

UNPUBLISHEDUNITED STATES COURT OF APPEALS
FOR THE FOURTH CIRCUIT

No. 25-1158

VIR2US, INC.,

Plaintiff – Appellee,

v.

SOPHOS INC.; INVINCEA, INC.,

Defendants – Appellants,

and

SOPHOS LIMITED; SOPHOS GROUP PLC,

Defendants.

Appeal from the United States District Court for the Eastern District of Virginia, at Norfolk. Jamar Kentrell Walker, District Judge. (2:19-cv-00018-JKW-RJK)

Argued: May 7, 2026

Decided: June 23, 2026

Before KING, AGEE, and HEYTENS, Circuit Judges.

Affirmed by unpublished per curiam opinion.

ARGUED: Jeffrey A. Lamken, MOLOLAMKEN, LLP, Washington, D.C., for Appellant. Kevin P. Martin, GOODWIN PROCTER LLP, Boston, Massachusetts, for Appellees. **ON BRIEF:** Lauren F. Dayton, New York, New York, Lucas M. Walker, Washington, D.C.,

Bonnie K. St. Charles, MOLOLAMKEN LLP, Chicago, Illinois; Brian A. E. Smith, BARTKO PAVIA LLP, San Francisco, California, for Appellant. Wendy Leben, New York, New York, Douglas J. Kline, Boston, Massachusetts, Rohiniyurie Tashima, GOODWIN PROCTER LLP, Washington, D.C., for Appellees.

Unpublished opinions are not binding precedent in this circuit.

PER CURIAM:

This case involves the proper interpretation of the royalty provision of a patent license agreement under Virginia law. In a prior appeal, we held that one part of the salient definition meant that royalties were *not* due on the sales of certain of Defendants' products. After rejecting the district court's contrary conclusion, we remanded for that court to address whether another part of the definition applied to those products. On remand, the district court relied on a plain-language analysis to conclude that the second part of the definition did not apply to the products at issue, leading it to deny summary judgment to Plaintiff and award summary judgment to Defendants.

Plaintiff appeals, arguing that the definition's plain language supports its interpretation, meaning that it should have been awarded summary judgment. In the alternative, Plaintiff argues that the provision is ambiguous and summary judgment should not have been granted at all. Finding no error in the grant of summary judgment to Defendants, we affirm the judgment of the district court.

I.

This case involves competing corporations that design and sell computer-security software.¹ Plaintiff Vir2us, Inc., is a California corporation that, among other things,

¹ When reciting the underlying facts, we draw liberally from our prior decision's account. *Vir2us, Inc. v. Sophos Inc.*, 2023 WL 2136379 (4th Cir. Feb. 21, 2023) (per curiam).

developed and patented antivirus software that uses a process called “containerization” to isolate and then test potentially malicious files in a virtual safe room.

Defendant Invincea, Inc., is a Delaware corporation that sells antivirus software, including a line of products called “X by Invincea.” Some products in this line employ containerization, while others employ a different type of antivirus technology called “machine learning.” At all times relevant to this appeal, X by Invincea products used the same underlying source code regardless of which type of technology the individual product employed. Invincea then activated (or left deactivated) the specific portions of source code for the product a purchaser licensed: containerization, machine learning, or both.

After Vir2us sued Invincea for allegedly selling software that infringed Vir2us’s patented containerization technology, the parties entered into a settlement agreement (“the Agreement”). One part of the July 2016 Agreement granted Invincea a “worldwide license” to use “all patents and patent applications owned by Vir2us.” S.J.A. 717, 718. Another provision bound Invincea to a number of conditions, including that it would (1) deliver quarterly reports to Vir2us containing specific information regarding certain of its sales, and (2) pay Vir2us a royalty for certain other sales. While both of these obligations are relevant here, the second obligation—the royalty provision—is the focal point of the parties’ dispute.

