TuSimple's Driver-Out Pilot Safety Framework

tu simple

Table of Contents

A. Introduction

Purpose

The purpose of this safety framework document (Safety Framework) is to communicate our overarching approach to safety for our upcoming Driver-Out Pilot, and to document the structured safety framework that we have developed that is designed to ensure risks will be appropriately mitigated for driver-out operation within the selected Operational Design Domain (ODD).

Unlike the traditional automotive industry, which has long-established standards to prove safety and roadworthiness of human-driven vehicles, the autonomous vehicle (AV) industry is at its beginning stages of development. This gives us the opportunity to be an integral and active part of creating wholly-sufficient standards. We believe it is our responsibility to develop and employ standards-based methodologies to guide AV-specific standardization and to communicate our overall safety solution.

Given the unprecedented complexity of autonomous driving technology, we choose to build on the current automotive industry regulations around safety by incorporating multiple substantive safety methodologies into our safety framework. That is, our safety framework goes beyond basic compliance with standards and instead seeks to quantitatively establish that we have produced the safest autonomous trucks on the road. Since our autonomous freight lanes will always operate on public roads in the presence of other road users and essential infrastructure, our safety framework must inherently provide the means to evaluate our true performance in that environment. We believe our safety framework achieves that objective.

At TuSimple, our approach to safety is holistic, spanning our technology, processes, organization and operations. As such, this document covers aspects of both traditional safety case framework structure as well as augmentative elements relevant to their successful implementation. Naturally, we begin with explicit safety principles. We then derive our overall safety framework by taking full advantage of available best practices and internal frameworks to ensure rigor in our safety perspective. However, ultimately, we see safety as a continuous journey rather than a

single destination; safety is never truly complete. Thus, while our Safety Framework fully describes the safety approach we are taking with our Driver-Out Pilot, it represents just one significant step toward our development of substantively safe autonomous freight operations at the global level.

Scope

This document solely and specifically presents our safety framework for our Driver-Out Pilot and supersedes all prior safety framework documentation relating to that application.

The scope of this document includes all product development and operational activities that directly contribute to the safety of a pilot of this technical complexity, including:

- The technical definition, design, implementation, verification and validation of the trucks that will operate in a driver-out configuration during the pilot.
- The activity design, planning, training and testing implemented to ensure all hazards not explicitly handled by the autonomous trucks are appropriately mitigated by way of operational controls.
- The organizational processes and safety governance mechanisms used to ensure quality and sufficiency of the technical and operational activities above.

While the document may contain elements that also apply to the safety of our future SAE Level 4 (L4) autonomous trucks or our future full scale autonomous freight lanes, those applications are explicitly outside the scope of this document.

B. Organizational Structure & Safety Governance

It is impossible to design a device as complex and safety-critical as an autonomous freight lane without first building a culture of safety, accountability and compliance designed to ensure discipline in analysis and design, predictability in execution, and conformance in quality.

Corporate governance serves as the foundation underpinning successful safety culture by:

1. Assigning unambiguous leadership accountability for safe outcomes in development and operation of the company's products and services.
2. Establishing safety policies that clearly define the safety-centric commitments that the company makes to its employees, investors, regulators and the public.
3. Defining and clearly communicating the safety standards, processes, procedures, goals and milestones of the organization.
4. Implementing the organizational structure and processes that enable efficient, cross-functional execution of all necessary safety assessments, analyses and reviews.
5. Monitoring, measuring and controlling the organization's overall safety status, and implementing reliable quantitative mechanisms to ensure continuous improvement of its safety performance.

Given the impact of governance on our safety outcomes, we have actively prioritized optimization of our organizational structure, policies and product development processes in an effort to ensure success of our safety mission. Specifically, at the corporate level, we have formed a Safety Policy Steering Committee (Safety Committee), formalized our company safety culture and implemented alternate safety reporting feedback channels like our safety hotline. These three items are elaborated in the remainder of this section.

Our Safety Committee holds primary responsibility for establishing, validating and confirming the Driver-Out safety framework, managing necessary adjustments and assuring adherence to the Safety Framework before launching the Driver-Out mission on public roads. A significant amount of work has gone into structuring the committee and defining its objectives and expectations. The Safety Committee is composed of a cross-functional group of our senior-most technology, legal and regulatory executives, including our President & Chief Executive Officer, Chief Technology Officer, Chief Product Officer, Chief Administrative & Legal Officer and Vice President of Systems Engineering & Testing.

Since our inception, we have prioritized a culture of safety and have instituted a structured methodology behind our safety practices and procedures. The goal of this cornerstone of our culture is to enable the entirety of our team at TuSimple to always act with safety as a central guiding principle. As a company, we work to address a significant unmet need for improving overall driver safety on our public roads by establishing a safer driving system with TuSimple's virtual driver. Internally, our prioritization of safety is evident throughout our hiring, training and day-to-day employee experience. Along with building the right teams, we equip and train our employees on industry-leading safety methodologies, practices and procedures. Furthermore, we empower employees to raise concerns and validate through leadership action that such concerns are heard, taken seriously and acted upon.

Though our Safety Committee holds ultimate responsibility, we have worked to ensure that all employees feel empowered to participate in an open dialogue around our safety practices. Along with promoting a culture that challenges the status quo and prioritizes safety, we have instituted a safety hotline where employees can report behaviors or decisions that they believe to be unsafe. This provides all employees across our organization the ability to confidentially and anonymously report any existing issues regarding safety matters and contribute to company-wide adherence to our safety policies and procedures. The safety hotline serves as an essential component to TuSimple's safety initiatives, ultimately allowing us to take action and ensure the highest levels of safety for our team and the motoring public.

C. Product Development Process

At TuSimple, we understand the need to balance the creation of innovative, industry-leading technologies with the need for safety of our products, employees and the broader community of road users. Vehicular autonomy entails (1) creating novel robotics technologies, (2) maturing those technologies to a readiness level suitable for safety-critical operations, and (3) integrating those technologies together to produce L4 autonomous trucks. A critical capability of our product development process, therefore, must be robust requirements discovery at speed. Consequently, at TuSimple we implement a hybrid of the V Model traditionally used in safety-critical industries and the Agile design maturation model more common in technology companies. Together these provide a multifaceted view of the safety needs of the product along with the accelerated mechanism needed to implement and validate them.

V Model at TuSimple

The V Model is a top-down methodology for designing complex systems. It specifically addresses the need to make appropriate decisions about system requirements, architecture and design in a stage-wise manner, to allow gradual refinement of the design as decisions become more concrete. By taking a stage-wise approach, conflicts between desired system features are made visible early in the process, preventing implementation of hardware or code that will ultimately fail to satisfy the overall requirement set. In addition, through use of analysis and testing tools geared to the level of design abstraction available at each stage, it is possible to verify and validate performance of the design at each level, thereby increasing confidence in the robustness of the design as the process progresses. These characteristics of the V Model yield both cost and schedule savings in the design process, as compared to a purely iterative, test-driven approach to achieving the same level of product performance and quality. Furthermore, these benefits grow proportionally with the complexity and/or safety-criticality of the application because the costs of rework escalate rapidly in those applications as the implementation progresses.

A L4 fully autonomous truck is significantly more complex than a traditional automobile. In addition to the complexity of the base vehicle platform, which is essentially a traditional automobile, there are the sensing, computation and high-speed communications necessary to artificially replicate human-like driving abilities. The complexity of the artificial intelligence alone, which is used to perform the perception, prediction and planning aspects of autonomous driving, easily exceeds that of the rest of the system. The V Model was designed to manage exactly this type of complexity and to ensure that design trade-offs and safety are considered at each layer

of a system definition, including its subsystems, their modules and any other complex functions present.

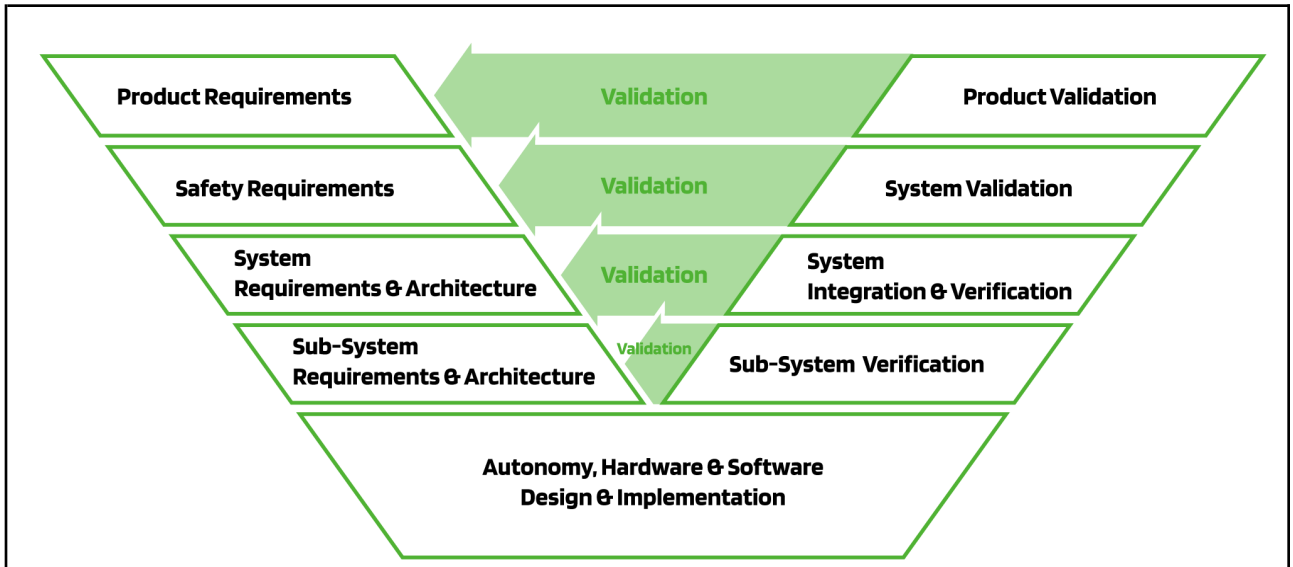

Figure 1: TuSimple V Model

A high-level view of TuSimple's V Model is shown in Figure 1 above. Critically, the model enforces systematic end-to-end development of our autonomous platform by ensuring that:

1. Product and safety needs are factored into the design from its inception.
2. System-level requirements, architecture and safety analyses thoroughly cover the application scope and provide technical coordination of subsystem implementation.
3. Autonomy, hardware and software subsystems are implemented in compliance with product and system-level needs.
4. Verification and validation are performed incrementally as system integration progresses.
5. Every autonomous truck design passes rigorous product validation testing before being released for broader autonomy applications on public roads.

This V Model provides us with a mutually exclusive and collectively exhaustive design approach, one that enables deep understanding of system risk and safety performance at every step along the process. The trade-off, however, is slower *initial* progress than a purely Agile model.

