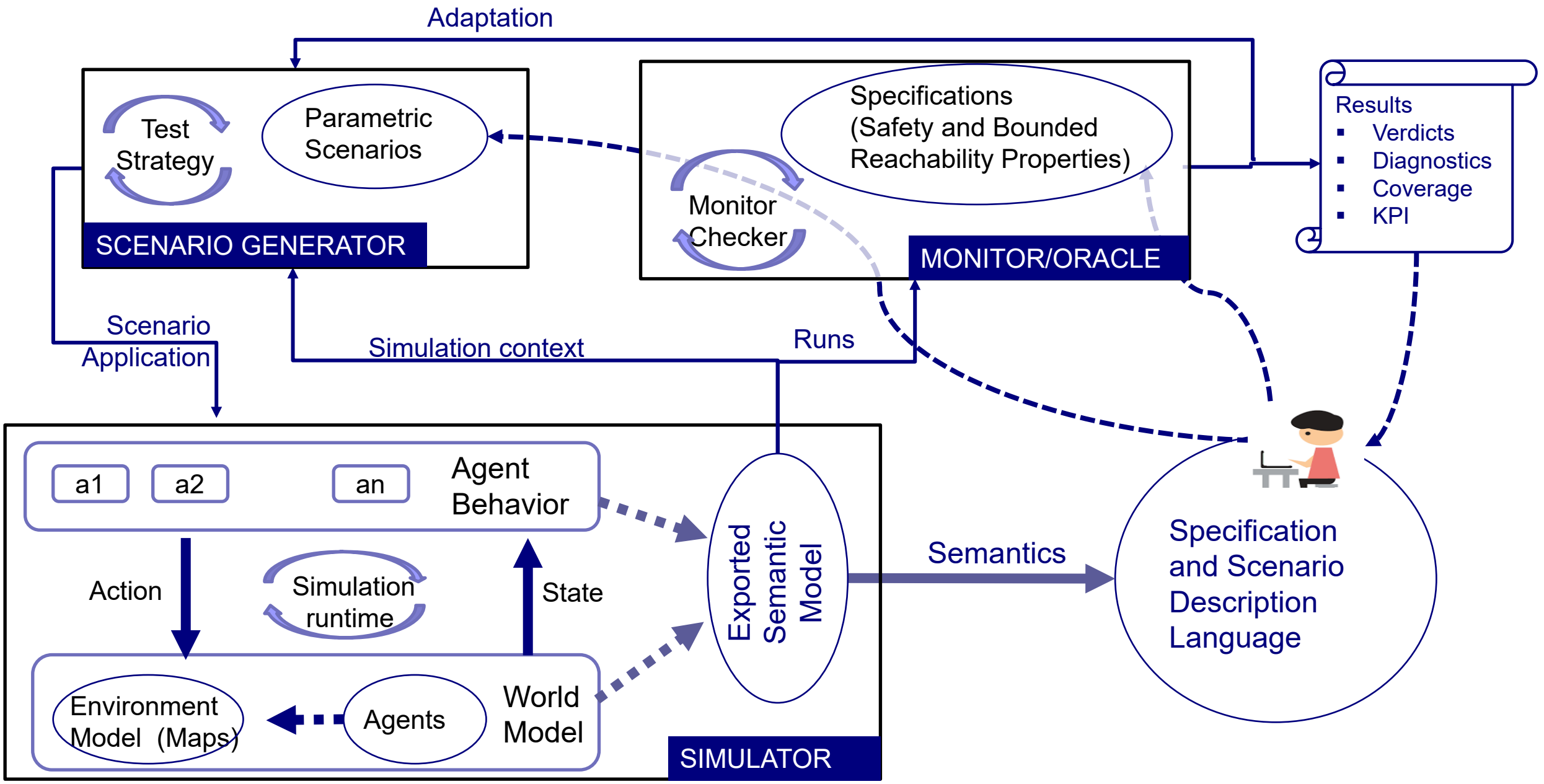



- RTA improves the SIMPLEX architecture paradigm as it allows continuity of service.
- RTA is considered as a solution to enhance safety of car autopilots.
- Nonetheless, a key issue is **smoothness** and **timeliness** of the taking over transitions.

