

Assured Autonomy Survey

Other titles in Foundations and Trends® in Privacy and Security

Hardware Platform Security for Mobile Devices

Lachlan J. Gunn, N. Asokan, Jan-Erik Ekberg, Hans Liljestrand, Vijayanand Nayani and Thomas Nyman

ISBN: 978-1-68083-976-0

Cloud Computing Security: Foundations and Research Directions

Anrin Chakraborti, Reza Curtmola, Jonathan Katz, Jason Nieh, Ahmad-Reza Sadeghi, Radu Sion and Yinqian Zhang

ISBN: 978-1-68083-958-6

Expressing Information Flow Properties

Elisavet Kozyri, Stephen Chong and Andrew C. Myers

ISBN: 978-1-68083-936-4

Accountability in Computing: Concepts and Mechanisms

Joan Feigenbaum, Aaron D. Jaggar and Rebecca N. Wright

ISBN: 978-1-68083-784-1

A Pragmatic Introduction to Secure Multi-Party Computation

David Evans, Vladimir Kolesnikov and Mike Rosulek

ISBN: 978-1-68083-508-3

Assured Autonomy Survey

Christopher Rouff

Johns Hopkins University Applied Physics Laboratory
christopher.rouff@jhuapl.edu

Lanier Watkins

Johns Hopkins University Applied Physics Laboratory
lanier.watkins@jhuapl.edu

now

the essence of knowledge

Boston — Delft



Documents

Document Sharing Hub

Foundations and Trends® in Privacy and Security

Published, sold and distributed by:

now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:

now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

C. Rouff and L. Watkins. *Assured Autonomy Survey*. Foundations and Trends® in Privacy and Security, vol. 4, no. 1, pp. 1–116, 2022.

ISBN: 978-1-63828-039-2

© 2022 C. Rouff and L. Watkins

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording or otherwise, without prior written permission of the publishers.

Photocopying. In the USA: This journal is registered at the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923. Authorization to photocopy items for internal or personal use, or the internal or personal use of specific clients, is granted by now Publishers Inc for users registered with the Copyright Clearance Center (CCC). The 'services' for users can be found on the internet at: www.copyright.com

For those organizations that have been granted a photocopy license, a separate system of payment has been arranged. Authorization does not extend to other kinds of copying, such as that for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale. In the rest of the world: Permission to photocopy must be obtained from the copyright owner. Please apply to now Publishers Inc., PO Box 1024, Hanover, MA 02339, USA; Tel. +1 781 871 0245; www.nowpublishers.com; sales@nowpublishers.com

now Publishers Inc. has an exclusive license to publish this material worldwide. Permission to use this content must be obtained from the copyright license holder. Please apply to now Publishers, PO Box 179, 2600 AD Delft, The Netherlands, www.nowpublishers.com; e-mail: sales@nowpublishers.com

Foundations and Trends® in Privacy and Security

Volume 4, Issue 1, 2022

Editorial Board

Editors-in-Chief

Anupam Datta

Carnegie Mellon University, USA

Jeannette Wing

Columbia University, USA

Editors

Martín Abadi

*Google and University of California,
Santa Cruz*

Michael Backes

Saarland University

Dan Boneh

Stanford University, USA

Véronique Cortier

LORIA, CNRS, France

Lorrie Cranor

Carnegie Mellon University

Cédric Fournet

Microsoft Research

Virgil Gligor

Carnegie Mellon University

Jean-Pierre Hubaux

EPFL

Deirdre Mulligan

University of California, Berkeley

Andrew Myers

Cornell University

Helen Nissenbaum

New York University

Michael Reiter

University of North Carolina

Shankar Sastry

University of California, Berkeley

Dawn Song

University of California, Berkeley

Daniel Weitzner

Massachusetts Institute of Technology

Editorial Scope

Topics

Foundations and Trends® in Privacy and Security publishes survey and tutorial articles in the following topics:

- Access control
- Accountability
- Anonymity
- Application security
- Artificial intelligence methods in security and privacy
- Authentication
- Big data analytics and privacy
- Cloud security
- Cyber-physical systems security and privacy
- Distributed systems security and privacy
- Embedded systems security and privacy
- Forensics
- Hardware security
- Human factors in security and privacy
- Information flow
- Intrusion detection
- Malware
- Metrics
- Mobile security and privacy
- Language-based security and privacy
- Network security
- Privacy-preserving systems
- Protocol security
- Security and privacy policies
- Security architectures
- System security
- Web security and privacy

Information for Librarians

Foundations and Trends® in Privacy and Security, 2022, Volume 4, 4 issues. ISSN paper version 2474-1558. ISSN online version 2474-1566. Also available as a combined paper and online subscription.

Contents

1	Introduction	2
1.1	Autonomy	3
1.2	Assurance	8
1.3	Monograph Outline	10
2	Overview of Assured Autonomy	12
2.1	Safety of Autonomous Systems	12
2.2	Security of Autonomous Systems	17
2.3	Reliability of Autonomous Systems	23
3	Governance, Trust, Ethics and Privacy	29
3.1	Governance of Autonomous Systems	29
3.2	Trust in Autonomous Systems	36
3.3	Ethics and Autonomous Systems	38
3.4	Privacy and Autonomy	39
4	Assuring Correct Operation	42
4.1	Formal Verification	43
4.2	Model Checking	54
4.3	Testing Autonomy	61
4.4	Monitoring Autonomous Systems	66

5	Certifying Autonomous Systems	70
5.1	Example Certification of Aircraft	71
5.2	Proposals for Certification of Autonomous Systems	73
5.3	Challenges in Certification of Autonomous Systems	82
6	Research Challenges and Conclusions	86
6.1	Governance, Trust, and Ethics	86
6.2	Reliability and Safety	88
6.3	Security and Privacy	90
6.4	Verification, Model Checking, Testing and Monitoring	92
6.5	Certification	97
6.6	Conclusion	99
	References	103

Assured Autonomy Survey

Christopher Rouff and Lanier Watkins

*Johns Hopkins University Applied Physics Laboratory, USA;
christopher.rouff@jhuapl.edu, lanier.watkins@jhuapl.edu*

ABSTRACT

Autonomous robots and other systems are no longer just subjects of science fiction, but are becoming common occurrences in our everyday lives. Autonomous vacuum cleaners, lawnmowers, and other household helpers are starting to be common place, with autonomous cars now being tested around the world and autonomous drones starting to be used to deliver packages and groceries. Though they will soon be common occurrences in everyday life, assuring their safety, privacy and security is still a huge challenge. A number of autonomous car accidents have occurred after millions of miles of testing, as well as other injuries from other types of autonomous systems. Assuring the proper behavior and safety of autonomous systems is an important endeavor to reduce risks in using them. This monograph discusses assurance for autonomous systems, the different approaches to assuring autonomy, formal analysis, cybersecurity, certification and research challenges.