Agile at TuSimple

In contrast to the V Model, Agile methodologies were specifically designed for rapid requirements exploration and to quickly evolve a full system design starting from a more limited initial scope. With Agile, the intent is to create an initial useful output, or minimum viable product, and then iterate to expand and mature it into a more robust process, system or technology. At the end of each iteration, new requirements are identified from the reviewed results of prior deployments, which themselves are used to design the next implementation and so forth, as depicted in Figure 2 below. Given that vehicular autonomy is still an unsolved challenge, such iteration through real world application is unavoidable if we are to eventually understand the full scope of capabilities required. Agile fills this need effectively, using rapid prototyping to develop and incrementally validate new capabilities as they are found necessary. At TuSimple, prioritization of which capabilities are to be prototyped in this way is done via the V Model, such that both methods are hybridized to deliver rapid progress in areas with the most significant need for safety or performance improvement.

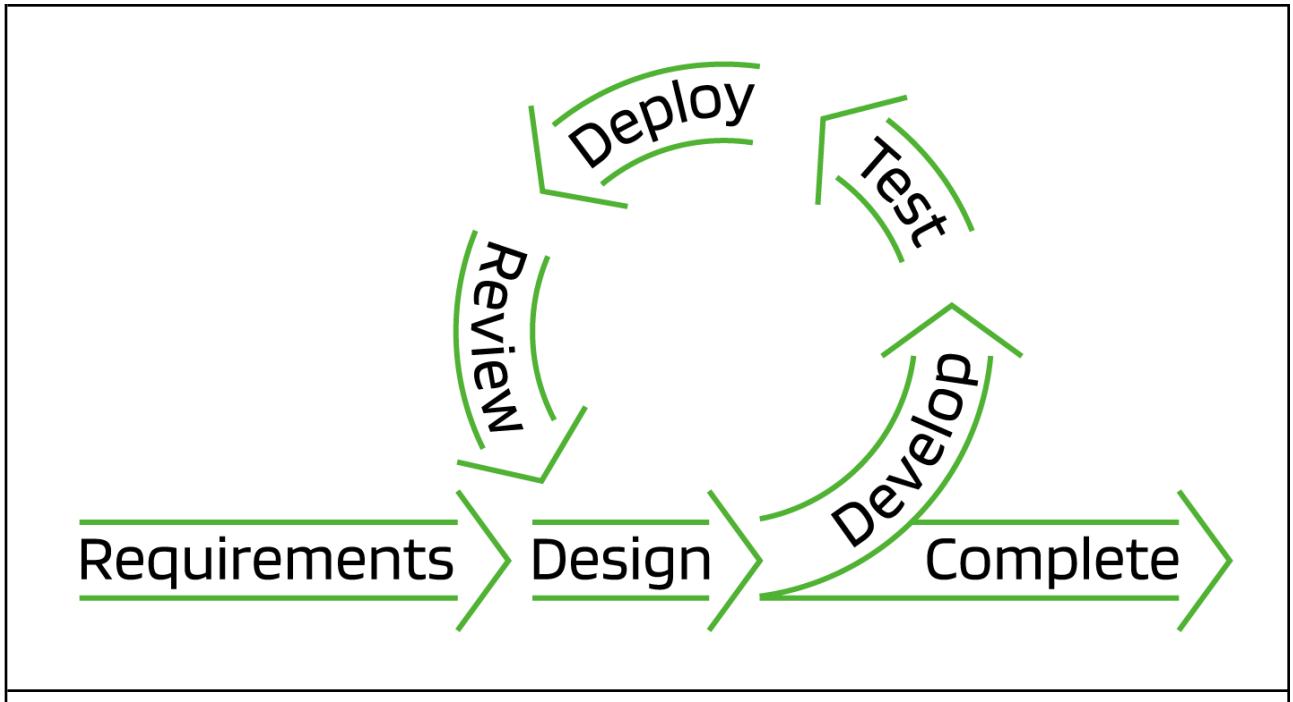

Figure 2: The Agile Model

Ultimately, each new capability goes through a comprehensive set of testing that includes virtual and regenerative simulation, hardware-in-the-loop, track testing, then road testing with a certified professional safety driver and test engineer. A summary view of this test flow is depicted in Figure 3 below. As shown in Figure 3, each of these testing modalities independently generates feedback to the agile team and to higher level requirements owners where necessary. This thereby helps to detect and correct deficiencies as early as possible in the development process to prevent safety-critical issues from progressing to public roads.

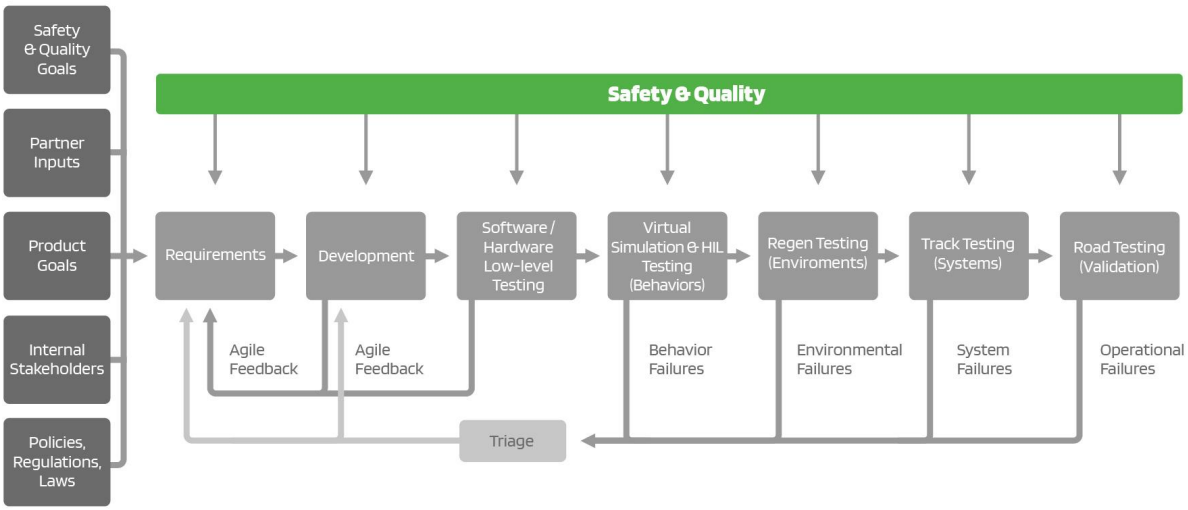

Figure 3: TuSimple Verification & Validation Flow

Safety of Autonomous Trucks

When we think and talk about safety, we start by defining what matters to us the most. Our top priority is to avoid injury or fatality caused to any road users. At the same time, reducing the potential for fatality or any level of injury due to non-liable accidents is equally important.

In order to achieve what matters to us the most, we have defined a behavioral policy that our autonomous trucks need to abide by:

Behavioral Policy:

1. We must never operate in an uncontrollable manner

Our top priority, to avoid injury or fatality, relies on our ability to keep our autonomous truck under control at all times and under all circumstances. Keeping the truck under control means that our goal is to prevent any situation in which our autonomous truck:

   a. disengages from autonomous mode before reaching the end of a mission.
   b. behaves incorrectly on its intended route during its mission.
   c. proceeds on its mission without the availability of our accompanying operations team (which includes our survey and chase vehicles).

2. We must not proceed to our destination after loss of functional redundancy

We have built our trucks based on the principle of functional redundancy. Functional redundancy allows for the architectural accommodation to provide fail-operational capability which is required to allow an autonomous truck to run on public roads without any fallback to a human driver. If functional redundancy is lost, the autonomous truck would still have the capability to functionally operate with its full operational capability; however, the ability to fall back to components that provide that functional redundancy will have already been used up. At that point, additional potential losses in components may lead to total loss of functionality beyond simply loss of functional redundancy. For that reason, and depending on the functional redundancy that may be lost, we have developed multiple strategies that define how the autonomous truck will operate to achieve a minimal risk condition (MRC) instead of proceeding along the mission route.

3. We must avoid staying in close proximity to non-compliant drivers that may cause accidents

Our road test miles have shown frequent occurrences of drivers who struggle to stay within their lane boundaries. These drivers present an additional safety concern for our autonomous operations. For that reason, we must avoid staying in close proximity to these non-compliant vehicles to reduce the risk of being in a situation where a collision with these vehicles becomes imminent.

4. We must do everything we can to reduce the impact speed of imminent accidents

Although our goal is to avoid all potential causes of accidents within the ODD, we know from historical traffic data that there may be non-trivial cases where other road users might cause an accident involving our truck. Our commitment to safety, even in such potentially unavoidable cases, is to leverage every available option to minimize the probability of fatality or injury to human drivers, their passengers or other vulnerable road users. We will attempt to achieve this by always applying the brakes, removing as much kinetic energy from the truck as possible to reduce the impact speed, thereby reducing the risk of severe injury or fatality.

Risk Reduction vs Standard Certification

Our safety focus for a driver-out demonstration is the identification of unreasonable risks and implementation of measures to reduce those risks. We realize that current industry standards are designed to address safety-related risks of production-intended electric and/or electronic devices in the presence of a human driver behind the wheel and, hence, do not fully apply to the safety development of L4 autonomous driving technologies. For that reason, we have invested our efforts in leveraging what applies to our technology from those standards as engineering best practices, but kept our focus on risk identification, reduction and control.

Inductive and Deductive Dual Approach

Industry safety standards focus on a requirements-based approach. However, theoretical aspects of development cannot represent the entire safety solution. To be specific, safety goals derived from worst case scenarios based on product requirements can provide a theoretical picture of what could go wrong at the vehicle level. However, hazards identified from actual disengagements on the road reveal actual safety risks that the development team needs to address immediately, enabling the team to build a more comprehensive list of safety goals retrospectively.

At TuSimple, we adopt both approaches since they are complementary, providing a cross-verification framework to work with while, we believe, also ensuring a more comprehensive view of safety hazards across the board. Another advantage of this approach is that it serves as a validation of our product requirements. Similar to the V Model, where safety goals are generated from product requirements, safety goals generated in a data-driven approach must also be traced to product requirements. Specifically, autonomy disengagements reveal insufficiencies in product requirements surrounding the situation leading to the disengagement. To detect such insufficiencies and accelerate product validation, we have a trained team of triage experts that

builds a hazard analysis and risk assessment item list from actual disengagements on a weekly basis. In addition, the same team works on establishing traceability to product requirements and helps to identify missing product requirements accordingly.

Safety Execution That Extends Beyond State-of-the-Art Industry Standards

As a leading autonomous trucking technology company, we believe that a holistic safety framework should not fall short of leveraging the autonomous driving industry's use of automotive and technology standards, tools, methods and principles. At the same time, we are aware of the inadequacy of some of the standards and guidelines to fully cover the autonomous trucking application. For that reason, we have structured our safety solutions around the use of both nominal safety standards and substantive safety principles, yielding a comprehensive safety solution that we believe addresses the most-relevant safety aspects of autonomous driving within our targeted ODD. We have considered relevant safety standards, methods, tools and principles currently in use in the autonomous driving industry that fit the context of our own technology and ODD, and have integrated them into a holistic approach that also includes our own proprietary methods.

Figure 4 shows a top level view of what we believe fits the context of our application. In the sections that follow, we elaborate on each component of Figure 4 and how we have applied it at TuSimple.