The reporting and royalty provisions require Invincea to report how many of certain products are sold and then pay a royalty only “for each Container Products and Services Sold.” S.J.A. 719. (We will shorthand this label to “Container Products.”) The Agreement defines Container Products as “the accused container products currently called Invincea X

Endpoint – Spearphish Protection and formerly known as Invincea FreeSpace, Invincea Enterprise, and Invincea Advanced Endpoint Protection, as well as natural evolutions and derivations of these products.” S.J.A. 717.² The first part of the definition thus identifies four specific products by name, while the second clause extends the definition to “natural evolutions and derivations” of those listed products. *Id.*

The year after the Agreement was executed, Invincea became a wholly-owned subsidiary of Defendant Sophos, Inc., which is also a computer-security software company. Before acquiring Invincea, Sophos developed and marketed its own software, including Sophos Intercept X, Sophos Intercept X for Server, and Sophos Sandstorm. Sophos acquired Invincea to access its machine-learning technology, and, consistent with that objective, Sophos integrated Invincea’s source code into its own software and activated only its machine-learning technology (i.e., Cynomix files). It’s undisputed that none of Sophos’s products employ containerization technology. Put differently, some of Sophos’s post-acquisition products contain source code that is also found in X by Invincea products, but none of Sophos’s products employ X by Invincea’s containerization technology.

After acquiring Invincea, Sophos undertook Invincea’s reporting and royalty obligations for X by Invincea products, but it did not pay royalties on sales for any of its

² The accused container product Invincea X Endpoint – Spearphish Protection activates containerization-enabling files in the source code and, although it has source code (including the files identified as Cynomix.cpp and Cynomix.h) related to machine-learning processes, those files are deactivated.

own products. And approximately a year later, Sophos announced an immediate end to its sale of all X by Invincea products.

The following year (2019), Vir2us invoked the federal courts' diversity jurisdiction to sue Invincea and Sophos for breach of contract (the Agreement) under Virginia law.³ Vir2us alleged, among other things, that Invincea and Sophos (1) failed to provide accurate quarterly reports (Count I), and (2) owed it more money under the royalty provision (Count II). Relevant to this appeal, Vir2us asserted that post-acquisition versions of some Sophos products (including Sophos Intercept X and Sandstorm, but collectively, "the disputed Sophos products") were actually Container Products subject to the Agreement's royalty provision. Invincea and Sophos denied these allegations, and Invincea filed a counterclaim alleging that Vir2us breached the Agreement by failing to comply with another of its provisions, which is not at issue in this appeal.

An earlier summary judgment order and appeal centered around the first part of the Container Products' definition—the exhaustive list of specific products that fall within its scope—while this appeal concerns the last part of the definition—"as well as natural evolutions and derivations of those products." S.J.A. 717. While different parts of the definition are at issue now than were before, we must interpret this language in light of our prior holding and understanding of the Agreement. Accordingly, we briefly recount the

³ Complete diversity between the parties exists and the dispute involves more than \$75,000. *See* 28 U.S.C. § 1332.

The Agreement contains a choice-of-law clause that provides for application of Virginia law; moreover, throughout the proceedings, the parties have represented that Virginia law applies. The district court used Virginia law, and we do also.

district court's earlier order and the first appeal before diving into the issues currently before us.

On the parties' original cross-motions for summary judgment, the district court—Judge Henry Coke Morgan, Jr., presiding—found that Sophos breached the Agreement by failing to report sales and pay royalties on the disputed Invincea and Sophos products. “In the court's view, because those products contained some or all of the source code found in Spearphish—one of the named ‘accused container products’—each of the disputed products was itself ‘classified as one of the named Container Products.’” *Vir2us, Inc.*, 2023 WL 2136379, at *3. Based on this holding, the district court did not reach the question of whether these products satisfied the “natural evolutions and derivations” part of the definition. *Id.* at *3 n.2. Consistent with its decision, the district court ordered Sophos to file a corrected quarterly report and pay the royalties claimed by Vir2us.

Sophos appealed, and argued that the district court erred in finding that the disputed Invincea and Sophos products satisfied the first part of the definition of Container Products. We agreed and vacated the district court's grant of summary judgment. We explained that this first part of the definition “unambiguously identifies only four ‘accused container products,’ [and so] . . . provides an *exhaustive* list of the ‘accused container products.’” *Id.* at *3. Because the disputed Invincea and Sophos products were “not included in that list,” they could not “be classified as such.” *Id.* As further support for this understanding of the definition's plain language, we noted that two disputed Invincea products (Detect and Prevent) existed at the time the parties signed the Agreement. *Id.* So the parties “could have

included those products in the defined list of accused container products,” but did not do so, and that “omission carrie[d] force.” *Id.*

We also observed that the district court had erred in concluding that the disputed products were the same as the products in the defined list because they share the identical source code. Indeed, we rejected the district court’s conclusion that a delineated product’s source code had any bearing on whether it was listed in the first part of the definition. We observed that the district court had mistakenly “inserted a term [relating to the product’s source code] into the Container Products definition that the parties didn’t include,” thus ignoring the provision’s plain language. *Id.* at *4.