Christopher Rouff and Lanier Watkins (2022), “Assured Autonomy Survey”, Foundations and Trends® in Privacy and Security: Vol. 4, No. 1, pp 1–116. DOI: 10.1561/33000000027.

©2022 C. Rouff and L. Watkins

1

Introduction

Autonomous systems will soon be ubiquitous in our society, saving us time, performing tasks we do not want to do, caring for us and keeping us safe, often referred to as dull, dirty and dangerous tasks (Connelly *et al.*, 2006). Autonomous robots in homes and businesses are already cleaning floors, mowing lawns, delivering meals and packages, and the technology is now driving cars and trucks. Assuring autonomous systems performance and safety is still a huge challenge. A number of autonomous car accidents have occurred after millions of miles of testing and injuries are also occurring from other types of autonomous systems (Banks *et al.*, 2018; Favarò *et al.*, 2017). Though some autonomous system accidents are minor, others have resulted in deaths to occupants or users, and there is the potential of other damage and injuries from the increasing number and types of autonomous systems that are being proposed.

With the increase of autonomy being used for a wide range of applications, assuring their behavior, trustworthiness, safety and security is still a huge challenge. Providing proper assurance can help prevent injuries, deaths and financial loss. The following subsections give a brief introduction to assured autonomy, providing definitions and terms that will be used in the remainder of this monograph.

1.1 Autonomy

Autonomous systems, also referred to highly automated systems (Falco *et al.*, 2021), have been defined by a number of authors, including Connelly *et al.* (2006), Huang (2007), Huang *et al.* (2007), and Truszkowski *et al.* (2009). Merriam-Webster dictionary defines autonomy as “the quality or state of being self-governing.”¹ For software systems, this means they are not dependent on an outside entity for control or decision making. Connelly *et al.* (2006) define an autonomous system as “one that makes and executes a decision to achieve a goal without full, direct human control.” Hutchison *et al.* (2018) describe autonomous systems as having the following properties:

Stateful - autonomous systems may need to use a large amount of internal memory to represent the environment in which they are operating, keep track of interactions with people and other entities, making models of the physical world around them, developing plans of actions, and reading and storing sensor data that is constantly being received and that needs to be analyzed. Much of the data autonomous systems receive is interrelated and needs to be retained for differing periods of time for future reference and reasoning purposes.

Temporal - autonomous systems often have time and sequence related requirements. They often execute checklists or algorithms that are sequence oriented or where future actions are dependent on past results. An example is going point to point from an initial starting place to a destination. Decisions on the direction from one point may not be known until the autonomous system arrives at that point. In the future these points may be needed to backtrack or return to its starting point.

Distributed - autonomous systems typically contain multiple subsystems that are all communicating with each other over an internal network. Actuators and sensors often have their own processors

¹See <https://www.merriam-webster.com/dictionary/autonomy> (date accessed: 28 May 2021).

and commands are sent to them from a central controller. Data may be sent to a modeling agent that keeps track of the system's current position, goals, plans and other high-level information. Race conditions, deadlocks and other distributed system errors can occur and need to be either tested for or checked through formal methods or other techniques.

Cyber-Physical - autonomous systems are often cyber-physical systems (CPSs) that are an integration of hardware and software components. Griffior *et al.* (2017), at the National Institute of Standards and Technology (NIST), describe CPSs as “smart systems that include engineered interacting networks of physical and computational components” (see Figure 1.1). In the NIST description, multiple CPSs can make up a system (such as an autonomous system) and multiple systems can make up a system of systems (such as a smart city). This makes autonomous systems different from traditional software systems in that they need to detect and deal with hardware failures and sensors that may provide faulty or no data since a human may not be available to detect and fix these problems.

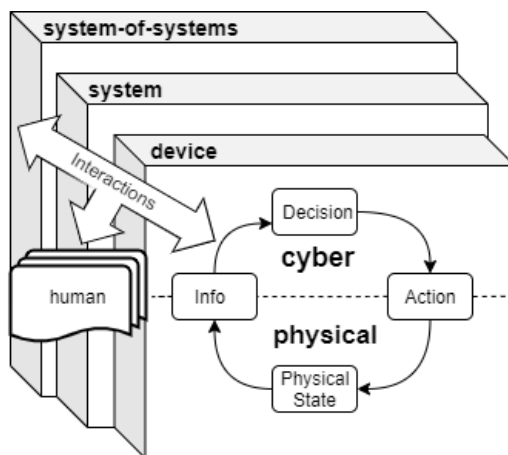


Figure 1.1: NIST conceptual model of a cyber-physical system (based on Griffior *et al.* (2017)).

A question that often comes up when describing autonomous systems is what is the difference between an automated and autonomous system. Both terms refer to processes that may be executed independently from start to finish without any human intervention. Truskowski *et al.* (2009) describe automated processes as replacing routine manual processes with software and/or hardware. Automation follows a step-by-step sequence of steps that may or may not include human participation. The authors describe an autonomous system as having “self-governance” and “self-direction” and can complete a task independently of a human, and have the goal of emulating human processes rather than simply replacing them. Replacing human processes often requires the use of artificial intelligence (AI). Kunze *et al.* (2018) and Nascimento *et al.* (2019) provide some examples of the types of AI used in autonomous systems.

Autonomy may be applied gradually to systems as the technology is developed, making the system more autonomous over time (Truskowski *et al.*, 2005). The system may start out with automation, with increasingly sophisticated or intelligent automated steps added until the system is self-governing and emulating human processes. Sheridan and Verplank (1978) describe ten levels of automation, with the final level being able to operate without human supervision, which could be construed as fully autonomous. The ten levels are (Sheridan and Verplank, 1978; MahmoudZadeh *et al.*, 2019):

1. System is controlled by an operator.
2. System helps operator by determining options.
3. System helps operator by determining options and suggesting one option.
4. System selects an action, which the operator may or may not execute.
5. System selects an action and executes it if approved by the operator.
6. System selects an action, informs the operator in plenty of time for the operator to stop the action.

7. System does the whole job and tells the operator what it did.
8. System does the whole job and only tells the operator if the operator asks.
9. System does the whole job and decides whether to tell the operator what it did.
10. System decides if the job should be done, does the whole job and decides if it should tell the operator.