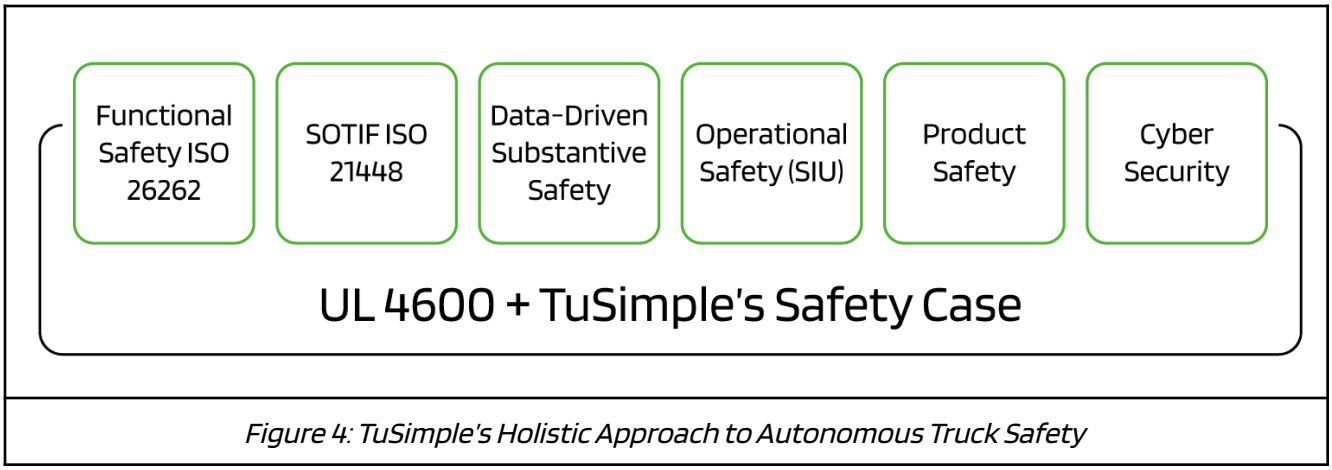

Figure 4: TuSimple's Holistic Approach to Autonomous Truck Safety

1. Functional Safety ISO 26262

The ISO 26262 Functional Safety standard is about reducing the risks of simple and complex systems, such that they function safely in the event there is an electrical or electronic malfunction. Being a requirements-driven approach, it is coupled with product requirements and systems-engineering work products (requirements and architectures) and focused on adding safety elements and processes to vehicle electronic systems.

We have applied functional safety analysis on our hardware (HW) and software (SW) system elements, starting with identifying safety goals at the vehicle level and then decomposing them into lower-level safety requirements using fault tree analysis. We have endeavored to cover every aspect of our architecture including sensing, vehicle control interface, vehicle actuators and body controls, external communication interfaces, power management and data communication. We have supplemented our deductive fault tree analysis with design failure mode and effects analysis (DFMEA) on the system architecture, and our internally-developed software and hardware designs of our compute and sensing units. The overall result is a safety concept that identifies potential single points of failure, latent failures, common cause failures, safety-related unavailability and loss of functional redundancy.

In addition, we have identified all diagnostic information reported by each off-the-shelf sensor, compute and actuator component that interfaces with our autonomous driving system and developed a safety monitor that monitors for faults and initiates the appropriate maneuver to achieve MRC. Further, we have installed measures in place to provide storage protection while transferring map data from and to the on-board storage drive to avoid accidental loss or corruption of map data.

Empirically, we have injected faults into our system to ensure the proper responsiveness of the system is initiated to transition to the redundant components, we have tested our system to ensure that our truck's stability is retained when our system transitions to redundant controls and actuators, and that the switching to the redundant components is fast enough and smooth without any operational interruption. Also, we continuously test to ensure that our redundant and fault-domain separated battery packs can provide sufficient power to the rest of the system upon loss of power generation source. And finally, we make use of on-board legacy collision avoidance advanced driver-assistance systems (ADAS) in the very unlikely event that our own autonomous driving system falls short of avoiding any collision on the road.

2. SOTIF ISO 21448

The ISO 21448 Safety Of The Intended Function (SOTIF) standard addresses the sufficiency of the intended function in the absence of faults or failures. It is highly relevant for automated systems in removing unreasonable risk of functional insufficiencies or by foreseeable human misuse. It is also a requirements-driven approach that is coupled with product requirements, systems-engineering work products (requirements and architectures) and focused on making vehicle electronic systems safe by design.

Our application of SOTIF is split into two levels; the vehicle level and the realization level. At the vehicle level, we have been identifying cases where product requirements may lack completeness, contain conflicts, or miss certain requirements. Our application of SOTIF at the vehicle level comes from two areas; analytical and empirical. Analytically, we have applied a SOTIF-based Hazard Analysis and Risk Assessment (HARA) method as well as System Theoretic Process Analysis (STPA) to identify potential conflicts with planner decision intentions and deniers. This is an ongoing process that we have been applying iteratively. On the flip side, our triage team that performs road test data-based HARA also helps reveal missing product requirements where autonomy behaviors may have been absent or incorrect in the absence of any electric or electronic faults.

At the realization level, we focus on identifying technology limitations within our sensors, compute platforms or actuators that cannot perform to satisfy our product requirements. The use of STPA at that level helps reveal design aspects that may provide insufficient capabilities. For example, certain sensors may not provide the required range of detection under specific environmental constraints causing inability to detect objects. Another example would be an actuator that is unable to provide the required level of actuation under specific driving scenarios to avoid destabilization or collision with other road users. We rely on our ODD database that helps us identify objects within our ODD, in combination with our regular disengagement reports that help us understand the challenging scenarios where technical limitations may have been discovered. This information is fed into the STPA process to help document the empirical data and identify design measures to circumvent those limitations.

In addition, we follow a systematic process to identify out-of-ODD objects (objects that are possible to be encountered on public roads but which our self-driving truck is not yet designed to handle) within our operational zone. However, we realize there may be additional objects that our truck may not have been exposed to yet while testing on public roads. These objects are classified as unknown objects. Utilizing our long range light detection and ranging sensors (LiDARs), our proprietary algorithms provide a virtual shield that helps bring our truck to a safe stop and avoids collision with unknown objects in its driving path.

Lastly, the SOTIF specification suggests defining a target validation criteria to empirically prove the functional sufficiency of our system. Unfortunately, it does not provide sufficient guidance on how to determine such a validation target, especially for driver-out autonomous driving systems. For that reason, we have developed our own target miles-per-event validation methodology based on statistical extrapolation of our road metrics over mileage. Crucially, we have determined the number of validation miles we need to run without any unplanned disengagements before we can perform the driver-out demonstration.

Empirically, our simulations, track tests and road tests provide us with the confidence that our trucks are robust to scenario variations, tolerant to noise and resistant to latency and jitter with upstream sensors. In addition, we constantly monitor the behavioral intentions our planner provides in each scenario we are exposed to in real world testing to ensure we are compliant with traffic rules and regulations. Furthermore, we aim to ensure that non-compliant road users who invade our driving lane are reacted to swiftly and correctly to avoid potential accidents.

3. Substantive Safety - Advanced version of ISO 21448

Although there is not a standard available that fully addresses this area of safety, we believe that our integrated tools and safety checks within our algorithms are excerpts from safety industry standards carefully curated to our specific application. This area addresses the safety of non-deterministic behaviors of autonomy functions in the absence of fault(s) or failure(s). It is a data-driven approach focused on designing a safety envelope for algorithmic modules developed using machine learning (ML) tools and methods. Although the current SOTIF ISO 21448 is specifically developed for L2 autonomy or lower and with the human driver as a fallback measure, we believe that our solutions form the basis of what could be the advanced version of ISO 21448 (L3-L5 autonomy). We apply non-classical and progressive safety analysis methodologies that focus on causality analysis with respect to identifying triggering events and developing design measures to eliminate or reduce risk.

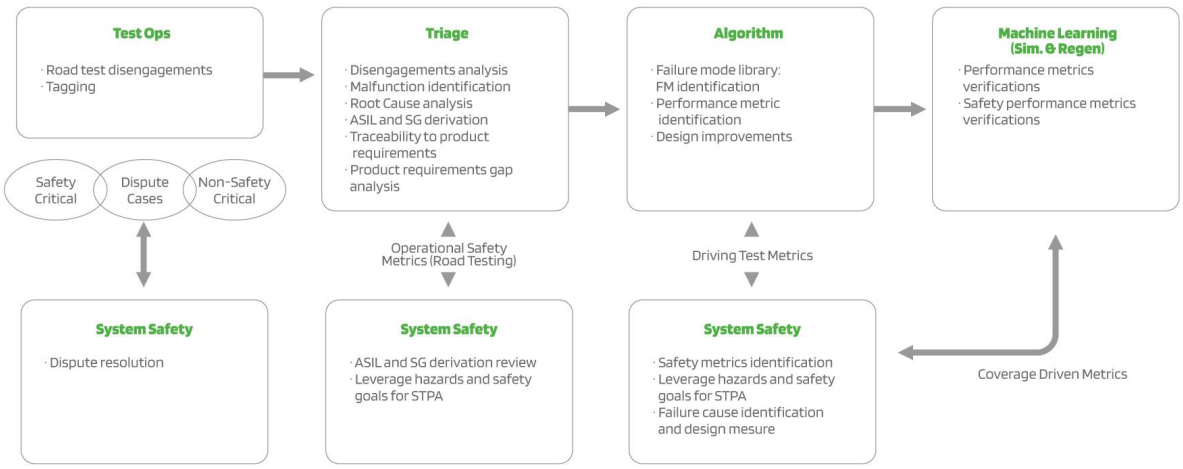

Figure 5: Substantive Safety Analysis Flow

Figure 5 shows the high-level structure of our substantive safety flow. We begin with our test operations team reviewing and assigning tags to every single disengagement occuring during public road testing. Each disengagement is attributed with an abbreviated severity risk index. In some cases, our system safety team helps resolve disputed cases where test engineers may not easily be able to determine if the case would have been safety critical or non-safety critical.

All cases are handed over to our triage team that helps identify the root cause behind the behavioral malfunctions leading to the disengagements. As already mentioned, the same team helps construct hazard analysis cases, derive safety goals and product requirements traceability or gap analyses with support from our system safety team. The system safety team ensures these safety goals are compiled along with the safety goals derived from product requirements flowing top-down throughout the safety analysis process.

When a root cause is identified within an algorithm module in the software stack, our algorithm development team constructs a library of failure modes to compile the list of all observed failure modes responsible for the disengagements. Each failure mode is then assigned a performance metric that serves as guidance on how to improve the performance of the algorithm by suggesting design improvements. The system safety team provides safety metrics associated with those performance metrics in an effort to ensure the absence of all failure modes within the

critical areas where these failure modes have appeared. Also, leveraging STPA, our system safety team guides our algorithm team in the identification of design measures. In addition, the system safety team closes the loop on the process by linking all the data-driven hazards to the data-driven system safety constraints (safety goals) identified by the triage team and to the design measures identified by our algorithm team.

Finally, our machine learning team uses state-of-the-art simulation and regeneration methods to verify that the identified safety performance metrics are valid. In addition, our benchmarking team applies conditional benchmarking methods and tools to help identify all the scenarios that meet the safety performance metrics that the system safety team defines and, using conditional mining, helps find the most interesting combinatorial scenarios in an effort to ensure we have complete coverage over our ODD.

