



2026 Product Liability & Complex Torts Practice Group Seminar

September 16-18, 2026

**FROM TANGIBLE GOODS TO DIGITAL CODE:
THE EXPANDING SCOPE OF PRODUCT LIABILITY**

"One cannot reach out and touch them."

M. Tucker Blaser

Moderator

BROWN & JAMES, P.C.

Belleville, Illinois

tblaser@bjpc.com

Megan Sterchi Lammert

BAKER STERCHI COWDEN & RICE LLC

Kansas City, Missouri

msterchi@bakersterchi.com

One cannot reach out and touch them.

In many jurisdictions, product liability law has traditionally been limited to tangible goods. That boundary is now being tested. In a recent appellate decision, a court allowed a wrongful death claim to proceed against a rideshare company, concluding that a mobile application could qualify as a “product” for purposes of product liability law. This chapter examines how courts across the country are rethinking what constitutes a product in an increasingly digital economy. It traces the doctrinal history of the term, analyzes the anchoring Missouri authority and the national split that surrounds it, and then turns to the problem this practice group will find most pressing: not the platform whose product is an app, but the manufacturer whose tangible goods now ship with software inside them. It closes with practical guidance for evaluating risk, advising clients, and developing defense strategies as product liability law continues to expand beyond physical goods.

I. The Line That Is Being Redrawn

Nearly every manufacturer and retailer now sells a product that contains code. Commercial equipment reports its own diagnostics to the cloud. Vehicles update themselves overnight. Medical devices run on embedded firmware that is patched years after the sale. For a growing share of consumer goods, a mobile application is the interface through which the customer actually configures, orders, or operates the thing they bought.

For sixty years, American product liability law has drawn its central line at tangibility. That line is now being redrawn, and it is being redrawn in litigation against the companies in this room. The question is no longer academic. It determines whether a plaintiff pleads strict liability or ordinary negligence, whether a design-defect theory reaches an entire product line or a single unit, and whether a company's own engineering records become the design file in a products case.

What follows is organized in three movements. Sections II through IV map the doctrine: where the definition of “product” came from, what Missouri's decision in *Ameer v. Lyft* did to it, and how the rest of the country is dividing. Section V turns to the harder problem for manufacturers — what happens when the code is not the product but a component of it. Sections VI and VII offer practical guidance and a look at the European regime that takes effect three months after this seminar.

II. What Is a “Product”? The Doctrine and Its Built-In Ambiguity

The traditional definition, and how we arrived at it

Before 1960, American courts did not recognize strict liability in tort for harm caused by defective products, particularly where there was no privity of contract between plaintiff and defendant. Plaintiffs were required to prove causal negligence. That changed in 1963, when the California Supreme Court decided *Greenman v. Yuba Power Products, Inc.*, 59 Cal. 2d 57 (1963). *Greenman* was injured by a woodworking tool, and the court recognized strict liability for defects in products placed in the market. It did not define “product.” It did not need to.

Two years later the Restatement (Second) of Torts § 402A imposed strict liability on one who sells “any product in a defective condition unreasonably dangerous to the user or consumer,” provided the seller is engaged in the business of selling such a product and the product is expected to and does reach the user without substantial change in its condition. Section 402A likewise supplied no definition. What its comments supplied instead was a list of examples: an automobile, a tire, an airplane, a grinding wheel, a water heater, a gas stove, a power tool, a riveting machine, a chair, an insecticide.

From that list, courts inferred the working rule under which the field operated for decades — a product is tangible personal property placed into the stream of commerce. It is worth pausing on how that rule came about, because the point matters to every argument that follows. The tangibility requirement was never reasoned to. It was assumed. Section 402A was drafted around physical goods because in 1965 physical goods were the only things injuring people in the ways tort law then cared about. No court sat down and held that intangibility should defeat a claim; the question simply never presented itself. When a court today invokes “the traditional rule requires tangibility,” it is invoking an inference rather than a holding — and plaintiffs' counsel know it.

Restatement (Third) of Torts § 19 — a definition, and an escape hatch

The Third Restatement, published in 1998, finally defined the term. Section 19(a) provides that a product is “tangible personal property distributed commercially for use or consumption.” The definition has two components: a descriptive one (tangible personal property) and a purposive one (commercial distribution for use or consumption).