At level 7, one could argue that the system is autonomous, since it is performing a job and only telling the operator about it after it is performed, which would mean that it is emulating a human process at this level (telling someone after a task is performed is often what humans do). Clough (2002) also defined ten levels of autonomy that ranges from a remotely piloted vehicle to a vehicle with human-like performance.

SAE international has defined six levels of automation that a vehicle may have (SAE, 2021). The SAE driving automation levels are:

- Level 0: No driving automation
- Level 1: Driver assistance
- Level 2: Partial driving automation
- Level 3: Conditional driving automation
- Level 4: High driving automation
- Level 5: Full driving automation

Level 0 is no autonomy and may not even have any automation. Level 1 is when one or more functions are automated. Level 1 does not require sensor information from the environment, and could just be a human activity that is automated. An example would be a simple cruise control that maintains a selected speed, but does not maintain distance from other vehicles or have other safety features. Level 2, partial driving automation, requires sensors to sense the environment, but still

requires driver assistance. Examples would be lane control maintenance or collision avoidance breaking, which would need to sense the lines on the road or obstacles ahead of the vehicle. Level 3 requires a driver, but the driver is not required to monitor the environment, though the driver must be ready to take control of the vehicle at any time with little notice. Level 4, high driving automation, is where a vehicle is capable of performing all driving functions under certain conditions, with the driver having the option to take control of the vehicle. An example might be that the vehicle can operate autonomously only in good weather or in highway environments with good lane markings. Level 5 is defined as full automation, where the vehicle is able to perform all of the driving under all conditions, with the driver having the option to take control of the vehicle when they want. Level 5 could also be construed as emulating human processes, so could be considered an autonomous system.

At the lower levels of SAE driving automation, such as Levels 2 and 3, vehicles may have several independent systems providing automation, such as adaptive cruise and lane keeping technologies. They are usually different systems and can be operated at the same time, or one without the other. For a Level 4 or 5 vehicle, the automation/autonomy must be one integrated system since all of the components must work together (Figure 1.2). The adaptive cruise control may work with the lane keeping function to pass slower vehicles on a highway, along with the other autonomy components to make decisions about speed limits, directions, obstacle avoidance and other functions.

Adjustable autonomy is where the level of autonomy can be adjusted based on the task being performed (MahmoudZadeh *et al.*, 2019; Maheswaran *et al.*, 2003). Mostafa *et al.* (2019) define adjustable autonomy as providing “an autonomous system with variable autonomy in which its operators have the options to work in different autonomy states.” Zieba *et al.* (2010) define adjustable autonomy as “the property of an autonomous system to change its level of autonomy while the system operates. The human operator, another system or the autonomous system itself, can adjust the autonomy level.” Adjusting the level of autonomy in a system can allow a user to take control when the autonomy is no longer necessary, the autonomy is not operating correctly or the user

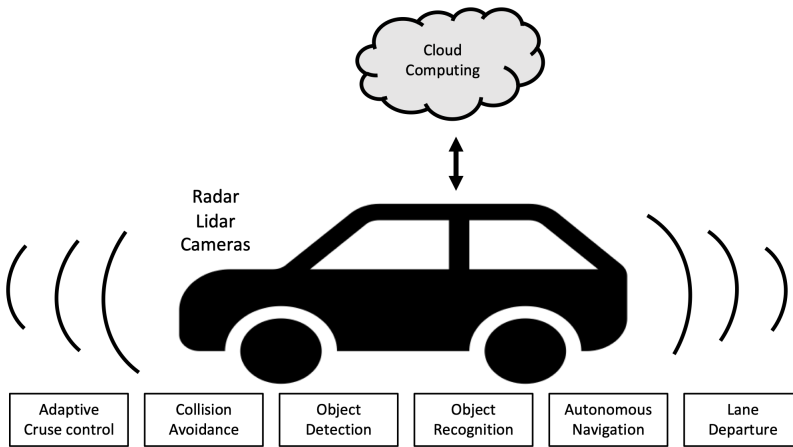


Figure 1.2: Components of an autonomous vehicle.

prefers to have manual control in a given situation. For an autonomous automobile, this could be when the road conditions or the weather makes it difficult for the autonomy to operate, when a sensor fails, or when the driver would just prefer to drive the car themselves. Other examples may be when moving a robot through a tight area, where some of the autonomy is still necessary for navigation, or when learning or other intelligence is not operating properly and some manual control is necessary.

Though there is not a clear agreement between practitioners and researchers on when a system is autonomous and not autonomous, the ability to achieve goals given to it by a human with little or no input is important. For this monograph, the authors will use the definition.

1.2 Assurance

The ability of an autonomous system to operate independently of a human adds a large amount of complexity to the system. This added complexity greatly increases the chances of errors in the system, which adds risks since there may not be a human in the loop that could stop it from causing harm. There need to be assurances that autonomous sys-

tems will operate as intended, and appropriately, even in unanticipated or emergency situations.

Topcu *et al.* (2020) describe assurance as an interdisciplinary research area. What it means for a system to have assurance can differ depending on the community involved. For software engineers, assurance relates to the correct operation of the software that implements the system (Abrams and Zelkowitz, 1995; Saidi *et al.*, 2020; Smith *et al.*, 2020; Wing, 1990). For software engineers, software assurance can refer to formal verification, testing and other approaches used to ensure the system implementation matches the system requirements.

In cybersecurity, assurance can reflect the confidentiality, integrity and availability of a system, referred to as the CIA triad (Gamundani and Nekare, 2018; Samonas and Coss, 2014). Cherdantseva and Hilton (2013) define information assurance (IA) as providing protection by:

reducing risks associated with information and information systems by means of a comprehensive and systematic management of security countermeasures, which is driven by risk analysis and cost-effectiveness.

Cooper *et al.* (2010) define IA as a:

set of technical and managerial controls designed to ensure the confidentiality, possession of control, integrity, authenticity, availability, and utility of information and information systems. IA includes measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and nonrepudiation.

Similarly, NIST defines IA as (Barker, 2003):

Measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. These measures include providing for restoration of information systems by incorporating protection, detection, and reaction capabilities.

To provide IA for autonomous systems there needs to be a way to detect when they are under cyber attack, protecting themselves from attack, protecting the information they contain from being stolen, and ensuring that those that try to access the system and its information are authorized to do so.