Empirically, we have been stress-testing our system by performing worst case testing on test tracks to probe for SOTIF-related functional insufficiencies, and adversarially testing the system to find ways we can break the system to reveal weaknesses not identified analytically or proven statistically on public roads or in simulation. For example, we thoroughly test the effectiveness of our unknown object detection along with other challenging scenarios wherein the planner may be forced to trade between different combinations of safety-critical responses.

4. Operational Safety (Safety of Use) - Advanced version of ISO 21448 Annex E

ISO 21448 contains guidance on how to address human driver misuses of ADAS technologies. While our technology is targeting driver-out applications, there are still a considerable number of operators that interact directly or indirectly with our autonomous trucks. Although these actors are not behind the wheel, the role they play can factor into operational safety hazards. Safety hazards due to human operational errors and misuses related to on-board operators and connectivity services (oversight) are given special attention at TuSimple, including defining those hazards using the HARA method that leverages our product requirements, as well as our application of a Safety of Use analysis using STPA.

We have applied these methods and derived operational safety goals and operational safety requirements so our oversight operational safety team members can avoid safety-critical operator errors. Specifically, we have defined detailed instructions on the timeliness and order of communication between our survey vehicle, chase vehicle and law enforcement vehicles, as well as the dedicated roles and seating positions for the operations team in these vehicles to ensure

the teams have the proper awareness levels of all risks assessed. We have also defined requirements on when to remotely request specific types of safe stop maneuvers depending on the triggering condition. The requirements form the basis of the operational support team training to be performed prior to the final validation testing, in preparation for the Driver-Out Pilot. In addition, our operations team is continuously improving our system by continuing to monitor for disengagements and map updates, and taking corrective and preventive actions as necessary.

Our operational safety work does not stop with the operations team. Our engineering support teams are also carefully trained to ensure that the correct SW configuration and calibration are used, the correct start-up sequence is initiated, and the correct takeover is performed at the end of the pilot. We have designed the system to require no SW build change while transitioning from the completion of the final validation to the initiation of the Driver-Out Pilot. Instead, we employ an interlocked SW configuration coupled with a multi-step authentication process. This is designed to ensure a tightly controlled operation that is free from accidental or unauthorized access to the on-board system, and prevents the SW from operating in the wrong mode. This is a crucial step in our transition to the Driver-Out mode that also inhibits accidental disengagement of the system while it is operating without a safety driver in autonomous mode.

Finally, as we reinforce our system with all possible technical safeguards, we endeavor to ensure that in the very unlikely event when our autonomous truck experiences an unforeseen event, our oversight operations team is trained and has practiced to execute the appropriate safe stop maneuver as well as the emergency response teams are prepared and drill tested.

5. Product Safety - Extension of IATF 16949

Quality Management Systems are the industry standard in automotive engineering. In fact, safety standards such as ISO 26262 are built in compliance with IATF 16949. This standard makes reference to product safety without diving deep into the details. At TuSimple, we reference this standard but expand its scope to cover safety hazards due to non-detectable mechanical issues, serviceability, and maintainability (not addressed by ISO 26262 and ISO 21448). It includes pre-trip inspections, service hazards, and maintenance hazards.

Prior to each test trip, our safety drivers perform the required Federal Motor Carrier Safety Administration (FMCSA) inspections on each truck. We have expanded the inspection to include autonomous operations inspections related to the overall system health status, HW health status

and SW health status. The status of each SW node is monitored over a Human Machine Interface (HMI) in real time and at all times. In addition, we have developed an application that provides a report from the vehicle controls to ensure all startup safety checks are clear. An operator ensures all manual and Autonomous Driving System (ADS) safety checks have been successfully completed and passed, before providing the final signal to the autonomous truck to engage in autonomy and begin its mission.

Our HW and truck build staff monitor all HW faults and failures experienced by our trucks, compiling and tracking a list of all observed issues and working to find a resolution for each issue. After root causes are identified, some of these issues may be service or maintenance related and are addressed accordingly. Specific instructions are provided to ensure specific maintenance actions are followed in a specific manner to avoid safety concerns. Examples include how to handle manual truck washing, including the type of cleaning solution to use to avoid residue on camera lenses that may leave the sensor blinded.

6. Safety Performance Indicators

We believe that it is not enough to simply achieve a required level of safety; we must also maintain that level of safety and measure its improvement over time. Every safety framework is implicitly built on a number of underlying assumptions, thus it is critical that the validity of those assumptions be continually monitored and verified throughout use of that safety framework. For example, it is clear that only tracking the number of disengagements in a given number of miles is not sufficient to reflect the true distribution of potential risks in an ODD. If traffic patterns change, so too will the observed disengagement frequency. For this reason, we have developed a list of safety performance indicators to track organizationally and across several verticals to gauge the continued applicability of our underlying assumptions and monitor overall safety improvement.

6.1 Operational Safety Metrics (Road Testing)

These metrics help us track the on-going effectiveness of our operational controls, including maintenance sufficiency, operator training, driver disengagements and safety driver attentiveness during autonomous runs. As an example, they can show the frequency of unpredicted disengagements initiated by our safety drivers, which provides implicit feedback on whether our training and other operational controls appropriately address the root causes of the disengagement.

6.2 Autonomy Driving Test Metrics

These metrics help us track ADS algorithm performance against the original validation criteria from training and test data, with a view to detecting module-related insufficiencies and shifts in autonomy performance relative to their baseline. For example, these metrics can include measuring perception precision and recall relative to their baselines, as well as logging detections of the need to disengage, false positive/negatives and mispredictions, near-hits and near-misses, etc.

6.3 Application Coverage Metrics

These metrics help us track simulation coverage of the ODD, real world test coverage of the ODD, analytical safety hazard identification coverage relative to disengagement-related hazards identified, and arrival rates of unknown objects with respect to ODD violations.

6.4 Software Quality Metrics

These metrics help us track software quality relative to key ISO 26262-6 and Hersteller Initiative Software (HIS) source code quantitative metrics (code quality, code complexity, static analysis defect rates, development process metrics).

6.5 Design Maturity and Progress Metrics

These metrics help us track software reliability and capability growth. For example, these metrics often involve measuring and contrasting the number of bugs identified in each release with those found in prior releases, as well as monitoring safety and performance margin evolution with code releases.

6.6 Conformance Metrics (for SW, much of it is covered in SW quality metrics)

These metrics help us measure our compliance of our ADS to applicable portions of the standards mentioned above, including ISO 26262, ISO 21448, MISRA C, HIS, and ISO 21434.

7. Cyber Security ISO 21434

Cyber security and functional safety are tightly coupled together in the area of safety-critical systems. The standard ISO 21434 covers all stages of a vehicle's lifecycle — from design to decommissioning. It applies to vehicle electronic systems, components, software and external connectivity. It provides a structured process to ensure that cybersecurity considerations are incorporated into automotive products throughout their lifetime and has specific

requirements for software development including analysis to check for inherent weaknesses and the overall consistency, correctness, and completeness with respect to cybersecurity requirements.

It is not our specific intent to obtain industry standard certification for our autonomous trucks that will be used in the Driver-Out Pilot, given that they are not intended for mass production. However, we still emphasize use of applicable methods and best practices – such as those found in ISO 21434 – to identify potential security risks and threats in order to develop effective security measures. For example, we use internal threat modelling, Threat Analysis & Risk Assessment (TARA) and third party security consultation to derive security goals, requirements and risk mitigation strategies. Additionally, we have an internal risk sign off procedure designed to ensure each risk and security goal is reviewed, addressed and approved by our internal stakeholders.

8. UL 4600

We have developed a customized safety framework for our specific Driver-Out Pilot. Despite this fact, we still find value in the relatively new UL 4600 standard as a rough checklist to ensure our safety framework covers the basic principles that matter most in deploying autonomous vehicles safely on public roads. We have reviewed and mapped certain parts of the standard to our specific organizational areas and ensured that our safety framework arguments are broadly aligned with the standard.

Interdependence Between All Safety Guidelines

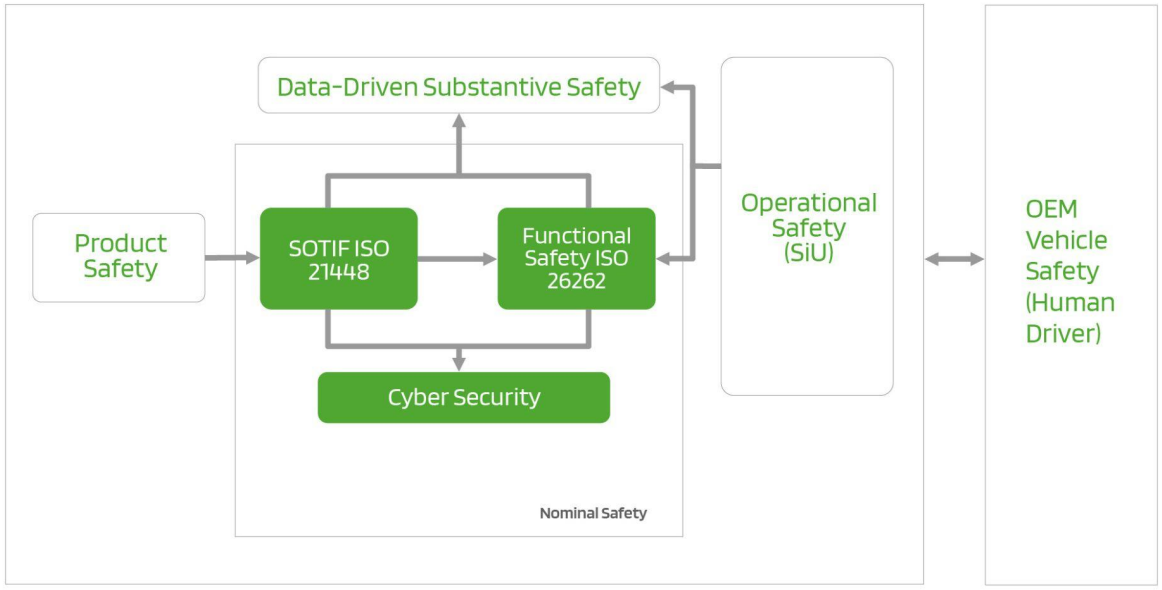

Figure 6: Relationship between Safety Areas

Now that we have covered how we apply the referenced standards and guidelines in the context of our application, a cohesive relationship between all the different areas becomes apparent. Figure 6 shows a depiction of the interdependence between these standards and guidelines, particularly highlighting how functional safety, SOTIF and cyber security combine to yield nominal safety. In comparison with a truck original equipment manufacturer's (OEM) safety case for a vehicle operated by a human driver, our safety case framework covers many aspects that an OEM's does not, particularly in the areas of substantive safety and operational safety. An OEM's safety case is usually centered around claims that the human driver would act as the safety measure for many of the safety hazards identified. In contrast, we are replacing a human driver with a virtual driver within the same operating environment. Thus, we go above and beyond an OEM's safety case in an effort to ensure all corners of development within the autonomous ecosystem are addressed effectively.

While we do not claim that safety can be achieved without any residual risk, we continue to expand on our safety case framework to maximize the diagnostic coverage of our safety concept

and minimize the residual risks to negligible values. Ultimately, our aim is to eventually implement prognostic safety measures that predict system faults or deficiencies before they take effect, so we can preemptively take corrective actions and further reduce residual risks in our autonomous trucks.