- The magnitude of the undertaking
  - Autonomy and autonomic complexity
  - Systems engineering issues
- Trustworthy agent design
  - Hierarchical control architecture
  - Design for dependability
- Global system validation
- The way ahead

OVERVIEW

# Global System Validation – The Big Picture



# Global System Validation – Simulation Key Issues

- Whatever design approach is taken, simulation is of paramount importance for validation – and raises a large variety of problems from purely technical to theoretical ones.
- Not only the appearance should be realistic but also it should be real: the execution mechanism should rely on a semantic model of the environment consistent with laws of Geometry and Physics.
- Note that realism and consistency with reality are hard to reconcile - simulation environments built on top of game engines lack semantic awareness.

1. Realism: agent behavior and environment look real in a way that is accurate or true to life.
2. Expressiveness: supports rigorous modeling language e.g. DSL for
   - component-based description of mobile agents and their dynamic coordination;
   - modelling of physical environment in which the agents operate (maps).
3. Semantic awareness: the simulated system dynamics is rooted in transition system semantics.
   - Notion of state allowing repeatability of experiments.
   - Distinguishing between controllable and uncontrollable actions.
   - Multiscale multigrain modeling of time scales and of their correlation with space scales.
4. Performance: run-time infrastructure federating simulation engines e.g. HLA, FMI.

# Global System Validation – Gaps in the State of the Art

- Validation should rely on model-based criteria defined on an implicit or an explicit system model.
  - Superficial quantitative criteria such as simulation hours, miles travelled, do not provide sufficient evidence of trustworthiness.
  - Any technically sound safety evaluation should be model-based providing evidence that simulation covers a good deal of the many and diverse situations specified by the system properties to be validated e.g. for self-driving cars, different types of roads, traffic conditions, weather conditions.

- Property specification languages supporting genericity (types of objects) and parametricity (quantification over domains) e.g. first or higher order temporal logics and associated runtime verification techniques.

- Scenario description languages for the controlled simulation of agents so as to explore situations based on
  - Coverage criteria measuring the degree to which relevant system configurations have been explored, as for structural testing of software systems;
  - Functional criteria to explore/detect corner cases and high risk situations, exactly as for functional testing software systems;
  - Metamorphic relations that define similarity relations between scenarios used by the Scenario Generator to cope with complexity of their space – similar scenarios should produce close enough responses;
  - Verdicts and diagnostics about the relationship between failures and various risk factors e.g. for self-driving cars, the road structure, congestion level, weather and violations of traffic regulations.

- The magnitude of the undertaking
  - Autonomy and autonomic complexity
  - Systems engineering issues
- Trustworthy agent design
  - Hierarchical control architecture
  - Design for dependability
- Global system validation
- The way ahead

OVERVIEW

# The Way Ahead – Matching Human Situation Awareness

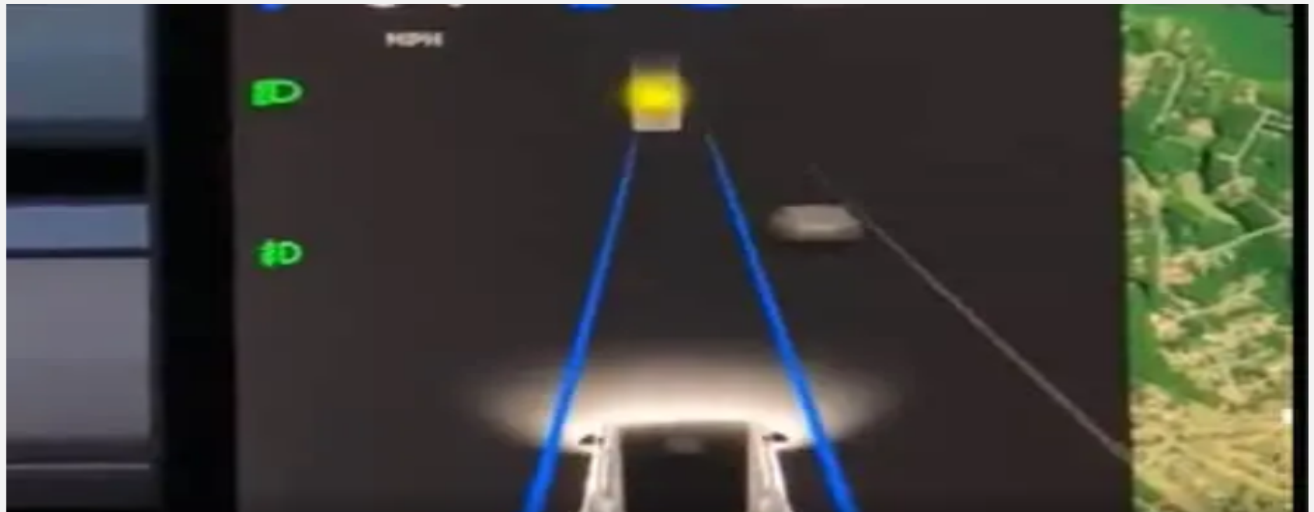
WIRED BACKCHANNEL BUSINESS CULTURE GEAR IDEAS SCIENCE SECURITY MY ACCOUNT GIVE A GIFT