Mueller (2019) describes three features that can serve as a foundation of assured autonomy: accuracy, reduction in bias and the ability to reverse engineer the decision-making processes. Accuracy refers to how the autonomy algorithm “senses and perceives the environment in a manner relatable to humans.” This means that the perception of the autonomous system would be similar to how humans perceive the world so that humans can then better relate to how the autonomous system is acting, which reduces the perceived complexity of the system. The reduction in bias refers to the algorithmic and training of the autonomous system, whether the autonomous system is directed toward a result different than what it was originally intended. Algorithmic bias is usually the result of an improper specification of a function, coding errors and other bugs, which causes a degradation in performance of an autonomous system. Training bias is when data that is used by an AI algorithm that is controlling an autonomous system does not represent the environment in which the autonomous system is deployed. This can also be caused by malicious actors feeding the wrong or false data into an AI system, either during training or operations. This means that the autonomous system is taught the wrong information and will not act as intended. The ability to reverse engineer the decision-making processes allows a human to understand why an AI decision was made. When an AI system can explain a decision, it is referred to as Explainable AI (Phillips *et al.*, 2021). AI decision systems often have opaque algorithms, but it is important for humans to understand why a decision was made so that it can be corrected if needed, or just for an operator to understand why an intelligent system operated in a particular manner.

1.3 Monograph Outline

The remainder of this monograph expands on the above descriptions of assured autonomy. Section 2 provides an overview of assured auton-

omy and different aspects of system and software assurances. Section 3 discusses governance, trust, ethics and privacy of autonomous systems. This includes ways the government can be involved in assuring autonomous systems, ways of increasing the trust people have in them, how ethical behavior can be instilled in them, and maintaining the privacy of people who are using or coming into contact with them. Section 4 discusses assuring the correct operation of autonomous systems. This can be done through techniques, such as formal verification, testing and monitoring. Section 5 describes certification of current systems and proposals for certifying autonomous systems. It provides an example of the certification of aircraft software and multiple proposals for how autonomous systems could be certified. Section 6, the conclusion, discusses areas of research in assuring autonomous systems, and some concluding remarks.

2

Overview of Assured Autonomy

Topcu *et al.* (2020) describe assured autonomy as “understanding and mitigating risks of operating autonomous systems in our society.” These risks include the safety, security and reliability of an autonomous system (SAE, 2016; Topcu *et al.*, 2020; Zhao *et al.*, 2019). The risks are not only to the autonomous system, but to the people and the property near them during their operation. The following discusses these risks and how assured autonomy can address them.

2.1 Safety of Autonomous Systems

Safety of autonomous systems, like other systems, is important because they can cause injury or death to passengers or humans that are nearby, as well as cause damage to property and financial loss. Safety of systems have similar definitions across domains which can be leveraged for autonomous system. The Society for Automotive Engineers in their Cybersecurity Guidebook for Cyber-physical Vehicle Systems (SAE, 2016; Schmittner *et al.*, 2016), defines system safety as:

the state of a system that does not cause harm to life, property, financial or its environment ... [a] safety-critical

system is a system that may cause harm to life, property, financial or the environment if the system does not behave as intended or desired.

Schmittner *et al.* (2016) also state that:

system safety considers potential hazards to identify safety mechanisms that can be integrated into the design to address the causes of the potential hazards.

Lastly, Falco *et al.* (2021) define safety as being:

in the context of physical, emotional and financial safety as described in Maslow's Hierarch of Needs.¹

Because of the mobility and physical nature of many autonomous system, physical safety is a top priority for autonomous systems (Falco, 2020). This is especially true for environments where there may be a large number of people in the vicinity of these systems, such as smart cities (Figure 2.1). In the future, the number of autonomous systems operating in a smart city could number in the thousands, from personal and commercial vehicles, to drones, and home and industrial robots. The safe operation of these systems will be a primary concern for both users and those in close proximity to these systems.

Because autonomous systems are still in their early stage and safety strategies are still being developed, there are no safety standards available for developers to follow. One approach suggested is to follow the safety standards of each individual component, and if the components of each individual subsystem is safe, an argument is that then the system that consists of the safe components would also be safe. This may seem intuitive, but system components need to be analyzed for safety as an integrated system, not just individually, since the individual components may act differently when integrated together than when acting individually. Leveson (2020) states that:

One cannot show that each component is safe in isolation and then use that analysis to conclude the system as a whole will be safe.

¹See Maslow (1943)

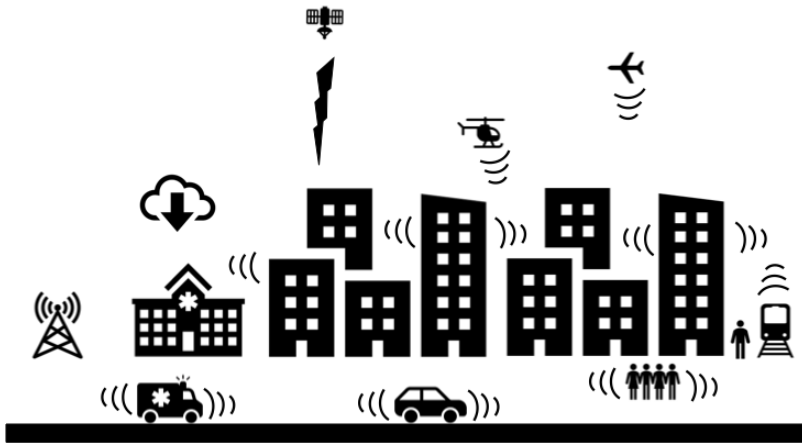


Figure 2.1: Smart city

Leveson describes how systems cannot be modeled or analyzed for safety in isolation from each other, that safety issues can arise due to interactions across un-failed components. Because of this, systems and their components must be tested while interacting with other components or systems in which they will be operating.

Leveson (2020) also states that “virtually all accidents involving software stem from unsafe requirements” and “the most effective approach to dealing with safety or computer-controlled systems is to focus on creating the safety-related requirements.” Based on Leveson’s experience, one of the most effective ways of assuring a system, autonomous or not, is to make sure that safety is part of the requirements and that analysis of components and systems are not be done in isolation, but with other components and systems in which they will be interacting.

Falco (2020) has found similar results and states that autonomous systems are better assessed at the system or network level with all of the autonomous systems that will be operating together tested as part of an ecosystem. Falco has proposed the Assured Autonomous Cyber-Physical Ecosystem (AACE) network architecture that will establish a baseline of safety across all autonomous systems on an AACE network. He proposes that rather than having all of the autonomous systems be designed for safety individually, that an entire ecosystem be made

safe by embedding safety parameters in the network in which they communicate. An example is a smart city that may have thousands of autonomous systems all operating at the same time and interacting either directly, or indirectly due to their proximity to each other. To facilitate this, testing them on an active network is important. Examples of the active network functions would include ensuring that the network is separate from the Internet, demand management, proximal insight, reliable multicast, conflict detection and controlled failure.