D. Safety Case Framework

In an effort to ensure safety for our Driver-Out Pilot, we have developed an extensive safety case framework that begins with two overarching safety principles:

[1] Driver-Out Trucks are Safe to Operate Autonomously on our Driver-Out Route
[2] Driver-Out Technical Operations Assure Operational Safety

These principles are shown as the foundations of our safety case framework in Figure 7, the components of which are expanded on in detail throughout the remainder of this document.

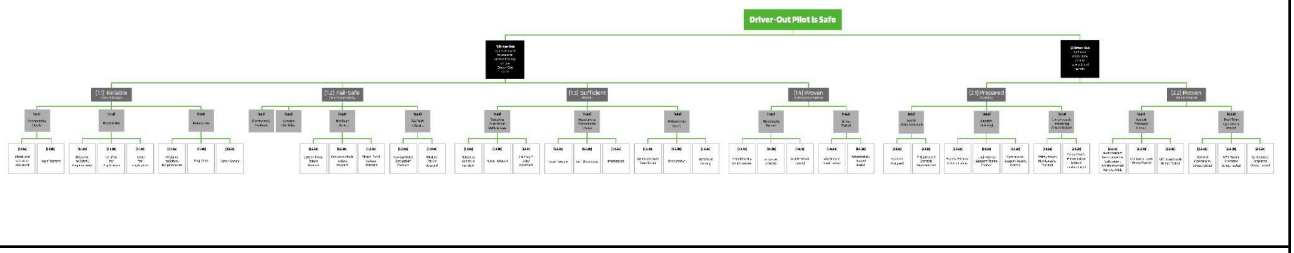

Figure 7: TuSimple Safety Case Framework built to ensure we achieve a safe Driver-Out Pilot
Download full Driver-Out, Pilot Safety Case Framework Flowchart.

[1] Is the Driver-Out Truck Safe to Operate Autonomously on the Designated Route?

Along with following a regimented product development and corresponding testing process as our internal best practices, we have also instituted a safety case framework specifically for assessing and ensuring safety throughout all of our developed systems. Our safety case framework works to ensure success of our first safety principle, that Driver-Out trucks are safe to operate autonomously on our Driver-Out route by endeavoring to guarantee that each aspect of the system is *reliable*, *fail-safe*, *sufficient* and *proven*.

This safety case framework is designed to ensure that we analyze every aspect of our system technology and act as a central function to all engineering disciplines; from software development to hardware development to the integration of all software and hardware components. This safety case framework will help guide our engineering teams to develop engineering design documentation that details out the specifics of our software and hardware design as well as apply state-of-the-art safety analysis methods and tools to identify safety-critical points of failure. Furthermore, with the help of engineering experts, the System Safety function will define, implement and test measures to detect and prevent each safety-critical failure at different levels of our development.

[1.1] Reliable (Robust Design):

Reliability addresses the required baseline level of performance of each system with its hardware and software components over a period of time. All hardware components (including sensors, computing, harnesses and connectors) have been procured or built to relevant specifications and stress tested for the specific Driver-Out ODD application.

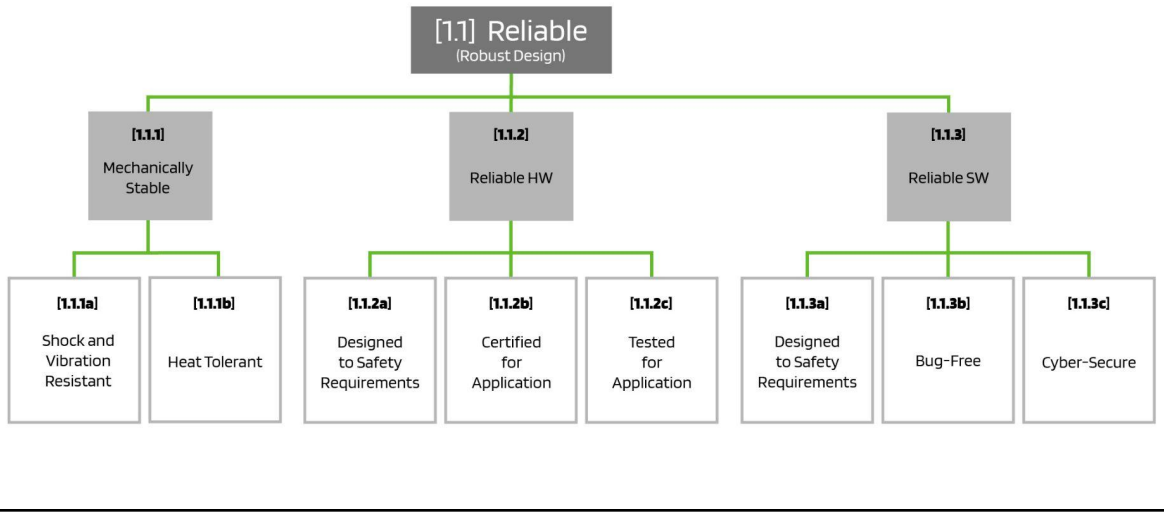

Figure 8: Deep Dive of TuSimple Safety Case Framework section [1.1] Reliable (Robust Design)

[1.1.1] Mechanically Stable

The stability of all internally-mounted and externally-mounted HW components that are on the cab and/or the trailer is paramount to ensure components do not accidentally fall off when exposed to excessive vibration, shock, and variable environmental conditions such as temperature and humidity.

[1.1.1a] Shock and Vibration Resistant

When shock or vibrations due to rough road conditions or cab movement occur, the mounts of HW components on the cab and the trailer can have severe mechanical stress that challenges the strength of each mount. If one or more mounts break, this could lead to having components falling off the cab and/or trailer and potentially causing safety hazards to the operation of the

autonomous truck or the surrounding road users. For that reason, all internally-mounted and externally-mounted HW components that are on the cab and/or trailer must be resistant to excessive shock and vibration.

[1.1.1b] Heat Tolerant

Excessive heat from the environment as well as heat generated by devices causing increased ambient temperature can impact the operation of some HW components (including sensors) and stress the environmental operational ranges of each HW component. In addition, excessive heat generated by the engine can quickly cause the temperature of any HW component located in the engine bay area to increase dramatically and in a short period of time. For that reason, all sourced HW components must be shielded from excessive heat outside its normal operating temperature range, selected appropriately to operate in the targeted application environments and provided with the proper cooling methods to keep it operating within its range.

[1.1.2] Reliable HW

The HW components we use must be designed to safety requirements, certified for application and tested for application. Compliance with automotive grade and safety standards, developing quality management systems, component availability, conforming to assumptions of use, and field test data are among a few artifacts that we use to support the argument of the use of reliable HW components.

[1.1.2a] Designed to Safety Requirements

Many of the HW components that we use are off-the-shelf components. These components are either designed as a safety-element-out-of-context (SEooC) device with specific assumptions made to interfaces and constraints (also known as assumptions of use), or developed in compliance with specific safety standards. Our safety requirements that are derived from our functional safety process determines whether the assumptions of use can be satisfied by each HW component used. Similarly, the HW components that we have developed are designed according to safety requirements that specify how fault detection and monitoring needs to be carried out. All of the design details and analyses are subject to configuration management, documentation management and change management processes under a quality management system.

[1.1.2b] Certified for Application

Some of the most critical devices that we use need to provide the adequate level of safety-compliance to ensure the proper execution. This means that these devices must provide the appropriate level of failure-in-time (FIT), diagnostic coverage for single point faults and latent faults, as well as being automotive grade. There are a wide variety of standards that components can be certified to or comply with, and many of these standards belong to the same family.

[1.1.2c] Tested for Application

Compliance with safety requirements and certification to safety standards does not suffice without proving the functional ability to carry out the safety functions and possessing the functional capability to prolonged reliability. For that reason, every component we use must pass rigorous testing and field monitoring. This includes all field failure data obtained from our truck build process, truck operations, bench tests and stress tests. Testing also extends to functional testing, fault injection testing and electrical and environmental testing.

[1.1.3] Reliable SW

Much like HW reliability, SW components must also provide functionality that is bug-free, safe and secure. A reliable software stack must be developed according to clear specification that is change controlled, version controlled and has clear requirements governed by a set of quality management systems and processes. In addition, our SW quality metrics demonstrate maturity in achieving a bug-free code.

[1.1.3a] Designed to Safety Requirements

The SW components that we have developed are designed according to safety requirements that specify how fault detection and monitoring need to be carried out. All of the design details and analysis is governed using configuration management, documentation management and change management processes under a quality management system. In addition, system power up, system shut down, and system engagement and disengagement capabilities are ensured to be bypassed while the system is engaged in autonomy to avoid unintended disengagements.

[1.1.3b] Bug-Free

When SW code is written and compiled, there are a large number of coding errors that the compiler can identify. Some of those errors (also known as lint errors) do not cause the software to malfunction, but some SW errors (bugs) can break the function of the software when it is used on the trucks. These software bugs must be debugged to eliminate safety critical malfunctions from the SW. At the same time, the allocation of SW on the target computing HW may cause the processors to run slower or may use more memory than can be provided. Therefore, careful consideration must also be given to processing power and memory usage when the SW is deployed on the HW units. We perform SW code reviews, measure SW quality metrics, perform SW unit tests, review compiler errors and warnings and measure resource usage and performance.

[1.1.3c] Cyber-Secure

Our autonomous truck is designed to be operated by the virtual driver only, with very limited allowable remote intervention by our operations team. Any additional intervention by third parties can be considered as an unauthorized and unauthenticated intrusion that can interfere with the truck's ability to be solely controlled by the virtual driver. For that reason, we aim to ensure that all entry points to the vehicle's communication network that provides access to the powertrain and the actuators are secured from unauthenticated access. In addition, access protection to the truck's HMI system is secured to prevent unauthorized access to the SW.

[1.2] Fail-Safe (Functional Safety)

Our system must continue to operate safely in the presence of the full array of hazardous malfunctions that can occur within it, including failures in software and hardware components, system interfaces, power networks, communication network and electrical/electronic subsystems. We apply functional safety methods and principles to identify system functionalities whose availability is safety critical, identify hazards at the vehicle level, define safety goals and break it down to lower-level safety requirements that are allocated to system HW components and SW components for each safety-critical single point failure, latent failure, and common cause failure, following certain clauses from ISO 26262 parts 3, 4, 5 and 6.