The district court’s prior decision, however, addressed only the first part of a two-part definition. It had not addressed whether the disputed Invincea and Sophos products fell within the second clause of the definition of Container Products. Rather than address that question for the first time on appeal, we followed our usual practice and remanded to the district court to consider that question in the first instance. *Id.*

Because Judge Morgan passed away during the pendency of the appeal, the case was reassigned on remand to Judge Jamar K. Walker. The parties again filed cross motions for summary judgment, and argued that the definition of Container Products unambiguously supported their positions for why the disputed Sophos products were or were not “natural evolutions and derivations” of the accused container products. For its part, Vir2us argued only that they were “derivations” and did not rely on the “evolutions” part of this clause. In its view, the disputed Sophos products qualified as Container Products because they “derived machine learning technology from [one of the accused

container products listed in the first part of the definition,] Invincea X Endpoint – Spearphish Protection.” S.J.A. 304.⁴

The district court determined that the disputed Sophos products were not “natural evolutions and derivations of [the defined] products,” and thus were not products for which royalties were due. S.J.A. 717. The court concluded that to be a derivation of one of the accused container products, a product had to employ containerization technology. It reached this result based on the plain meaning of the word “derivation”; a plain language reading of the Agreement’s definition of Container Products; that definition’s context within the Agreement as a whole; and other features of the Agreement, like the fact that it arose from the patent-infringement litigation only over Vir2us’s containerization technology. From this understanding of what constitutes a “derivation,” the district court’s analysis was relatively straightforward. It observed that the undisputed record showed that the disputed Sophos products used only Invincea’s machine-learning source code and did not employ any containerization technology. Accordingly, as to the disputed Sophos products, the district court granted Sophos’s motion for summary judgment and denied Vir2us’s motion.

⁴ Specifically, Vir2us pointed to the presence of two source code files (Cynomix.cpp and Cynomix.h) in both Spearphish and the disputed Sophos products as the basis for claiming that the latter is a derivation of the former.

As discussed, the record reflects that Cynomix.cpp and Cynomix.h are files solely relevant to the machine-learning method for protecting malicious files, and they are not relevant to containerization. Put differently, these files are enabled in products that employ machine learning. But because X by Invincea’s products all included the same source code regardless of which method of protection was deployed, these files were present in its container products (including Spearphish), even though they were deactivated.

The district court's grant of summary judgment as to the disputed Sophos products did not resolve all claims in this case, though, as it only entered partial summary judgment. In the same order, the district court determined that genuine issues of material fact existed with respect to whether certain disputed Invincea products satisfied the natural evolutions and derivations definition, so it denied summary judgment on that claim. But the parties later resolved those claims and asked the district court to dismiss them with prejudice, which the court did. So, they are no longer at issue in the case or this appeal. The district court also denied summary judgment on Invincea's counterclaim against Vir2us after concluding that genuine issues of material fact existed as to it. This counterclaim remains pending before the district court.

Ordinarily, we could not exercise appellate jurisdiction over Vir2us's challenge to the district court's non-final order granting partial summary judgment as to the disputed Sophos products. *See Fox v. Baltimore City Police Dep't*, 201 F.3d 526, 530–31 (4th Cir. 2000) (observing that, with certain exceptions, “[w]e lack jurisdiction to review a district court's order unless that order constitutes a ‘final’ judgment”). But the district court granted an unopposed motion under Federal Rule of Civil Procedure 54(b) to enter final judgment with respect to the claim involving Sophos's disputed products. *See Fed. R. Civ. P. 54(b)* (permitting a district court to “direct entry of a final judgment as to one or more, but fewer than all, claims or parties only if the court expressly determines that there is no just reason for delay”). Relatedly, it granted a stay of district court proceedings on Invincea's counterclaim during the pendency of this appeal.

Here, neither party contests the Rule 54(b) certification. And the district court's order engaged in the proper two-step analysis for weighing the particulars of this case to determine whether its judgment was "final in the sense that it is an ultimate disposition of an individual claim entered in the course of a multiple claims action" and that "there is no just reason for the delay in the entry of judgment." *MCI Constructors, LLC v. City of Greensboro*, 610 F.3d 849, 855 (4th Cir. 2010) (cleaned up); *see also Braswell Shipyards, Inc. v. Beazer E., Inc.*, 2 F.3d 1331, 1335–36 (4th Cir. 1993) (setting out factors that the district court should consider in determining whether no just reason exists to delay entry of judgment). Having reviewed the district court's order and finding no abuse of discretion in the Rule 54(b) certification of its judgment as final, we have jurisdiction to address the merits of the summary judgment entered against Vir2us as to the disputed Sophos products. *See MCI Constructors, LLC*, 610 F.3d at 856.