Koopman and Wagner (2017) describe how “safe” for an autonomous vehicle should be at least as safe as a non-autonomous vehicle. This would include obeying traffic laws, handling non-routine road hazards, fail-over mission planning, validating learning strategies, resilience in gaps in early requirements, and having a safety certification strategy. The authors describe that developing a safe autonomous vehicle will not be accomplished using a single method, but will be through solving a set of problems in a coordinated, cross-domain manner.

Safety for systems with self-adaptive behaviors will also need to be validated. Autonomous behaviors may need to use techniques that are used in other safety critical systems, such as using a “safing system” (Koopman and Wagner, 2017; Koopman *et al.*, 2019). Safing systems take control when a critical component fails, the system starts operating outside its approved envelope of behaviors, or system degradation is detected. A safing subsystem could also be used for autonomous systems. In the case of an autonomous drone, the safing subsystem could land the drone in an open area, and provide the operator with error messages or results of self-diagnostics, with applicable actions to take. Such a subsystem could relax safety requirements if it could be shown that it can take over at the appropriate times and safely stop the system.

Arguing a system is safe is another method that has been used in some domains when there is no standardization or certification process. Koopman *et al.* (2019) state that there are many ways to argue a system is safe, including:

- Following a prescribed set of engineering practices,
- Brute force testing, and

- Claiming that something is safe because it has been proven in use.

Due to the newness of autonomous systems, a set of prescribed engineering practices have not emerged for autonomous systems. Engineering practices are developed over the course of many years in developing a type of system, so they may come after there has been a number of years of experience in developing autonomous systems. As mentioned earlier, due to the large state space of autonomous systems, completely testing an autonomous (and many other) systems would be impossible. The last way, claiming something is safe because it has been proven in use, is the approach that many autonomous automobile manufactures are using by driving their vehicles collectively millions of miles (McCarthy, 2021).

Koopman *et al.* (2019) describe a method for arguing that a system is safe through safety arguments. A safety argument provides evidence that a system is acceptably safe to be used in a given context (Anaheed *et al.*, 2013). A safety argument is made up of safety cases, which is defined as (UK Ministry of Defence, 2017):

a structured argument, supported by a body of evidence that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given operating environment.

Since autonomous vehicles are a composition of technologies, most of which are mature, with some new autonomy technology added, a heterogeneous safety argument could be made by combining safety cases for both the mature and new technologies. For example, for an autonomous automobile, a safety case could be made that the underlying automobile has been safely produced for years and adheres to regular automobile safety standards. A safety argument for the new autonomy technology could then be developed based on testing or other means, and the two combined into a composite safety argument. The safety argument for the autonomous vehicle would then be an amalgamation of the different safety arguments, techniques and assurance approaches into a heterogeneous safety argument. As noted above, a safety analysis

needs to consider the integrated system, not just the safety of the individual components, so a system level safety argument would also need to be made.

Separate safety cases could be used for new autonomous technology, with different safety cases aggregated and evaluated together, representing an integrated system (Koopman *et al.*, 2019). Different safety cases could also use different strategies based on the maturity and type of technology. A catalog of safety cases could then be developed of various autonomy technologies. Safety arguments for new technology could also use existing safety standards. If the technology can conform to an existing standard, then an argument could be made that it is safe. For example, autonomy software for automobiles may conform to existing automotive software safety standards. Safety cases for new sensors could use existing automotive sensor standards to show that they adhere to those standards. “Proven in use” safety arguments could be used based on field experience of a component or engineering process to argue field failure rates are low enough to assure the component. A component could also be deployed with a backup for redundancy until long term failure rates are known.

Similar to proven in use, field testing is based on using a fleet of systems that are tested in the real world, and then after a large number of hours in use, they can be declared safe in those contexts that were tested, and used in production. Simulations are sometimes used in safety arguments where they are used as a substitute for field testing, but given the large state space of most autonomous systems, only a relatively small number of possible scenarios can be simulated. Formal proofs of correctness can also be used that prove system safety properties, which could include model checking, control system analysis, use of validated synthesis tools and kinematic analysis, which is discussed further in Section 4.1.

2.2 Security of Autonomous Systems

Though autonomous systems can operate without the assistance of a human operator, they can still have security vulnerabilities that need to be addressed. These may be from wireless communications that can be

intercepted and manipulated, manipulation of sensory data to confuse the system or make it do something unintended, or take advantage of a security hole in an axillary component or system. The following discusses some of the security concerns and possible vulnerabilities of autonomous systems.

Cyberattacks on autonomous systems have already occurred (Vinnem and Utne, 2018). In one instance a hacker remotely disabled over one hundred cars (Poulsen, 2010). Vinnem and Utne (2018) describe a discovery that one million cars could be hacked simultaneously and how cars can become weapons of mass destruction on the road. Kim *et al.* (2021) describe several possible attack vectors for standard and autonomous vehicles. Since autonomous vehicles are being built on current vehicle technology, many of those attack vectors will still be present in autonomous versions. One of the primary attack vectors are vehicles' electronic control units (ECU). ECUs control such things as the doors, seats, anti-lock brakes, engine control unit, and other devices. A medium sized car can have 50 or more ECUs. The ECUs communicate over vehicle networks, such as the controller area network (CAN), local interconnect networks, and other networks. If attackers can gain access to the CAN, they may be able to exploit the ECUs on the network and take control of the vehicle.

Autonomous vehicles also have sensors that are not on standard vehicles. These sensors provide data for the autonomy software, which can also have vulnerabilities (Yan *et al.*, 2016). These can include light detection and ranging (LIDAR) sensors, video cameras, radar and ultrasonic sensors for object detection, as well as computers to process sensor data and execute the autonomy software. Communications technologies are being proposed and developed to connect cars together to warn them of accidents, road conditions and other hazards. Communications include vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-roadside (V2R), vehicle-to-pedestrian (V2P), vehicle-to-device (V2D) and other technologies (Skrickij *et al.*, 2020). Together they are referred to as V2X (Figure 2.2). Using these technologies vehicles are able to form ad hoc networks for V2V communications, communications with road-side units (RSU), and for use of cellular communications. Similar technologies are being developed for drones and other autonomous systems in smart cities.