The same subsection then gives much of it back. It continues: “Other items, such as real property and electricity, are products when the context of their distribution and use is sufficiently analogous to the distribution and use of tangible personal property that it is appropriate to apply the rules stated in this Restatement.” The comments cut in both directions. Comment c excludes human blood and tissue, which are plainly tangible. Comment d permits electricity to be treated as a product although it is not. Comment d also excludes ideas, information, and data. Comment f excludes services.

Worked through, § 19 yields an uncomfortable conclusion: tangibility is neither necessary nor sufficient. Blood is tangible and is not a product; electricity is intangible and is. What the drafters actually created is a functional test wearing a descriptive test's clothing. The operative question is whether the context of an item's distribution and use is sufficiently analogous to that of goods. Once that is understood, the case law surveyed below becomes considerably more predictable. Courts are not truly asking whether a plaintiff can touch the thing. They are asking whether it reaches the user the way a manufactured good reaches a user, and whether the defendant occupies the position a manufacturer occupies.

The four categories § 19 created

In application, § 19 produced four categories: services; tangible goods; intangibles sufficiently analogous to goods; and items akin to ideas, content, and expression.

The exclusion for ideas and expression has generated the most instructive case law. The seminal decision is *Winter v. G.P. Putnam's Sons*, 938 F.2d 1033 (9th Cir. 1991), in which mushroom enthusiasts became seriously ill after relying on a reference book that misidentified a species as safe. The Ninth Circuit held that the content of the book was ideas, thoughts, and expression, and could not form the basis of a product liability claim. The Sixth Circuit reached the same conclusion as to violent video games in *James v. Meow Media, Inc.*, 300 F.3d 683 (6th Cir. 2002), holding that intangible thoughts, ideas, and expressive content are not products.

Two features of that line deserve attention. First, *Winter* expressly reserved the question now before us, observing that “computer software that fails to yield the result for which it was designed” may be a different matter. That sentence, written in 1991, anticipated the entire modern dispute. Second, the exclusion is narrower than it appears. In *Aetna Casualty & Surety Co. v. Jeppesen & Co.*, 642 F.2d 339 (9th Cir. 1981), the Ninth Circuit permitted strict liability for a defective aeronautical navigation chart — paper and ink, pure information — because pilots relied on the chart for its physical, functional correspondence to the world rather than for the ideas it contained.

The line, in other words, is not tangible versus intangible. It is expressive versus functional. That distinction does most of the work in the decisions that follow.

III. *Ameer v. Lyft: Missouri's Question of First Impression*

The incident and the pleadings

On the evening of September 28, 2020, Andrew D. Ameer, a twenty-seven-year-old Lyft driver, was carjacked and fatally shot in St. Louis, Missouri. According to the petition, two fifteen-year-olds — both ineligible under Lyft's terms to hold accounts — created an account using a false name, a false email address, and an anonymous form of payment. From that fraudulent account they requested a ride at approximately 11:00 p.m. from a neighborhood miles from their actual location. The Lyft App routed the request to Mr. Ameer, who accepted. Upon his arrival the passengers announced a robbery, drew firearms, and shot him.

The framing plaintiff selected is critical and repays close study. She did not plead that Lyft operated an unsafe business. She pleaded that specific technological capabilities were absent from the software: adequate age, identity, and violent-propensity verification technology; adequate GPS coordinate verification technology; and adequate data-pattern analysis to identify fraudulent or violent riders. Framed that way, the omissions read as missing safety features — the guard that should have been on the machine. As will be seen, that pleading choice largely determined the outcome.

FROM TANGIBLE GOODS TO DIGITAL CODE: THE EXPANDING SCOPE OF PRODUCT LIABILITY



The petition asserted three product liability counts against Lyft — strict liability for defective design, negligent design, and negligent failure to warn — along with counts for negligent training and general negligence.

The trial court

Lyft moved to dismiss on a single threshold proposition: the Lyft App is a service, not a product, and absent a product there can be no product liability claim. On July 6, 2023, the Circuit Court for the City of St. Louis granted the motion and dismissed the petition in its entirety. As to the product counts, the court's reasoning was brief and, on the traditional rule, unremarkable: under Missouri case law and the Restatement (Third), a product is something tangible; the Lyft App is not tangible; and under *Hobbs v. Boy Scouts of America, Inc.*, 152 S.W.3d 367 (Mo. App. W.D. 2004), product liability theories do not apply to services.