II.

In assessing the merits of this appeal, we review de novo both the district court's ruling on the cross-motions for summary judgment as to the disputed Sophos products and its "contract interpretation underlying its summary judgment ruling." *Young v. Equinor USA Onshore Props., Inc.*, 982 F.3d 201, 205–06 (4th Cir. 2020). "When cross-motions for summary judgment are before a court, the court examines each motion separately, employing the familiar standard under Rule 56 of the Federal Rules of Civil Procedure." *Desmond v. PNGI Charles Town Gaming, L.L.C.*, 630 F.3d 351, 354 (4th Cir. 2011). Under that standard, summary judgment is appropriate only "if the movant shows that there is no

genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law.” Fed. R. Civ. P. 56(a).

The same rules govern our review of the Agreement now as governed it last time: Under Virginia law, “when the terms of a contract are clear and unambiguous, a court is required to construe the terms according to their plain meaning.” *Golding v. Floyd*, 539 S.E.2d 735, 736 (Va. 2001). Put differently, “if no patent or latent ambiguities exist, a court should enforce the plain meaning of the contractual language without resort to extrinsic evidence.” *Smith v. Smith*, 597 S.E.2d 250, 254 (Va. Ct. App. 2004). “An ambiguity exists when the contract’s language is of doubtful import, is susceptible of being understood in more than one way or of having more than one meaning, or refers to two or more things at the same time.” *Va. Elec. & Power Co. v. Norfolk S. Ry.*, 683 S.E.2d 517, 526 (Va. 2009). In assessing whether an ambiguity exists, the contract “must be considered as a whole without giving emphasis to isolated terms,” *TM Delmarva Power, L.L.C. v. NCP of Va., LLC*, 557 S.E.2d 199, 200 (Va. 2002) (internal quotation marks omitted), and “the words employed in the contract” must be given “their usual, ordinary, and popular meaning,” *Video Zone, Inc. v. KF & F Props., L.C.*, 594 S.E.2d 921, 924 (Va. 2004).

With those principles in mind, this appeal poses a single question: are the disputed Sophos products “natural evolutions and derivations of” the “accused container products” identified in the Agreement, such that they are subject to its royalty provision?⁵ To answer

⁵ The parties’ disagreement about what constitutes a “derivation” does not itself demonstrate that the Agreement is ambiguous on this point. *Douglas v. Hammett*, 507 S.E.2d 98, 101 (Va. Ct. App. 1998) (“Although parties may advance different

(Continued)

that question, we must first understand the meaning of the contractual term “natural evolutions and derivations.” And, because Vir2us has never asserted that the disputed Sophos products are “evolutions,” our focus is just on the meaning of “derivations.”

Dictionary definitions of “derivation” uniformly suggest a sequential relationship between two things in which the latter originates from—or has been produced from—the former. For example, Webster’s (Third) New International Dictionary defines a “derivation” as “[a] handing on or transmission from a source; . . . the source from which a thing is derived.” *See also Derivation*, Merriam-Webster.com, <https://www.merriam-webster.com/dictionary/derivation> [<https://perma.cc/GL7B-HADV>] (last visited June 22, 2026) (defining “derivation” as “source, origin”; “descent, origination”; and “something that originates from something else : something derived”). The Oxford English Dictionary similarly defines “derivation” as, *inter alia*, “[t]he action of drawing, obtaining, or deducing from a source,” and “[o]rigination or coming forth from a source; extraction, origin, descent.” *Derivation*, Oxford English Dictionary, https://www.oed.com/dictionary/derivation_n1?tab=meaning_and_use#6972738 [<https://perma.cc/DV75-38E4>] (last visited June 22, 2026). And Black’s Law Dictionary defines “derivative” as “[s]omething that has been developed from or been produced from something else,” *Derivative*, Black’s Law Dictionary (12th ed. 2024), and defines “derivation” as “[t]he origin of something, esp. a word, phrase, or idea,” *Derivation*,

interpretations of the provisions in an agreement, this does not necessarily imply the existence of ambiguity where there otherwise is none.” (internal quotation marks omitted)).

Black’s Law Dictionary (12th ed. 2024).⁶ In light of these definitions, we conclude that for a product to be a “derivation,” one of the accused container products must have served as its starting point—that is, its origination—such that the later product was “developed from or produced from” it.