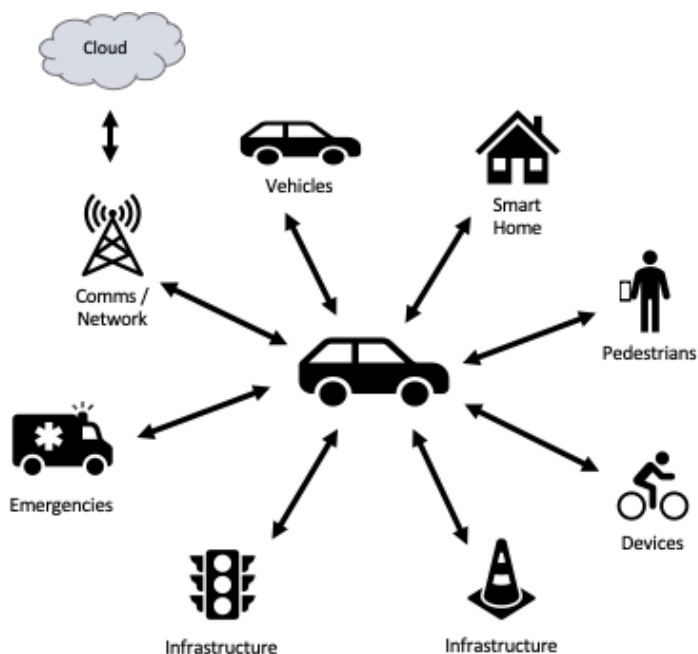


Figure 2.2: Vehicle to vehicle and infrastructure communications (V2X).

Malicious actors could use the above technologies as attack vectors if they are not sufficiently secure. Examples of previous attacks on autonomous cars include attacks on control systems, autonomy driving components and V2X communications (Kim *et al.*, 2021). Attacks on vehicle control systems target the ECU, in-vehicle networks and the vehicle's key. In-vehicle networks tend to have weak (or no) authentication and access control, and are subject to denial-of-service (DoS) attacks. Firmware has been able to be loaded onto ECUs without authorization, allowing attackers to perform malicious activities, such as manipulating the speed of the car and changing the driver's display. The tire pressure monitoring system (TPMS) was an early wireless vehicle network that could also be compromised and used to sniff personal information.

There is usually no encryption on in-vehicle networks, so a malicious actor can observe data and messages on the network, and create messages to be inserted into a CAN to send unauthorized commands to ECUs.

Attacks on the autonomous driving system components can include attacks on sensors and attacks carried out by compromised smartphones that may be in a vehicle. Attacks on the V2X communications technology include attacks via multimedia centers, cellular networks and through the ad hoc networks formed with other vehicles, which could have vulnerabilities. All of these attack vectors would be applicable to smart cities as well.

Koopman and Wagner (2017) describe how security measures will need to encompass system-level attacks from the autonomous vehicle infrastructure as well as attacks on specific vehicles. Communicating vehicles may not want to blindly trust other vehicles or roadside infrastructure in which they are communicating, all of which could themselves be compromised. Encrypting communications and requiring certificates would help by authenticating connections, but if the vehicles or roadside infrastructure that are performing the authentication have been compromised, then malicious actors may still be able to perform their attacks. Vehicle and roadside infrastructure will need to be prudent with their communications and authentication as well as verify information they receive to be correct.

For security, vehicles will need to be able to operate independently of V2X communications in case those communications go down, are subject to a DoS attack, or the information received is suspect or not verifiable. Vinnem and Utne (2018) describe how interconnection of autonomous vehicles may expose safety-critical systems of autonomous vehicles to security threats. Schmittner *et al.* (2016) describe applying the SAE Cybersecurity guidance for developing a secure ECU. They define a lifecycle process to incorporate cybersecurity, provide some common tools and methods, guiding principles, and further standards development in this area.

The SAE Cybersecurity Guidebook for Cyber-physical Vehicle Systems defines system cybersecurity as (SAE, 2016; Schmittner *et al.*, 2016):

the state of a system that does not allow exploitation of vulnerabilities to lead to losses, such as financial, operational, privacy or safety losses.

Schmittner *et al.* (2016) also state that:

system cybersecurity considers potential threats posed by a malicious attacker whose goal is to cause harm, wreak havoc, gain financial benefits, or simply to gain notoriety.

Not all security-critical systems are safety-critical. For example, an entertainment system may be security critical because it could contain personal information, but it is not safety critical. Cybersecurity is concerned with security vulnerabilities, not random failures. Security of sub-systems and access to safety-critical components through non-safety-critical components is also of importance, such as accessing ECUs from the entertainment system if it is on the same network as the ECU. Autonomous, and all computing systems, should be designed with cybersecurity in mind from the beginning.

Analysis of cybersecurity and system safety have the same goals, but the analysis may be different. System safety use techniques such as fault tree analysis (FTA), to identify potential hazardous events, while system cybersecurity may use analysis techniques, such as attack tree analysis (ATA) (Schmittner *et al.*, 2016). Some cybersecurity and safety processes are similar, but cybersecurity has some processes that are unique to it. For cybersecurity, potential threats to the system need to be analyzed to determine potential attack surfaces. Personally identifiable information (PII) and sensitive data should also be identified and protected, while minimizing PII data collection to limit it being compromised. Owners of the PII should have the ability to remove it when ownership of the system changes or the end of life of the system occurs. The practices of least privileges should be used and unauthorized changes to the system prevented.

Cybersecurity of a system also needs to be managed (Schmittner *et al.*, 2016). This includes managing the cybersecurity activities during the development lifecycle, as well as the overall management of cybersecurity, such as fostering a cybersecurity culture in the organization, training, monitoring of potential attacks against the domain, and ensuring compliance to engineering processes. A cybersecurity lifecycle plan can be developed that describes the activities to be carried out,

as well as a threat analysis and risk assessment (TARA) performed to assess potential threats. Any cybersecurity issues that are related to operations can be documented in the vehicle owner's manual, as well as any service that would effect cybersecurity documented in the maintenance manual.

Autonomous ships are being developed for both commercial and military applications (Andrews, 2018; Wright, 2020). As with automobiles, ships can also be vulnerable to cyberattacks. Vinnem and Utne (2018) describe potential effects of malicious actors taking over autonomous ships, which have a number of differences from automobiles. While ships go significantly slower than autonomous cars, and operate in open waters with few other ships nearby, compromised autonomous ships could be used as battering rams to cause major structural damage to ports, offshore drilling rigs and other infrastructure. Ships can have very high kinetic energy when traveling at even moderate speeds and could cause significant damage to structures or other ships.