The question of first impression

On appeal, the Missouri Court of Appeals for the Eastern District framed the issue as one of first impression: “Whether a mobile ridesharing application such as the Lyft App may be considered a product for purposes of a recognizable product liability claim is an issue of first impression for Missouri Courts.” *Ameer v. Lyft, Inc.*, 711 S.W.3d 534 (Mo. App. E.D. 2025), transfer denied (Mo. May 27, 2025).

The pre-*Ameer* landscape offered no useful category. Missouri had applied product liability law to cigarettes, automobile seats, airplane parts, and a chain and ratchet system. It had declined to apply it to a scouting program, the supply of electricity, medical services, and services generally. Physical things occupied one side of the ledger and services the other, with nothing in between.

The holding and the two-part test

The Court of Appeals reversed the dismissal of both the product counts and the negligence counts. Its central observation was that the categories themselves were the problem: “It is not appropriate to force the Lyft App into one of the pre-existing legal categories in Missouri product liability law.” 711 S.W.3d at 544-45. The App, the court noted, “is multi-faceted and involves readily-available new technology.”

The court then announced a two-part test:

First, a mobile ridesharing application such as the Lyft App is a product subject to product liability law if the facts alleged establish that the application has sufficient similarities to a tangible product; and

Second, in order to survive a motion to dismiss a product liability claim in a wrongful death case, the plaintiff must allege that the decedent's death resulted from a defect in the application itself — whether in the design or in the functionality of the app — rather than from problems with the application developer's services or some other aspect of its business model.

The two prongs perform materially different work, and the distinction is the single most useful practical point in this chapter. Prong one asks about the defendant: is this company positioned as a manufacturer or is it positioned as a service provider? Prong two asks about the allegation: is the plaintiff complaining about the code, or about how the business is run?

Applying the test

On prong one, the court found the petition sufficient. Lyft designed the App and placed it into the stream of commerce for the general public. Lyft profits from it. The App requires drivers to download it to accept rides and requires the reciprocal act of passengers, creating demand and reliance on both sides. Those facts placed Lyft in the best position to control the risk of harm arising from its own design choices — the position occupied by the designer of a defective tangible product.

The significance of that list for this audience is difficult to overstate. Designed it, distributed it, profits from it, controls the design, created reliance: there is scarcely a manufacturer that the description does not fit. For most defendants, prong one is satisfied on the first page of the complaint.

On prong two, the court held that the plaintiff had adequately alleged a defect in the App itself rather than in Lyft's services or business model, pointing to the absence of verification, GPS, and data-pattern technology. Prong two, accordingly, is where these cases are won and lost.

The dissent

The dissent agreed with the test and disagreed entirely with its application, and it supplies the defense with its most serviceable language. In the dissent's view, the alleged failure to screen and verify passengers does not arise from a defect in the application at all; it is a complaint about the business model and about how Lyft has chosen to provide ridesharing services. The defect, the dissent reasoned, must arise from something the application actually did defectively. By the plaintiff's own admission, identifying the assailant in advance was never something the Lyft App was designed or intended to do.

Stated more generally: a design defect claim presupposes that the product was supposed to do something and did it badly. The Lyft App was designed to match riders with drivers, and it performed that function without flaw. To say it should also have functioned as a background-check service is not to identify a defect; it is to ask for a different product. That formulation is the analytical fault line on which the defense of these cases turns.

After *Ameer*

Two decisions from the Eastern District of Missouri, issued four months apart, demonstrate that the test has genuine content. In *Newman v. Lyft, Inc.*, 2025 U.S. Dist. LEXIS 218111 (E.D. Mo. Nov. 5, 2025), the court held that amendment to add a product liability claim was not futile, finding it sufficient that Lyft possessed data that would have prevented the shooting and that the perpetrator exploited defects in the App to request a ride he should never have been permitted to request.