Applying this understanding of “derivations” to the case at bar, we conclude that the disputed Sophos products are not, as a matter of law, “derivations” of the accused container products. Principally, the definitions of “derivations” set out above each denote a relationship with the originating product that is stronger and more direct than what the record demonstrates here. Vir2us nonetheless maintains the requisite connection exists because the accused container products and the disputed Sophos products share some source code, specifically certain machine-learning (Cynomix) files. But that alone is insufficient to demonstrate that the disputed Sophos products are derivations of the accused container products, especially given the evidence in the record that cuts against any connection between the two categories of products.

As a starting point, the accused container products are the delineated products in the first part of the Container Products’ definition: the ones “currently called Invincea X Endpoint – Spearphish Protection and formerly known as Invincea FreeSpace, Invincea Enterprise, and Invincea Advanced Endpoint Protection.” S.J.A. 717. So, only derivations

⁶ Because the concept of “originating” features so prominently in these definitions, we observe that “origin” means “[t]hat from which anything originates, or is derived; source of being or existence; starting point.” *Origin*, Oxford English Dictionary, https://www.oed.com/dictionary/origin_n?tab=meaning_and_use#33354404 [<https://perma.cc/T9TQ-2Y9H>] (last visited June 22, 2026).

of *those* products are also Container Products subject to the royalty provision. And the summary-judgment record shows that versions of the disputed Sophos products existed *before* Invincea's source code was ever integrated into them. Therefore, they are not products solely "developed from" an accused container product, nor are the accused container products their exclusive originating source.

That doesn't end the inquiry, however, because it's possible for a later product to derive from more than one originating source. Here, though, the record does not show any clear lineage between the accused container products and the disputed Sophos products. While the disputed Sophos products eventually integrated some source code from Invincea, the record shows nothing more than that: common source code. Vir2us marshaled no evidence, for instance, that the common source code itself originated with or was otherwise unique to the accused container products.⁷ Nor did Vir2us come forward with *any* evidence that Sophos physically derived (i.e., obtained, extracted, etc.) the source code it integrated into its products from any of the accused container products as opposed to using some other means once it acquired Invincea. And in fact, there's evidence in the record to suggest that Sophos did not pluck the code from any of the accused container products before modifying

⁷ To the contrary, there is evidence in the record showing that even at the time the Agreement was executed, the source code was not used exclusively in the accused container products. Rather, it was also present in other Invincea products (like Detect and Prevent) that existed when the Agreement was executed. Notably, those products employed machine learning technology rather than containerization technology. Yet the parties to the Agreement did not include them in the defined list of products that were Container Products. So, the source code was not the defining feature of whether an Invincea product was included in the first part of the definition of a "Container Product," much less the second part.

it for integration into its disputed products. *See, e.g.*, S.J.A. 678–80 (deposition of Sophos employee involved in the integration process, responding that they did not take the source code from an Invincea product, but instead got it from an Invincea employee’s team). Thus, the record does not demonstrate that Sophos derived *anything* from the accused container products to develop their disputed products. All that Vir2us is left with is the source code’s presence in both the accused container products and the disputed products. But again, that alone does not make the disputed products “derivations” of the accused container products. *Cf. Vir2us, Inc.*, 2023 WL 2136379, at *4 (observing that products’ shared source code should not be conflated with the products themselves).

Quite apart from this gap in the record, a practical understanding of what it would mean to be a “derivation” of an “accused container product” cuts against Vir2us’s argument. The “accused container products” serve as the anchor for what could be a “derivation” of them. And as their name suggests, the accused *container* products only employed *containerization* technology. So while their source code included the Cynomix files for machine learning, those files were deactivated and thus not part of the accused container products’ functionality. They had no practical role—hidden or obvious—in what the accused container products were or did. One would reasonably anticipate a “derivation” of an accused container product to resemble in some fashion its originating source’s primary operations or functions. Yet the disputed Sophos products do not employ containerization technology and Vir2us came forward with nothing else to suggest there was some other factor unique to the accused container products’ source code, which could support a finding that all future products containing that code were necessarily

“derivations.” In short, the mere inclusion of a common piece of source code that is unused in one product and used in another product does not make the latter a derivative of the former’s source.