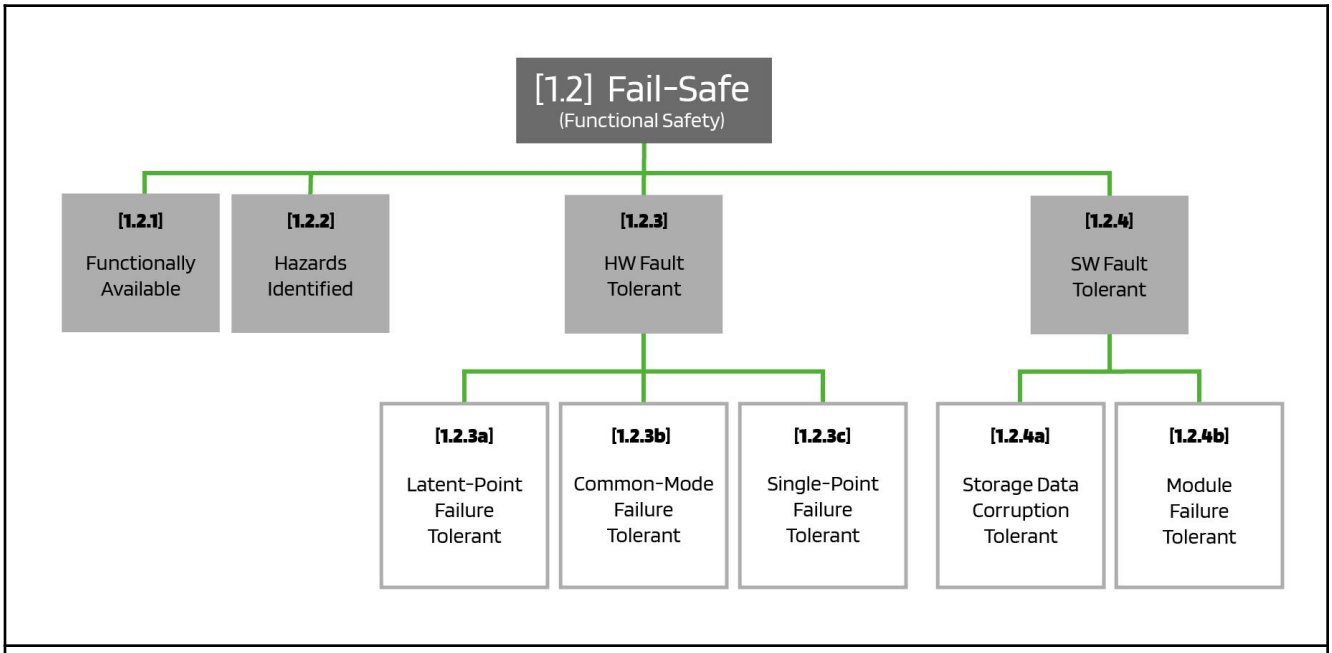

Figure 9: Deep Dive of TuSimple Safety Case Framework section [1.2] Fail-Safe (Functional Safety)

[1.2.1] Functionally Available

Our autonomous trucks are designed to be fail operational. This means that we need to ensure our safety-critical features (such as lane keeping) are always available while the truck is in autonomous mode and moving until the truck comes to a complete stop. The physical components that support these features must be functionally redundant to achieve fail-operational capability. There are numerous principles that the system components must

comply with to achieve that functional redundancy, such as maintaining the truck's stability, fast switching between redundant components, and the tolerance to the loss of the power generator. The use of legacy L2 ADAS systems as supplementary collision avoidance systems also supports our functional redundancy concept. Our safety-related availability analysis is carried out with the use of fault trees in our system safety analysis.

[1.2.2] Hazards Identified

Starting by analyzing the faults in nominal behaviors, the hazard analysis process examines the effect of those faulty behaviors and identifies the ones which can pose safety risks as safety hazards. Combining those safety hazards with results from data collected from actual driving situations provides the overall list of the hazards to be addressed. We apply the HARA technique as described in ISO 26262-3:2018 and we tailor the risk assessment definitions to reflect the specific exposure based on the real count of events in our ODD, and we reference the severity levels based on the type of potential accidents and impact speeds described in SAE J2980.

[1.2.3] HW Fault Tolerant

The HW components that we use must be tolerant to internal faults. This means that the internal faults within each component are monitored and handled by the system. In addition, before initiating an autonomous mission, we make sure that all HW components and their corresponding safety mechanisms are available and functional, whether done automatically or via pre-trip inspection.

[1.2.3a] Latent-Point Failure Tolerant

Latent-point faults are those that represent the unavailability or non-functionality of the safety mechanisms. Detection and prevention of latent-point failures persists throughout the operation and is achieved automatically via the ADS system, pre-trip inspections, built-in subsystem diagnostics and other startup safety checks. This includes on-board sensor diagnostics, on-board component diagnostics and power network diagnostics.

[1.2.3b] Common-Mode Failure Tolerant

Common-mode failures are those that can lead to loss of functional availability in fail-operational systems. In other words, a single point failure that can cause a total failure of the autonomous

truck's actuation or virtual driver leading to loss of control can be referred to as a common-mode failure. In general, diversified designs, fault domain separation and freedom-from-interference are principles that help eliminate any common-mode failures in the system. The most important components that are the focus on eliminating common-mode failures are the power network, the vehicle control units, the redundant brake system, the redundant steering system and the redundant localization sensors. All of these redundant components are required to ensure the fail-operational capability to ensure our autonomous truck can bring itself to a safe stop in the event of a component's total failure. Our fault tree analysis helps us identify these potential common-cause failures.

[1.2.3c] Single-Point Failure Tolerant

A single-point failure (SPF) is one that independently can lead to the violation of one or more safety goals. In other words, an SPF without any safeguards or safety monitors in place can cause our autonomous truck to have total loss of control. For that reason, we perform extensive safety analysis on the system, HW components and implement safety monitors to detect and handle every single point of failure that is detectable.

[1.2.4] SW Fault Tolerant

The SW components that we use must be tolerant to internal faults. This means that the internal faults within each SW component must be monitored and handled by the system. Each SW module contains a diagnostic function that monitors for the presence of single point failures and reports it to a higher-level diagnostic function at the system level that ensures system fail-tolerance to such single point failures by transitioning to redundant HW components with redundant SW.

[1.2.4a] Storage Data Corruption Tolerant

Our high definition maps are stored on a storage device which is accessible by other devices to provide mapping and localization functionality. Accessible storage data requires protection from error corruption mechanisms to ensure tolerance to such errors. We use solid state drives that have built-in error correcting code mechanisms.

[1.2.4b] Module Failure Tolerant

Just like single-point failures, a software module failure can independently lead to the violation of one or more safety goals. In other words, without any safeguards or safety monitors in place it can cause our autonomous truck to have total loss of control. For that reason, we perform extensive safety analysis on the system SW and algorithm modules and implement safety monitors to detect and handle every single point of failure that is detectable. We endeavor to follow principles of freedom-of-interference as much as possible and as applicable to partition safety-critical software from non-safety-critical software.

[1.3] Sufficient (SOTIF)

SOTIF is a concept that proves the absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or by reasonably foreseeable misuse by persons. We apply SOTIF at the vehicle level and the realization level using a suite of methods and tools such as HARA and STPA to identify functional insufficiencies, technical limitations and/or operator misuses.

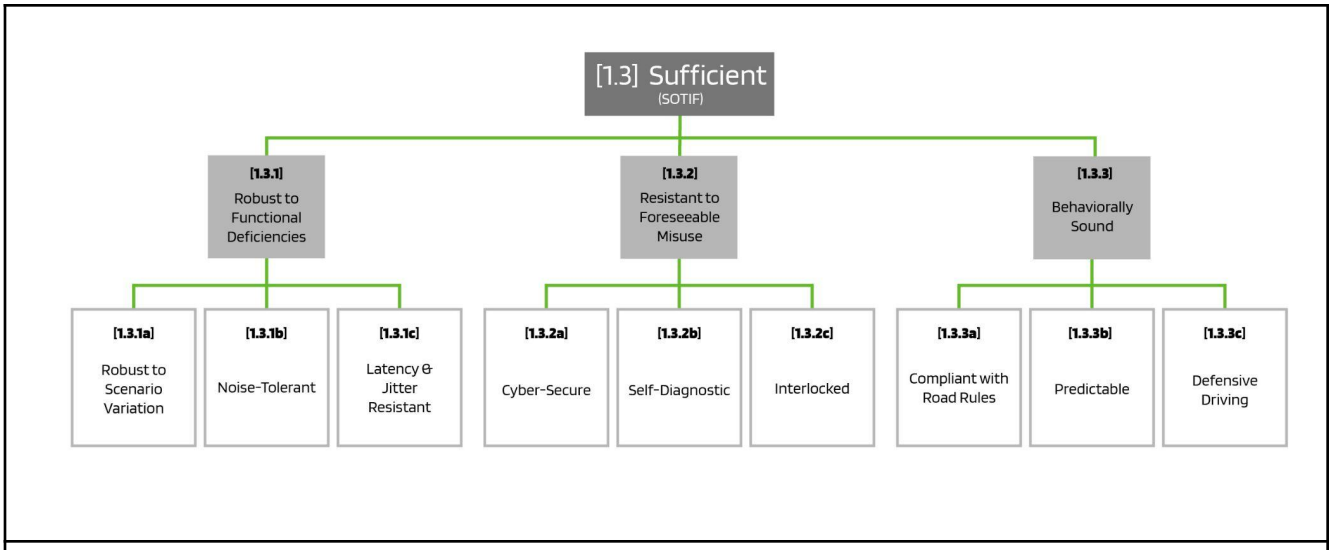

Figure 10: Deep Dive of TuSimple Safety Case Framework section [1.3] Sufficient (SOTIF)

[1.3.1] Robust to Functional Deficiencies

The safety of our core algorithms requires the ability to perform at the same level of required performance under all potential scenarios within our operational design domain. In addition, knowing that the environment does not always permit the full reception of signals or objects (due to occlusion for example), the consistency in being able to localize our autonomous truck or perceive partially seen objects is key to providing that level of performance. In addition, varying scenarios and types of objects can require different demands of performance from our compute unit which may result in varying system latencies. For that reason, we take into consideration the object type variations and scenario variations.

[1.3.1a] Robust to Scenario Variation

We know that driving on public roads within the same operational design domain every day does not always reveal the same scenarios that we commonly encounter. For example, we may encounter locations where there could be a total loss of satellite signals, occlusion to certain objects or complicated navigations through traffic patterns that our autonomous truck would have to handle. Our trucks must perform robustly in all possible variations of scenarios and under all possible object perceptibility. Using STPA coupled with regeneration and simulation tests, we identify those scenario variations and validate our system's robustness to them. This includes scenarios in the absence of Global Navigation Satellite System (GNSS) based localization, varying vehicle types, and situations of competing traffic handling priorities.

[1.3.1b] Noise-Tolerant

There are different types of noise that our system may encounter; there are areas or scenarios where satellite signals may be weakened, jammed or even lost. Objects that move dynamically in the environment can obscure certain objects of interest required for perception. The electric and electronic signals can be susceptible to lots of signal noise or electromagnetic interference. And the dataset that we use to train our perception algorithms may contain data that will not match that of road data. For all those reasons and more, we design and test our algorithms to be tolerant to all the noise such that it does not impact its performance. That includes noise to GNSS signals, partially occluded objects, perception sensor blockage, and signal propagation noise on electronic communication networks and embedded signals.

[1.3.1c] Latency & Jitter Resistant

When multiple HW components (sensors, actuators, compute units) are part of the same system, keeping all components working together for the system to resonate at the same frequency becomes a challenge. Different sensors may operate at different frequencies, signal frames may be dropped, actuators may not receive every control command on time. For this reason, we design our entire system to resist propagation of latency and jitter effects from affected components to the behaviors of the integrated system. That includes latency in GNSS sensor signals, perception sensor signals, latency in processing and tracking objects detected and trajectories generated.