In *Newman v. Uber Technologies, Inc.*, 2025 U.S. Dist. LEXIS 132209 (E.D. Mo. July 11, 2025), the court reached the opposite result, in one of the first decisions limiting *Ameer*'s reach. Counsel should be precise about the ground: *Newman v. Uber* turns on the duty analysis rather than the two-part product test. The decedent was injured on a competitor's platform, and the theory was that Uber breached a duty by failing to share information about the perpetrator with Lyft. The court rejected it. Uber never brought the decedent into contact with the perpetrator, and a cross-platform duty of that kind would have no stopping place — it would extend to limousine services, bus lines, and taxicabs, and would oblige every business to publicize risk information to every other business whose customers it might encounter. That limiting-principle reasoning transfers well beyond the rideshare context and belongs in the defense brief bank.

IV. The National Patchwork

Courts declining to treat platforms as products

Soc. Media Cases, Nos. JCCP 5255, 22STCV21355, 2023 WL 6847378 (Cal. Super. Ct. Oct. 13, 2023). Claims against Facebook, Instagram, Snapchat, TikTok, and YouTube premised on video-recommending algorithms. The court held the platforms “are not tangible,” that “one cannot reach out and touch them,” and that they are not “analogous to tangible personal property.” Services, it observed, usually involve a direct relationship with the customer and are shaped to the customer rather than mass-produced — a description the court found fit algorithms individually tailored to each user.

Jackson v. Airbnb, Inc., 639 F. Supp. 3d 994 (C.D. Cal. 2022). A platform connecting users is more akin to a service than a product. Notably, the court distinguished *Bolger v. Amazon.com, LLC*, 53 Cal. App. 5th 431 (2020), where Amazon's involvement in marketing, storing, shipping, and controlling the transaction placed it within the chain of distribution for a tangible product.

Ziencik v. Snap, Inc., 2023 WL 2638314 (C.D. Cal. Feb. 3, 2023). Although the plaintiffs adequately alleged a duty, Snapchat was held to be more like a service than a product.

See also *Thiam v. Lyft, Inc.*, 2022 WL 1008494 (Ill. Cir. Ct. Dec. 20, 2022); *Baxter-Armentrout v. Roberts*, 2022 WL 22869345 (Fla. 15th Cir. Ct. Aug. 29, 2022); *Behuet v. Uber Techs., Inc.*, 2022 WL 20318684 (Cal. Super. Ct. July 13, 2022).

Courts treating platforms as products, or permitting the claim to proceed

Doe v. Lyft, Inc., 756 F. Supp. 3d 1110 (D. Kan. 2024). The rideshare system has characteristics of both a product and a service, and neither party's exclusive characterization was persuasive. The court held the allegations sufficient to establish the App as a software or algorithmic product with sufficient similarities to a tangible product, emphasizing the plaintiff's obligation to trace how the alleged flaw in the app itself led to the specific injury.

In re *Uber Technologies, Inc., Passenger Sexual Assault Litigation*, 745 F. Supp. 3d 869 (N.D. Cal. 2024). The court reasoned that tangibility does not really mean “having some effect on physical reality”; the better analysis considers whether the context of the app's distribution and use is sufficiently analogous to that of tangible property. Because the app is mass-

FROM TANGIBLE GOODS TO DIGITAL CODE: THE EXPANDING SCOPE OF PRODUCT LIABILITY



distributed through app stores and downloaded onto a device the user owns, procurement resembles bringing home a tangible appliance. The court then compared the alleged GPS defect to an Apple AirTag — a physical product performing precisely the function the app was accused of omitting.

T.V. v. Grindr, LLC, 2024 U.S. Dist. LEXIS 143777 (M.D. Fla. Aug. 13, 2024). Grindr may be a product where the plaintiff alleged the defendant designed the app for its business, made design choices, placed it into the stream of commerce, distributed it in the global marketplace, marketed it, and generated revenue from it.

Brookes v. Lyft, Inc., 2022 WL 19799628 (Fla. Cir. Ct. 2022), holding the app to be a product under Florida law. Florida trial courts have gone both directions.

What is actually driving the split

Read together, the decisions are less inconsistent than they first appear. Three questions predict nearly every outcome, and none of them is really about tangibility.

- 1. Distribution.** Is the item mass-distributed in uniform form, or shaped to the individual user? Software downloaded from an app store in identical form resembles a good. An experience algorithmically personalized to each user resembles a service.
- 2. Control.** Who was positioned to design the risk out? Where the answer is the defendant, courts reach for product liability — this is the express rationale in *Ameer, Doe v. Lyft*, and *T.V. v. Grindr* alike.
- 3. Characterization.** Is the complaint about the code, or about content and conduct? Interface, controls, verification, and safeguards sound in product design. What users said, what was recommended, and how the business was run do not.