The context in which the relevant terms appear in the Agreement provides further support for our conclusion. *See TM Delmarva Power, L.L.C.*, 557 S.E.2d at 200 (emphasizing that courts should not analyze contractual terms in isolation, as the broader context in which they appear informs their meaning). That context includes two additional considerations for understanding what constitutes a “derivation”: First, the phrase “evolutions and derivations” appears as part of the fuller definition of “Container Products.” S.J.A. 717. And second, Container Products is a defined term included within the fuller Agreement. S.J.A. 717. We briefly address why each consideration bolsters our reading of the plain meaning of the word “derivation.”⁸

First, the phrase “evolutions and derivations” appears as part of the definition of the contractual term “Container Products.” Everything about this defined term draws attention

⁸ Given how the parties structured their arguments in the district court, that court interpreted the phrase “natural evolutions and derivations” as a whole and applied the word “natural” to modify both evolutions and derivations. *See* J.A. 194–97. While our resolution of this appeal rests solely on our understanding of the word “derivations,” that word’s presence as part of this phrase would only further support our conclusion.

First, Virginia interprets words found together harmoniously, *see TM Delmarva Power, L.L.C.*, 557 S.E.2d at 200, so it makes sense that “evolutions” refers to smaller changes from an original to a later product, while derivations encompasses larger changes that still reflect a closeness to and process of building from an original source in a way that demands more than what is alleged here. *See, e.g., Evolution*, Oxford English Dictionary, https://www.oed.com/dictionary/evolution_n?tab=meaning_and_use#5159051 [<https://perma.cc/3W3J-GNZ8>] (last visited June 22, 2026) (defining “evolution” as, *inter* (Continued)

to products employing containerization technology, meaning that “derivations” falling within the scope of this definition were likely intended to similarly employ containerization technology. After all, the Agreement is defining the term *Container Products*. The first part of the definition for this term identifies four current and past specific products (all of which used containerization technology) and refers to those delineated products as “the accused *container products*.” S.J.A. 717 (emphasis added). The second part of the definition looks to the future—evolutions and derivations—of those accused *container products*. Given that the defined term at issue is the sum of its parts, and absent any contrary indications, it makes sense to limit “derivations” of an accused container product to products that similarly use containerization technology.

Second, the term “Container Products” identifies which products are subject to the Agreement’s royalty provision, and we cannot ignore the broader context and language

alia, “[t]he process of development,” and “[a] process of gradual change occurring in a system”).

Second, the series-qualifier canon—which Virginia courts apply—would support “natural” modifying “derivations” within this clause. Antonin Scalia & Bryan A. Garner, *Reading Law: The Interpretation of Legal Texts* 147 (2012) (“When there is a straightforward, parallel construction that involves all nouns or verbs in a series, a prepositive . . . modifier normally applies to the entire series. . . . In the absence of some other indication, the modifier reaches the entire enumeration.”); *accord, e.g., McMillion v. Commonwealth*, 903 S.E.2d 578, 582 (Va. Ct. App. 2024). The modifier “natural” would further restrict “derivations” to those that are “normal[ly]” or “expected” to flow from the accused container products. *Natural*, Oxford English Dictionary, https://www.oed.com/dictionary/natural_adj?tab=meaning_and_use#35405502 [<https://perma.cc/4NR3-X2NA>] (last visited June 22, 2026).

Thus, interpreting the clause within which “derivations” appears as a whole supports our conclusion that the disputed Sophos products do not fall within the second part of the definition of a Container Product.

within this Agreement to understand what its provisions mean. The Agreement was executed in tandem with the parties’ settlement of Vir2us’s patent infringement claims against Invincea. The patents at issue? Only Vir2us’s containerization technology. Quite naturally, the Agreement references that litigation and the patents and products at issue in it, connecting it to the broader settlement of those claims. S.J.A. 716–28; *see also* J.A. 196; *Vir2us, Inc.*, 2023 WL 2136379, at *3 n.4 (discussing the significance of “accused” in patent-infringement litigations to understand the Agreement’s reference to “accused container products” in the definition of Container Products). Although the Agreement goes beyond the scope of that patent litigation, its provisions must be read with an eye toward what the parties were accomplishing, especially given that its royalty provisions are narrower than the license provisions and limited to “Container Products.” That winnowing of what the parties intended to include in the Agreement carries force and bolsters our understanding of what a derivation of the accused container products encompasses.

Having considered the Agreement’s language and the record before us, we discern no error in the district court’s determination that applying the plain meaning of “derivations” to the disputed Sophos products entitles Invincea and Sophos—not Vir2us—to summary judgment as to those products.

III.

For the reasons set forth above, we affirm the judgment of the district court.

AFFIRMED