Ships also travel long distances through international waters, so they could be compromised at an intermediate stop, an international location or even in route in open seas. Because of this, all autonomous ships will need to be treated as potential threats. If an operator is onboard, then the operator could manually override a ship that an intruder has gained control over. In the worst case scenario, an onboard operator could shut off power and let the ship drift until control is regained. But if there are no operators onboard, then the ship could be a potential threat until it is safely docked or someone can board it to take manual control. Other manned ships may be able to avoid a compromised autonomous ship, if they know it has been compromised, but infrastructure that is not moveable would be vulnerable. Compromised autonomous ships, if large enough, could be used to block narrow passages, such as the blockage that occurred in the Suez Canal by a manned ship (Lawrence, 2021; Ramos *et al.*, 2021), which in the past has been warned could happen (AlFadhli *et al.*, 2021; Herbert-Burns, 2009).

Keeping a small crew onboard is listed by Vinnem and Utne (2018) as the most effective risk mitigation, with another potential risk reduction being to limit the fuel an autonomous ship has so that its range is limited and could not go far off course to perform an attack. However,

a compromised autonomous ship could still attack infrastructure that was within its range. Another potential risk avoidance action cited by Vinnem and Utne (2018) would be to limit autonomous ships to stopping offshore and not autonomously docking, though this might prevent a compromised ship from attacking, it could also provide the people onshore time to engage defense mechanisms so that the autonomous ship could only cause limited damage, such as by automatically blocking the entrance to a dock. Though the above is specific to autonomous ships, it could be applied to other types of autonomous systems as well.

2.3 Reliability of Autonomous Systems

System reliability is defined by Klaassen and Van Peppen (2006) as:

the probability that this system uninterruptedly performs certain (accurately) specified functions during a stated interval of a life variable, on the condition that the system is used within a certain specified environment.

A definition of software reliability by Meshka (2020) is the probability of failure-free software operation for a specified period of time in a specified environment. Though reliability and safety seem close in meaning, Maurya and Kumar (2020) state that safety is more concerned with the threat to the environment or human life, whereas reliability focuses on the failure-free execution of a system for a specific period of time in a specific operating environment.

Reliability and safety of systems go together. An unreliable system can malfunction, which can then lead to safety issues (Falco, 2020). Falco (2020) describes the need for reliable delivery of data to all subsystems of an autonomous system as crucial for safety so that the subsystems can jointly orient themselves to an environment. An example is a drone that is trying to avoid a collision needs to provide its coordinates to other systems (autonomous or non-autonomous) in the area so they know where it is. If the coordinates only go out to a limited number of other systems, then there is an increased probability that a collision will occur since some systems may not know the drone's position.

An unreliable autonomous system could result in injury, financial loss or death. For example, an autonomous automobile could turn into a crosswalk and strike a pedestrian if its sensors are not reliably detecting obstacles in front of the vehicle. If an autonomous system has some reliability problems, and the system has the self-awareness (Truszkowski *et al.*, 2009) to know it has reliability problems, it needs to announce those problems to other systems it is interacting and in which it may come in contact (physically and logically). This will allow other systems to decide how they are to proceed given the known problems with a nearby system. If an autonomous system does not know that it is unreliable, other systems in contact with or observing it could vote that it is unreliable and modify their behavior based on that information. Being voted unreliable could also be communicated to the effected autonomous system, and if it can receive a message about its unreliable behavior, it could also modify its behavior, such as maintaining more distance between itself and other systems, not using a bad sensor, stopping itself or taking itself offline until it can be fixed (such a voting system could also be an attack vector, which should be considered in its design). If the unreliable autonomous system can not be contacted, then other systems operating around it, or dependent on it, may replan and adjust their actions with the knowledge that the unreliable system may act in an unpredictable manner or not be able to accomplish its mission.

Tanzi *et al.* (2014) state that when a reliability problem occurs it should trigger an emergency event, such as for UAVs, they could perform an immediate landing. The authors also suggest that autonomous systems continuously evaluate a ‘point of no return’ that is dependent on the type of mission. For UAVs, this would be when they no longer have enough fuel/energy to return to base, or for an autonomous automobile when it can no longer reach a gas or charging station. At this point, the plan would change so that it remains in range of its home base or a refueling station. Other examples of points of no return may occur when it is supposed to deliver something for a timed event and it will not be able to get there in time (such as due to bad traffic or weather). The authors also discuss that mitigating or preventing risks involves inserting safety and security mechanisms into components

of autonomous systems, as well as the overall architectural design, to detect these risks. The authors suggest inserting checks for securing communications (such as using authentication), performing plausibility checks on sensor and other data to make sure the data values are within a range (given the circumstances and environment), a security module for cryptographic and other security operations, and having real-time constraints, such as deadlines for UAVs so that they will be able to return safely.

Falco (2020) also discusses how safety controls need to be installed on the lowest level of sensors and autonomous controls that can detect when that subsystem is not reliable. This allows detection of errors as soon as possible. This would require that sensors or subsystems have self-tests and have some self-awareness as to whether they are operating correctly or not. If they are not operating correctly, then they need to warn the governing part of the autonomous system that the subsystem is not reliable and why. The autonomous system can then determine whether it should continue or not, and whether the issue rises to the level that other systems should be warned of its potential unreliable behavior.

Siil *et al.* (2020) discuss mission assurance and safety for autonomous undersea vehicles. The authors propose a Mission Assurance for Autonomous Undersea Vehicles (MA-AUV) resilience layer (RL) that replicates essential mission functionality with less-complicated safety systems (Figure 2.3). They propose placing the RL between potential faulty or compromised autonomy functions and the sensors and actuators that the autonomy uses. The RL limits the effects of any potentially dangerous actions coming from the autonomy engine (AE), with the goal of maximizing protection of the mission and the system, occupants and bystanders. The RL receives the same sensor data that is sent to the AE. In normal conditions the RL lets commands generated by the AE flow to the motors and actuators of the autonomous system. If the RL needs to take over, it can send commands to the motors and actuators directly rather than the AE, thus giving the RL control instead of the AE.

The RL is a safety layer that bases its decisions primarily on maintaining physical separation between the autonomous system and other

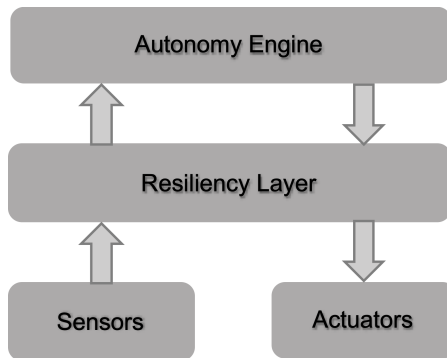


Figure 2.3: Resiliency layer for autonomous systems.

objects, fixed or moving. The RL is based on the five NIST Cybersecurity Framework functions of Identify, Protect, Detect, Respond and Recover (Barrett, 2018). The Protect functions grant permission to activate a tour plan. Without a plan, the AE is not allowed access to the motors and actuators so it can not proceed until a plan is available. The Detect function monitors the system's position and distance from obstacles or whether it is overdue at its destination. The Respond function is executed when the system is in danger and alerts the AE that it must take action. The Recover function is executed when the system has lost separation from obstacles or when a new plan is required and the AE has not executed one, in these cases the RL takes control of the system and finishes the mission or travels to a safe area.