The clearest demonstration is *In re Social Media Adolescent Addiction/Personal Injury Products Liability Litigation*, 702 F. Supp. 3d 809 (N.D. Cal. 2023). The court did not hold that the defendant platforms were products, or that they were not. It held that certain alleged defects sounded in product design while others did not — within a single platform. Failure to implement effective age verification, parental controls, and limits on the length and frequency of sessions were analogous to tangible personal property, because a myriad of physical products incorporate parental locks and timers: prescription bottles, television channels. Barriers to account deactivation and deletion were likewise product features, as was the design of image filters, insofar as the claim targeted the filter rather than the content. The recommendation of content was not.

The lesson for defense counsel follows directly. “Is it a product?” is the wrong question, and a defendant that permits the court to frame the inquiry that way has already conceded ground. The right question is whether this particular alleged defect is a design defect — asked defect by defect and count by count.

V. When the Code Is Inside the Box: The Manufacturer's Problem

Every decision surveyed above involves a defendant whose product is the application. Most members of this practice group do not represent that defendant. They represent manufacturers and retailers whose tangible goods contain software — vehicles with driver-assistance systems and telematics, medical devices with embedded firmware and cloud connectivity, commercial and consumer appliances with connected diagnostics, industrial equipment with programmable safety interlocks, and consumer goods whose ordering, configuration, or safety functions reside in an application.

For those defendants the threshold question is already resolved, and resolved against them. Nobody disputes that an espresso machine, a vehicle, or an infusion pump is a product. The contested question is subtler and, in practice, more dangerous: when the harm is traceable to a line of code rather than a piece of steel, does the ordinary design-defect apparatus apply to that code — and if it does, what does a reasonable alternative design look like when the alternative is software? The app cases are increasingly being imported to answer that question.

The hybrid product, and the disappearance of the manufacturing defect

Where software is embedded in a tangible good, the § 19 tangibility debate largely falls away. The product is the integrated whole — hardware, firmware, and interface — and the threshold dismissal available to a pure platform is generally unavailable. What remains is that every doctrine developed for steel now operates against code.

One category behaves peculiarly. Every compiled copy of a software build is identical to its specification; that is what compilation means. There is accordingly no practical analogue to the manufacturing defect claim in software. Every software claim is a design

claim or a warnings claim. That is favorable in one respect and considerably less so in another, because design claims implicate an entire product line rather than a single unit, and a single adverse design ruling therefore scales across every unit in the field.

The second consequence changes client behavior more than any other point in this chapter. The ordinary artifacts of software development — the source repository, requirements documents, the bug tracker, sprint records, release notes — function as the design file in a products case. A ticket reading “known issue, deprioritized for Q3” is, in litigation, a contemporaneous document in which the defendant's own engineers identified a shortcoming and elected not to remedy it. Nothing in traditional manufacturing generates a comparable volume of candid, timestamped commentary on a product's deficiencies.

“You could have coded it differently”: the reasonable alternative design problem

Most jurisdictions require a design-defect plaintiff to prove a reasonable alternative design. For tangible goods that requirement has long supplied a genuine defense. Redesigning a machine guard means retooling a line, requalifying a supplier, and absorbing a capital cost, and juries understand the significance of those facts.

When the proposed alternative is a change to the code, that intuition evaporates. Plaintiff's expert testifies that the fix would have required a few hours of engineering time — no retooling, no supply chain, no capital expenditure — and adds that the defendant has already shipped a hundred updates and could plainly have shipped one more. The feasibility defense, in its traditional form, does not survive that framing.

The defense has to be rebuilt on different ground. Three lines of response are worth developing early:

- Safety-critical software changes are not costless merely because they are downloadable. They require requirements analysis, validation, regression testing across every configuration in the field, and in regulated industries, recertification.

- Plaintiff's expert should be held to specify the alternative design at implementation level rather than aspiration level. What, precisely, would the code have done; on what inputs; with what threshold; producing what output? Experts who can describe the desired outcome frequently cannot describe the mechanism, and the gap presents a Daubert opportunity that is currently underused.