ANDY GREENBERG SECURITY 10.11.2020 04:00 PM

## Split-Second 'Phantom' Images Can Fool Tesla's Autopilot

**Researchers found they could stop a Tesla by flashing a few frames of a stop sign for less than half a second on an internet-connected billboard.**

## Tesla's Autopilot Feature Mistakes Moon for Yellow Traffic Light, Watch Video



In a viral video shared by Twitter user Jordon Nelson, the autopilot system of a Tesla car can be seen confusing the yellow moon in the sky with a yellow traffic light.

- TRENDING DESK
- LAST UPDATED: JULY 26, 2021, 17:02 IST
- FOLLOW US ON: Facebook Twitter Instagram Telegram Google News

BUZZ STAFF

# The Way Ahead – Matching Human Situation Awareness

- To match human-level performance, systems should be able to deal with common sense knowledge.
- Human mind is equipped with a semantic model of the world:
  - a vast network of knowledge progressively built and automatically updated throughout life by learning and reasoning, and involving concepts, cognition rules and patterns;
  - used to interpret sensory information and natural language.
- Human understanding combines:
  - bottom-up reasoning from sensor level to the semantic model of the mind;
  - and top-down reasoning from the semantic model to perception.

- The challenge is to develop “self-learning systems” able to build progressively semantic models of their environment linking concrete and symbolic knowledge, combining learning and reasoning techniques.

- This is a hard problem as evidenced by the little progress in semantic analysis of natural languages so far.

# The Way Ahead – Regulations and Ethical Issues

- Acceptance of autonomous systems will depend on our decisions about when to trust them or not influenced by the following factors:
  - our ability to set standards and regulations based on sound and transparent evaluation criteria;
  - the willingness of authorities to exercise effective control for the protection of users;
  - social consciousness - public opinion is more unforgiving of system failures than of human errors!

- Trustworthiness is a technical concept that also has a subjective and social dimension.
  - In modern societies, institutions contribute directly or indirectly to shaping public perceptions of what is true, right, safe, etc.
  - So far certification of critical systems has mostly remained the prerogative of independent agencies e.g. FDA, FAA, NHTSA, according to well-founded standards requiring conclusive evidence based on models.

- Shall we accept that critical decision-making processes rely on machine-generated knowledge that allow predictability without understanding?
  - The threat is not that computer intelligence surpasses human intelligence and that computers could take over human societies by “plotting”.
  - The real danger comes from granting decision-making power to autonomous systems without rigorous and strictly founded guarantees under the pressure of economic interests and in the name of a misunderstood performance benefit.

# The Way Ahead – For a New Scientific and Engineering Foundation

- Strong techno-economic trends challenge traditional methods and practices, and require new development methodologies for autonomous systems that incorporates the trends:
  - Development cannot be entirely model-based due to the diversity and complexity of autonomous systems;
  - Break with the idea of correctness at design time e.g. $10^{-9}$ failures/hour - we must move to runtime assurance techniques.

- Hybrid design leveraging on a solid body of knowledge for safe and efficient decision making.
  - Building trusted systems from untrusted components – Non-explainable AI will remain an open problem!
  - Linking symbolic and non-symbolic knowledge e.g. sensory information and models of the environment.

- Global system validation is achievable only through simulation and testing.
  - Realistic and semantically sound modeling becomes of paramount importance for validation.
  - Theory and methodology inspired by the development and validation of scientific knowledge.

- There is a big gap between automated and autonomous systems – the transition cannot be progressive, for instance, ADAS cannot gradually evolve into self-driving systems!!

- Nonetheless, autonomic complexity drastically scales down for enhanced situation awareness (perception) and environment predictability, e.g. “geofence autonomy”.

- To reach the full autonomy vision we need to develop a new scientific and engineering foundation. And this will take some time.

*“**Intelligence** is not what you know*

*it is what you use when you don't know what to do.”*

*Jean Piaget, Swiss psychologist, 1896-1980*

# Thank you