[1.3.2] Resistant to Foreseeable Misuse

Every system that interfaces with human actors is susceptible to misuse. Our system involves sensors that receive information, perceive information, and allows for reception of remote human

commands. Since sensors are upstream to our system modules, any external sensor threat can cause downstream issues with how the algorithm modules use the sensor information correctly. At the same time, any security-intrusions to our oversight systems may cause our operations team to become incapable of commanding safe stops to our autonomous trucks. We design our system with appropriate security measures, coupled with interlock mechanisms to prevent random access.

[1.3.2a] Cyber-Secure

We have analyzed the entry points of data communication to our system, assessed the security risks accordingly and implemented measures for those security risks. Furthermore, we place these measures under penetration testing to ensure functionality.

[1.3.2b] Self-Diagnostic

The autonomous behavior of our truck is delivered through the ADS pipeline including: localization and pose, perception, prediction and planning, and control modules. SPF of those modules are analyzed, and self-diagnostics measures are designed to detect and report such failures to the system. Meanwhile, those modules rely heavily on various artificial intelligence techniques, which are statistical in nature. Output sanity check is included in the self-diagnostics in an effort to eliminate the unlikely event of an outlier behavior.

[1.3.2c] Interlocked

Interlocks provide the ability to safely reconfigure the software in the autonomous truck between fully autonomous mode, where all human-intervention capabilities must be disabled, and supervised autonomous mode, where human driver intervention may be needed. We design our system such that when the ADS is in fully autonomous mode, human driver input to ADS is completely disabled, and when ADS is in supervised autonomous mode, the human driver is able to disengage the ADS by interacting with the HMI, including via steering, braking or acceleration pedal.

[1.3.3] Behaviorally Sound

An important prerequisite for road safety is the mutual awareness road users and pedestrians have of the likely behaviors of others present in the ODD. Adhering to road rules and regulations,

driving in a predictable manner, and following defensive driving best practices are proven approaches to minimizing the probability of ending up in dangerous traffic situations. Our autonomous trucks, while complying with traffic laws and regulations, must also do so in a smooth and expected manner, and must exercise the necessary caution to avoid unnecessarily ending up in dynamically challenging driving scenarios. At the same time, it is possible that other road users may drive erratically or in violation of traffic laws, and may themselves create situations in which a collision with our autonomous truck may be unavoidable. In such cases, our autonomous trucks must attempt to minimize the level of injury and the probability of fatality, regardless of perceived liability for the accident, by reducing the impact speed of any imminent collisions.

[1.3.3a] Compliant with Road Rules

Our autonomous truck is designed to comply with the federal and state traffic laws and regulations. Those traffic laws and regulations are parsed and implemented in all aspects of the ADS including: localization and pose, perception, prediction and planning, and control. Our track testing provides appropriate validation to further ensure the compliance of truck autonomous behavior to relevant traffic laws and regulations.

[1.3.3b] Predictable

Under the worst case scenarios, any of the modules of the ADS could lose its functionality. However, ADS as a system should still provide predictable autonomous driving behavior according to the safety design. The predictable behavior of our autonomous truck under the worst case scenario is based on the analysis of failure modes of ADS modules under such scenarios, so that ADS modules under stress can detect and report failures within a designated time frame, and follow a predefined degrading strategy. Our track testing then further validates the design and implementation of those mitigation strategies. Thus our autonomous truck is designed to be able to provide predictable behavior under the worst case scenarios. according to the safety design.

[1.3.3c] Defensive Driving

Non-compliant drivers and road users are the ones whose behaviors violate traffic laws and regulations. When encountering non-compliant drivers and road users, our autonomous truck is designed to detect and identify them early, and perform additional defensive driving maneuvers to avoid or reduce exposure to the non-compliant drivers and road users. Some examples include,

changing lanes to avoid non-compliant road users that are on the neighboring lanes, avoiding parallel driving scenarios by accelerating, decelerating, or changing lanes, and aggressively slowing down when neighboring road users invade our lane suddenly.

[1.4] Proven (Substantive Safety)

Substantive safety focuses on the level of safety in the real world. This is in comparison to nominal safety which represents the "textbook" version of a safety standard. Nominal safety provides experience from parallel industries, which is helpful. However, at TuSimple, we focus on substantive safety because we are pioneers in a new industry.

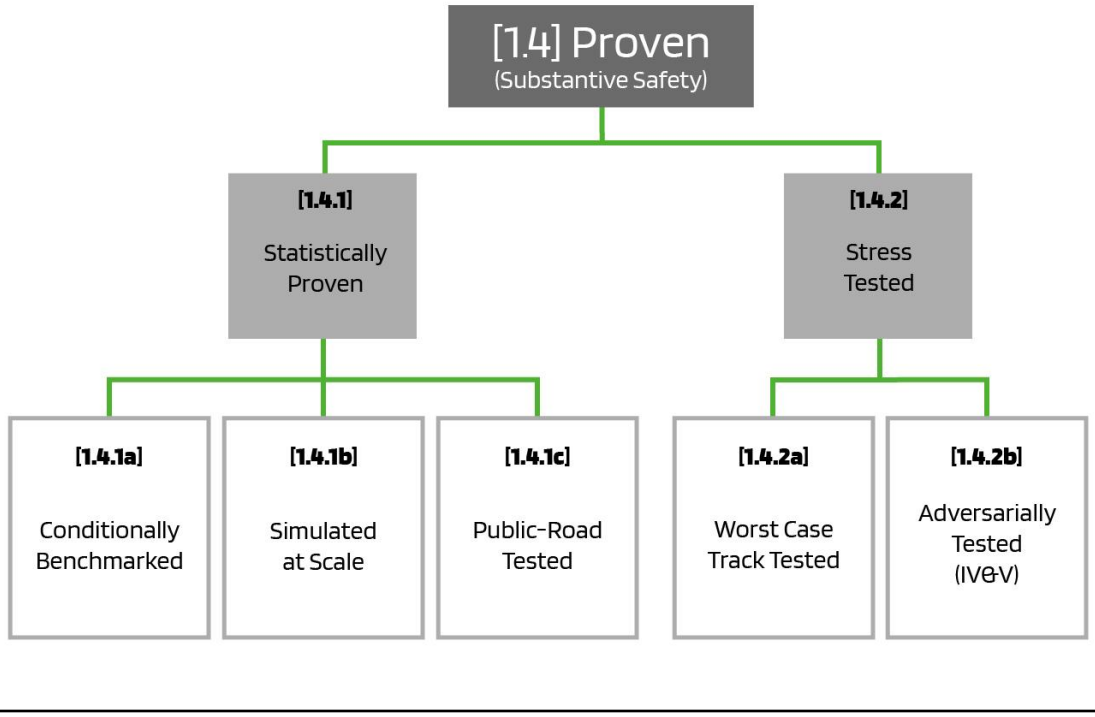

Figure 11: Deep Dive of TuSimple Safety Case Framework section [1.4] Proven (Substantive Safety)

[1.4.1] Statistically Proven

The substantive safety approach emphasizes the statistical analysis of the effect of the behavior of individual processing nodes on the overall ADS decision pipeline from perception to control. In

the case of improving robustness to processing latency and jitters of individual nodes, based on the statistical analysis of node behavior on ADS at system level, the heuristic optimization approach is designed to be able to resolve and relax local timing constraints to accommodate node level latency and jitters, while maintaining system performance with improved reliability and safety.

[1.4.1a] Conditionally Benchmarked

The accumulated road test datasets are annotated and categorized according to location, environment condition, physical infrastructure, and driving situations. Conditional benchmarking is capable of generating test scenarios based on the required permutation of annotation categories from the accumulated road test datasets, to test and verify newly implemented ADS features. Conditional benchmarking is part of the ML platform designed to probe ADS with statistically significant test scenarios across the entire statistical distribution to achieve full ODD coverage.

[1.4.1b] Simulated at Scale

Simulation is used to provide test scenarios for virtual tests on ADS for quick design verification and feedback, as well as providing test scenarios not feasible to be created in real life. The cloud infrastructure enables TuSimple to scale testing to cover scenarios with full parameter dimensions and full parameter ranges to ensure ODD coverage. Our autonomous driving features are probed and verified with simulation at scale before entering track tests and road tests. This technology enables faster design iteration and greatly improves product quality.

[1.4.1c] Public-Road Tested

Public-road test is the final validation step for ADS features after simulation tests and conditional benchmark tests have been completed and passed. The public-road test should be performed according to the approved test plan and using approved test specifications. It is carried out on public-roads according to government regulations.

[1.4.2] Stress Tested

Based on thorough analysis of the ADS, track and simulation tests are performed to test ADS under known worst case scenarios or to test known ADS weaknesses with adversarial test

scenarios. These tests include the most challenging scenarios for our localization, perception, planning and control logic that stress our system and push it to its performance limits.

[1.4.2a] Worst Case Track Tested

The worst case track test is a track test designed to provide the most severe operation situations for ADS, usually involving multiple triggers for known failure modes. Our successful worst case track test results provide validation evidence for ADS to meet particular worst case safety design requirements.

[1.4.2b] Adversarially Tested (IV&V)

Adversarial tests are a set of tests created by an independent verification and validation team that have the specific intent to detect and highlight weaknesses in the assumptions underlying the system implemented by the hardware, software and autonomy teams. Stated differently, these tests are designed to probe the limits of performance of the implemented design and to directly challenge the robustness of the ADS. For these tests, heavy emphasis is placed on the autonomy stack, safety and redundancy subsystems, and the health monitoring subsystem, given their safety-criticality, but they also implicitly test key aspects of broader system performance.

Our successful passing of adversarial tests is part of the verification and validation process.

[2] Does the Driver-Out Technical Operations Assure Operational Safety?

Along with developing a robust organizational structure and internal best practices, we have also instituted a safety case framework specifically for assessing and ensuring safety throughout all of our operations systems. The operations side of our safety case framework works to ensure the success of our second safety principle, that Driver-Out Operations are safe and that each aspect of our operations are *prepared* and *proven*.

This function bolsters all aspects of our operations and serves to create safe processes and procedures spanning across our employees (both engineers and operators), truck oversight as well as internal operational hazard analyses and risk assessments. It aims to ensure that we have monitored and triaged every disengagement that has taken place, assessed the level of safety risk and assigned it for resolution by respective engineering teams and continue to do so. Further, it focuses on our tools, processes and real-time operations. It endeavors to ensure that we have rigorously tested our chase vehicle HMI, redundant communication channels as well as MRC maneuvers to ensure MRC commands can be sent intuitively to our autonomous truck and ensure its capability to come to a complete stop.

Our safety teams have been working closely with all departments and across all levels of our organization to prove the safety of our Driver-Out truck based on our safety case framework and they will continue to do so leading up to our Driver-Out Pilot.

[2.1] Prepared (Training)

This refers to the preparedness of our safety operations. This includes ensuring that our employees who are providing oversight to the operations of our trucks during the driver-out demonstration are regaining control whenever a transition to a safe stop needs to occur in an effort to guarantee the safety of road participants.