- Every safeguard carries a false-positive cost, and the tradeoff belongs in the record before plaintiff's expert characterizes the proposed fix as free. *Ameer* supplies the clean illustration: an aggressive identity-verification requirement would necessarily have excluded some number of entirely legitimate riders, including those without conventional documentation.

The product that changes after it is sold

Over-the-air updates unsettle a premise embedded in § 402A itself, which contemplates that the product will reach the user “without substantial change in the condition in which it is sold.” A connected product changes — by the seller's own hand — for years after the sale. Four questions follow, and clients should be considering them now.

- Does the capacity to push a remedy create a duty to push it? This is Restatement (Third) §§ 10 and 11 territory — the post-sale duty to warn and the duty to recall — and the analysis was developed for a world in which reaching every owner was difficult and expensive. That premise no longer holds.

- Is a silent update evidence that a reasonable alternative design was feasible all along? Subsequent remedial measure rules offer some protection, but those rules have exceptions and they do nothing about the feasibility inference.

- When is an over-the-air update a recall? NHTSA treats a remedy for a safety-related defect as a recall whether it is delivered at a dealership or over the air, with the attendant notification obligations and civil-penalty exposure. In practice a great many updates are pushed with limited oversight or transparency, and the classification decision is frequently made by engineers rather than counsel.

- Which software version was running on the unit at the time of loss, and can the defendant prove it? In a traditional case the product's condition is established through the build record and is fixed for all time. With a connected product, counsel must be able to establish the configuration of a specific unit at a specific moment. A great many companies cannot, and the capability is far cheaper to build before it is needed.

Summed up: every update is simultaneously a remedial measure, a design change, and a potential admission.

When the application is the front door to a physical good

For retailers and consumer-products companies, a related exposure arises where tangible goods are ordered, configured, or delivered through software, and where the application carries information bearing on physical safety — allergen and dietary flags, handling and temperature instructions, assembly or dosing guidance.

Bolger and *Jackson*, discussed above, appear irreconcilable and are not. Amazon was held within the chain of distribution because the claim concerned a tangible product — a defective laptop battery — and Amazon's role in marketing, storing, shipping, and controlling the transaction resembled a retailer's. Airbnb prevailed because the plaintiff pleaded around the tangible asset entirely, attacking the design of the application rather than the condition of the rental property. The practical lesson runs in both directions: an application can draw a company into the distribution chain for goods it did not manufacture, but where a plaintiff pleads application design in place of product condition, prong two of *Ameer* becomes an asset rather than a liability.

Artificial intelligence as a product component

Garcia v. Character Technologies, Inc., 785 F. Supp. 3d 1157 (M.D. Fla. 2025), is *Ameer*'s analytical cousin, and the parallel is exact. A fourteen-year-old user of an AI companion application died by suicide after extended interactions with chatbot characters. The court held the application to be a product and denied the motion to dismiss the product liability claims, reasoning that the complaint addressed the defendant's content and design choices — including the failure to confirm user ages and the absence of reporting mechanisms — and that the harmful interactions were possible only because of the alleged design defects. It was a product “insofar as the allegations focused on the application itself, rather than the ideas or expressions of A.I.” On duty, the court held that releasing the system to the public created a foreseeable zone of risk that the defendant was positioned to control.

In *Ameer* the court separated the application's design from the defendant's business model. In *Garcia* it separated the system's design from the model's output. In each, the design half of the split is a product. That is the template plaintiffs will use against AI features embedded in tangible goods. They will not attack the model's judgment, because *Winter* and *Meow Media* make that a losing argument about ideas and expression. They will attack the absent guardrail, the threshold set too permissively, and the confirmation step that was never built. Those are design allegations, and design allegations survive.