Hutchison *et al.* (2018) state that autonomous systems are almost always safety critical systems. The physical components of an autonomous system, whether it is a vehicle or a robot arm, usually has safety issues associated with them since the physical component can unexpectedly collide with objects (such as autonomous drones with other drones or people) and cause harm. The authors propose a robustness testing framework that can automatically discover safety problems in autonomous systems. Such a system would have a robustness testing framework with the following key features:

Use of safety invariants - expresses safety properties that are more complex than traditional oracles.

Injection of messages - manipulates values in messages sent over internal networks, often spanning multiple messages, which leverages the distributed nature of autonomous systems.

Interception testing - mutates live message values, which reduces reliance on user-provided sequences, to test stateful and temporal properties.

Based on their experience of testing autonomous systems, the authors provide the following recommendations for building robust autonomous systems:

- Protect your autonomous system from data assumptions - often assumptions about data being received are made by developers, which can be wrong, such as the ordering of messages that will be received.
- Plan for the system to fail - since all programs have bugs, there should be a means to report errors to the rest of the system, and the system should compensate for the errors and not fail because of them.

Meshka (2020) describes an architecture that incorporates system reliability with runtime verification and validation (V&V). The architecture, shown in Figure 2.4, provides a high-level architecture for autonomous systems. Past and current system states are provided as input to a planner. The planner also takes a path sequence that the autonomous system is currently on and sensor information from the system's environmental inputs in which it is operating. The planner then produces one or more plans for achieving the goal state given the system state and the current location and sensor readings. The Planner then produces multiple plans and sends them to a V&V engine to verify them for completeness and optimality. A complete plan is then checked for reliability by the System Level Reliability module. This module uses a reliability model, shown as a Bayesian belief network (BBN), to assess the risks involved in the plan. Based on the assessment, the plan may or may not satisfy the risk constraints for this system. If the risk is outside the tolerance, then another plan is evaluated. If none of the

plans verified by the V&V engine are within tolerance, the V&V engine requests a new set of plans from the planner. Once the System Level Reliability module finds a plan within the acceptable risk tolerance, it sends it to the system as a final plan to be executed. This allows for a changing risk model as well as a changing risk tolerance that can be adjusted based on the user (e.g., passenger of a vehicle) or situation.

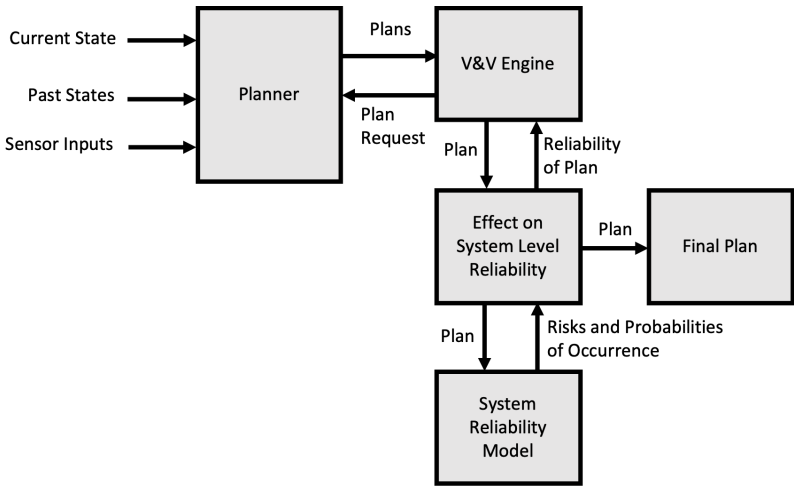


Figure 2.4: Architecture for incorporating system reliability into V&V considerations. Based on Figure 4 in Meshka (2020).

3

Governance, Trust, Ethics and Privacy

Governance, trust, ethics and privacy for autonomous systems are all interrelated. Proper governance of autonomous systems needs to be based on ethical actions which can lead to more trust and privacy. If autonomous systems take ethics into account when performing actions, this can help increase peoples' trust that the technology will perform with their and societies' best interests, which can help to increase adoption by the public. Ensuring privacy of users' and bystanders' information will also help people trust the systems and increase their adoption. This section describes each of these areas as well as how they interrelate to help people adopt autonomous systems technology into their lives.

3.1 Governance of Autonomous Systems

The U.S. Department of Transportation (USDOT), National Highway Traffic Safety Administration (NHTSA), developed the Federal Automated Vehicles Policy document to govern autonomous vehicles (NHTSA, 2016). Given the quickly changing technology used to implement autonomous vehicles, NHTSA is making updates on a regular basis with guiding principles for automated vehicles (NSTC and US-

DOT, 2020) as well as a comprehensive plan that describes how the USDOT is supporting the safe integration of automated vehicles into the transportation system (U.S. DOT, 2021). Though these policies and plans are specific to automated vehicles, many of these same policies and plans could be used for autonomous systems in other applications as well.

The NHTSA Automated Vehicles (AV) policy document (NHTSA, 2016) is the agency's guidance for the autonomous vehicle industry, as opposed to rule-making for the industry. The policy provides a regulatory framework to help with the safe introduction of highly automated vehicles (HAVs) into the transportation system. The framework is divided into four sections:

- Vehicle performance guidance for automated vehicles
- Model U.S. State Policies
- NHTSA's current regulatory tools
- New regulatory tools and authorities

The vehicle performance guidance is not mandatory or meant to be codified by states into legal requirements, but provides best practices for safe pre-deployment design, development and testing of HAVs prior to operation on public roads. HAVs would still need to meet all of the Federal Motor Vehicle Safety Standards (FMVSS) like standard vehicles do. Figure 3.1 shows the USDOT framework for the Vehicle Performance Guidance (NHTSA, 2016). The framework applies to areas that include all automation functions on a vehicle (cross-cutting), such as data recording and sharing, privacy, etc. It also applies to specific automation functions on the vehicle, such as the Operational Design Domain (ODD) Object, Event Detection and Response (OEDR), and fall back minimal risk conditions. To use the guidance, manufacturers first receive certification for all applicable FMVSS standards, and for HAV systems. The manufacturer would then address the cross-cutting items as vehicles are designed and developed, and define the ODD and corresponding SAE level to which the vehicles map.