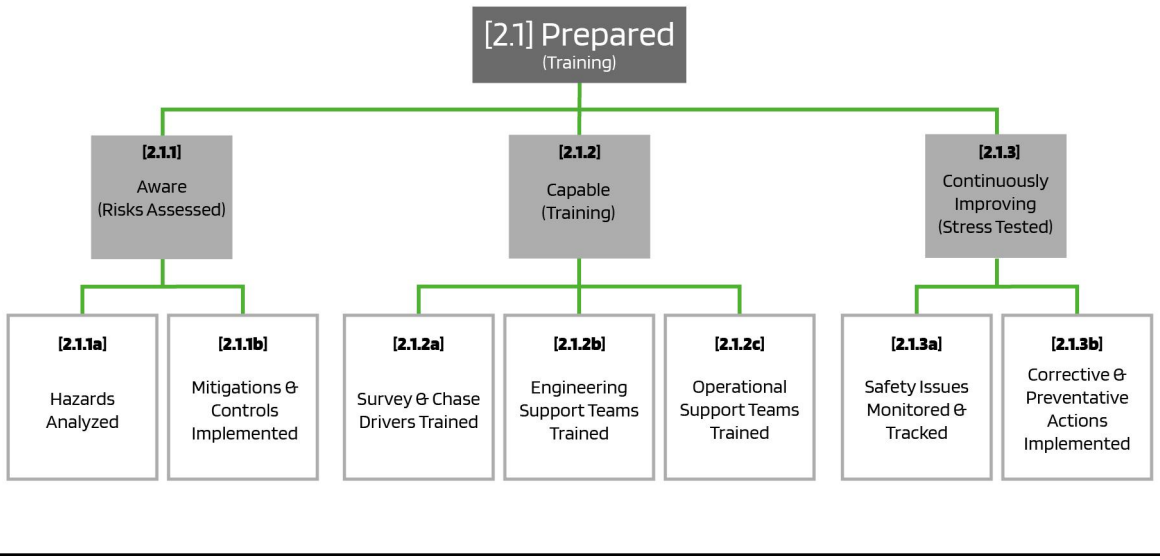

Figure 12: Deep Dive of TuSimple Safety Case Framework section [2.1] Prepared (Training)

[2.1.1] Aware (Risk Assessed)

Based on a thorough analysis, potential safety risks due to Driver-Out operation have been examined, such as truck pre-trip inspection, HW and SW self test, ADS version check, truck configuration check, post-trip inspection and trip summary report, etc. The causes for safety risks are identified, and mitigation plans are implemented.

[2.1.1a] Hazards Analyzed

Our hazard analysis examines potential causes that could lead to safety risk for the Driver-Out operation, identifies the hazards, and defines safety requirements to mitigate operational hazards. A few examples of our hazard analysis results are the checklist for pre-trip inspection,

HW and SW initiation sequences, ADS loading sequences, truck configuration checklist, the post-trip inspection checklist, summary report template, etc.

[2.1.1b] Mitigations and Controls Implemented

Based on our operational HARA, our documentation to show mitigation and control measures are implemented according to safety requirements. The results of mitigation and control implementation are the records of pre-inspection checks, HW and SW initialization records, ADS version check records, records of ADS loading results, records of truck configuration check results, records of post-trip inspection results, trip summary report records, etc.

[2.1.2] Capable (Training)

Our operations personnel have undergone necessary training to master the process and tools and are able to carry out the mitigation and control procedures successfully, including performing the entire routine of various checks and procedures according to the operational instructions.

[2.1.2a] Survey & Chase Drivers Trained

Our survey and chase drivers have undergone necessary training to master the process and are able to carry out the mitigation and control routines successfully.

[2.1.2b] Engineering Support Teams Trained

Our engineering support teams have undergone necessary training to master the process and are able to carry out the mitigation and control routines successfully.

[2.1.2c] Operational Support Teams Trained

Our operational support teams have undergone necessary training to master the process and are able to carry out the mitigation and control routines successfully.

[2.1.3] Continuously Improving (Stress Tested)

We are continuously improving our safety capabilities by identifying and resolving new safety issues based on our simulation and road tests. Our engineering process emphasizes fast

development iterations of our ADS software stack by continuously resolving feedback from our ongoing road tests through thorough analysis and timely resolution of safety critical disengagements, as well as from ongoing ADS module level regression tests.

[2.1.3a] Safety Issues Monitored & Tracked

Our periodic meetings of multi-discipline teams analyze ongoing safety issues in terms of safety-critical disengagements and ADS module regression tests. The root cause of safety issues are identified and tracked until their resolutions are implemented and documented.

[2.1.3b] Corrective & Preventive Actions Implemented

Our resolutions of safety issues are provided in the implementation of fixes or new features, designed to eliminate the disengagements and pass the regression and track tests.

[2.2] Proven (Stress Tested)

Our operations team practices what they are trained on to prove that our driver-out demonstration is operationally safe. This is a demonstration of the preparedness of our operational safety. It focuses on our tools, processes and real-time operations.

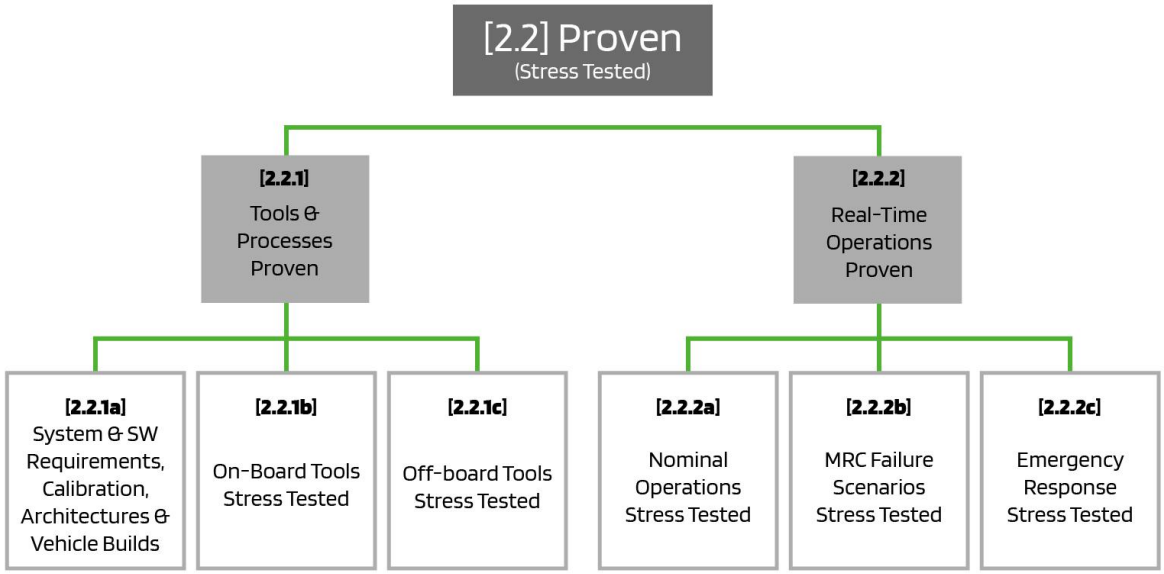

Figure 13: Deep Dive of TuSimple Safety Case Framework section [2.2] Proven (Stress Tested)

[2.2.1] Tools & Processes Proven

The tools used in loading, monitoring, debugging and testing ADS operations are proven for safety applications according to ISO 26262 standards. Operation processes as well as design processes (requirements management, change management, document management, configuration management, etc) are proven to ISO 26262 safety standards.

[2.2.1a] System & SW Requirements, Calibration, Architectures & Vehicle Builds

Our engineering processes including requirements management, calibration procedures, and documentation management are proven to ISO 26262 safety standards.

[2.2.1b] On-board Tools Stress Tested

Our on-board modules and tools are tested for known failure modes and evidence of successful passing is provided. Safety certification is provided for on-board tools according to ISO 26262.

[2.2.1c] Off-board Tools Stress Tested

Our SW tools are monitored for tool incidents. Safety certification should be provided for SW tools according to ISO 26262.

[2.2.2] Real-Time Operations Proven

In order to demonstrate that we have successfully integrated, verified and validated our HW and SW components into our autonomous driving system and to demonstrate our operational capabilities and readiness, we put our autonomous truck through a final public road test and track test to demonstrate absence of malfunctions, failures, functional insufficiencies and misuses that could potentially to lead to disengagements. Additionally, the test miles we accumulate at this stage are tracked until we achieve our target validation runs required before finally performing the driver-out demo.

[2.2.2a] Nominal Operations Stress Tested

Our engineering operations are put to the final test before the actual demonstration on public roads by driving the intended route autonomously with all operations staff in place. The intention is to nominally demonstrate our operational capability and readiness before removing the safety driver from behind the steering wheel. A successful nominal operational test includes performing all the pre-start of mission inspections, HW components power up, SW start up, and autonomy engagement and final truck recovery after the successful completion of the mission without any disengagement.

[2.2.2b] MRC Failure Scenarios Stress Tested

Our oversight operations are put to the final test before the actual demonstration on public roads by testing our remote MRC command capabilities with all operations staff in place. The intention is to demonstrate our oversight operational capability and readiness before removing

the safety driver from behind the steering wheel. A successful MRC failure scenario test includes our chase vehicle performing all remote MRC commands when requested by the survey vehicle.

[2.2.2c] Emergency Response Stress Tested

The emergency response teams are put to the final test before the actual demonstration on public roads by testing their response actions and timely reactions when their intervention is required. The situations that call for their intervention range from public road shutdown on an MRC event to providing the medical and safety needs in the very unlikely event of an accident involving the autonomous truck. The intention is to demonstrate their capabilities and readiness to address those very unlikely situations prior to the final demonstration on public roads.

E. Summary

This Safety Framework has explained the motivations and organization that TuSimple has developed in an effort to ensure the safety of our Driver-Out Pilot on public roads. The safety case framework derives directly from the two core principles below, and defines the corresponding sub-claims that must necessarily be true in order to claim successful adherence to these principles.

[1] Driver-Out Trucks are Safe to Operate Autonomously on our Driver-Out Route
[2] Driver-Out Technical Operations Assure Operational safety

Our Driver-Out Pilot is being developed in accordance with the requirements documented in this Safety Framework, with quantitative evidence documented and collated to demonstrate this compliance. We believe that this Safety Framework has provided TuSimple with a powerful tool to ensure that our Driver-Out Pilot is free of uncontained hazards and is safe for execution on public roads.

F. Appendix

Glossary of Terms

ADAS	Advanced Driver Assistance Systems
ADS	Autonomous Driving System
AV	Autonomous Vehicle
DFMEA	Design Failure Mode and Effects Analysis
FIT	Failure In Time
FMCSA	Federal Motor Carrier Safety Administration
GNSS	Global Navigation Satellite System
HARA	Hazards Analysis and Risk Assessment
HIS	Hersteller Initiative Software
HMI	Human Machine Interface
HW	Hardware
IV&V	Independent Verification and Validation
LiDAR	Light Detection and Ranging
ML	Machine Learning
MRC	Minimal Risk Condition
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
SEooC	Safety Element out of Context
SOTIF	Safety of the Intended Functionality

SPF	Single Point Failure
STPA	System Theoretic Process Analysis
SW	Software
TARA	Threat Analysis & Risk Assessment