VI. Practical Guidance

Defending the characterization

1. **Win on prong two, not prong one.** Concede what must be conceded about design, distribution, and profit. The contest is over what the alleged defect actually is.
2. **Attack defect by defect, not case-wide.** *In re Social Media Adolescent Addiction* establishes that a single platform may be a product as to some theories and not others. Resist any framing that resolves the case on a single categorical question.
3. **Reframe absent capabilities as business decisions.** An allegation that the software “should have screened users” is a complaint about whom the company chooses to serve, not about how the code performs.
4. **Hold plaintiffs to intended function.** The *Ameer* dissent's formulation: a defect presupposes that the product was supposed to do the thing it is said to have done badly.
5. **Press the limiting principle.** *Newman v. Uber* demonstrates that courts remain receptive to the argument that a proposed duty has no stopping place.
6. **Attack the alternative design at implementation level.** Require the expert to specify what the code would have done. Consider the timing of the challenge carefully — an early motion to dismiss tests the theory cheaply on the pleadings, while summary judgment develops a record but risks a published adverse ruling on a thin factual showing that will bind the industry.

Before the claim: what to advise clients now

Treat engineering records as design documentation. Establish conventions for recording safety tradeoffs so that a deprioritization decision reads as reasoned engineering judgment supported by analysis, rather than as indifference. This is the highest-value preventive measure available and it costs almost nothing.

- Bring legal into the room where software safety decisions are made, rather than reviewing them after launch. Where a physical product and a digital experience constitute a single offering to the customer, they are a single risk.
- Build the capability to establish software version and configuration for an individual unit at an individual moment, and retain it for the useful life of the product.
- Treat safety-relevant information delivered through an application — allergens, handling, temperature, dosing — as a warnings question governed by warnings law, not as a user-experience question governed by design preference.

Living with the claim

- Frame the characterization from day one. The claim file, the early case assessment, and the instructions to outside counsel should all reflect the product-versus-business-model distinction before the first responsive pleading.
- Anticipate the knowledge gap. Many courts encountering these theories for the first time will benefit from a clear explanation of how software is actually designed, built, and updated. That explanation is worth making affirmatively rather than in response.
- Address coverage early. Design-defect and failure-to-warn theories directed at software raise occurrence, intentional-acts, and professional-services questions that an ordinary automobile or premises claim does not, and the answers are frequently unsettled under the policy language in place.
- Account for scaling in reserving and in resolution strategy. A design ruling implicates every transaction on the platform or every unit in the field, not a single incident, and that asymmetry should inform the appetite for early resolution.

VII. The Horizon: Directive (EU) 2024/2853

One jurisdiction has stopped debating the question. Directive (EU) 2024/2853, which replaces the European Union's forty-year-old product liability regime, expressly includes software, digital manufacturing files, raw materials, and electricity within the definition of “product.” It covers movables even where integrated into or interconnected with another movable or an immovable, so stand-alone software, embedded digital elements, and AI systems all fall within its scope. It imposes disclosure obligations and establishes rebuttable presumptions of defect and causation that materially ease the claimant's burden. It creates exposure for online platforms. And it carves out free and open-source software supplied outside the course of a commercial activity.

Member States must transpose the Directive by 9 December 2026, and it applies to products placed on the market after that date. For any client selling into the European Union, that is a hard deadline less than three months after this seminar concludes, and the presumptions are the provisions most likely to change outcomes, because they shift work the claimant previously had to perform.

There is also a strategic consequence that cuts against the defense. Once a global manufacturer builds its documentation, safety architecture, and disclosure practices to the European standard, United States plaintiffs will discover that architecture. A client's own EU compliance file becomes a yardstick against which its domestic conduct is measured. The Directive is not merely a foreign regime to be managed by foreign counsel; in time it will be an exhibit.

VIII. Conclusion

The definition of “product” in the United States is evolving and remains inconsistent. Tangibility continues to occupy a prominent place in judicial rhetoric, but it is not what decides cases; distribution, control, and characterization are. For companies whose product is an application, the threshold question remains genuinely contested and jurisdiction-dependent, and no single answer can responsibly be given to a national client. For manufacturers whose tangible goods contain software, the threshold question is already lost, and the contest has moved to which alleged defects are cognizable — a fight that must be conducted defect by defect.

Two developments deserve particular attention in the year ahead. The first is the erosion of the feasibility defense as courts and juries absorb the idea that a design change can be delivered by download. The second is the post-sale duty question raised by connected products, which no court has yet addressed squarely and which is likely to be the next significant expansion.

In the meantime, the most valuable counsel available to a client is unglamorous: study the applicable state's product liability statute and case law before advising, because the answer genuinely differs by jurisdiction; treat software engineering records as the design documentation they will become; and read the European Directive if the client ships to Europe.