

Patent Litigation: An Introduction to Patent Claims, “Limitations,” Infringement, and Invalidity -- Part Three

[Andrew Schulman](#)

Senior Software Litigation Consultant, [DisputeSoft](#)

[SoftwareLitigationConsulting.com](#)

In [Part 1](#) of this six-part series, we discussed how the actionable part of a patent is its “claims,” and [Part 2](#) made the following points:

- Patent claims serve as devices for testing patent infringement and invalidity.
- Patent claims are made up of limitations, which are selected elements or steps implementing an invention.
- The set of limitations comprising a patent claim is not complete, but instead is the subset necessary to differentiate the claim from prior art, while still trying to leave a wide scope of infringement.
- The size of a patent claim is an indication of claim scope (*i.e.*, whether a claim protects a large or small area of technology); adding limitations generally reduces claim scope, but generally increases likelihood of validity.
- “Claim construction” uses the patent specification (and a hierarchy of other sources) to interpret the meaning of terminology in a claim.
- The boundaries of patent property are determined in part by steering between infringement and invalidity.

In this Part 3, we’ll discuss the following subjects:

- Using a patent claim as a parts list or checklist to investigate infringement, by searching for products that include the claim limitations.
- Combining the disaggregated checklist approach to claims with a look at the claim as a whole, and at interconnections among the limitations.

- Defending against a patent owner’s assertion of infringement, by showing the absence of one or more claim limitations, and/or by showing that the patent claim is invalid (*e.g.*, by finding each and every limitation in a piece of prior art).
- A limitation-by-limitation walk-through of a software patent claim, including important nuances such as negative limitations, interconnections between limitations, and small wording differences.
- Nomenclature, including differences between terminology used in a patent claim, and the names used for matching elements found in accused products or in the prior art.
- Changes made to patent claims during examination by the patent office, and the impact of such changes on the use and understanding of patent claims.
- Why the drawing on the front page of a patent isn’t necessarily a good representation of a patent claim.
- Why the seeming “core” of a patent claim, what is highlighted in the title of a patent, or what otherwise seems like what the patent is “all about,” may not be the focus of patent litigation.

Treating a patent claim as a parts list...

[Part 2](#) stated that a patent claim is used to test for patent infringement: one compares a specific patent claim (not the patent as a whole) to an accused product. Part 2 also showed that claims contain “limitations,” which are selected components of an invention.

Putting these two points together, we can see how the limitations of the claim fit into the determination of infringement: one breaks apart a patent claim into its constituent limitations, and looks, one by one, for whether each and every limitation can be found in the accused product. The same thing applies when comparing a patent claim against a given piece of prior art, as part of testing claim validity.

The patent owner must find each and every limitation in a product to correctly accuse it of patent infringement. If a patent claims limitations W, X, Y, and Z, the patent owner (P, or plaintiff) must show that the product made, sold, or used by the defendant (D) contains, embodies, or carries out each one: W + X + Y + Z.

A limitation in a patent, and a corresponding element in a product, might be exact matches, even if they have different names (see “Nomenclature” below). A non-exact match may also be possible, under the Doctrine of Equivalence, but even here, equivalence must be tested on a limitation-by-limitation basis; it is not a matter of seeing whether the accused product as a whole is somehow the same as what the patent claims.

A patent claim must function as a reliable device for testing infringement; this is the “definiteness” requirement of [35 USC 112\(b\)](#), discussed in Part 2. In effect, one uses the claim as a template, looking *through* the claim at an accused product, to see if everything in the claim is present in the product; see Figure 1 below. The product generally may contain additional things, but it must have all the claimed things. This is somewhat similar to determining whether part of a product meets the requirements of an industry standard, or the product’s specifications/requirements.

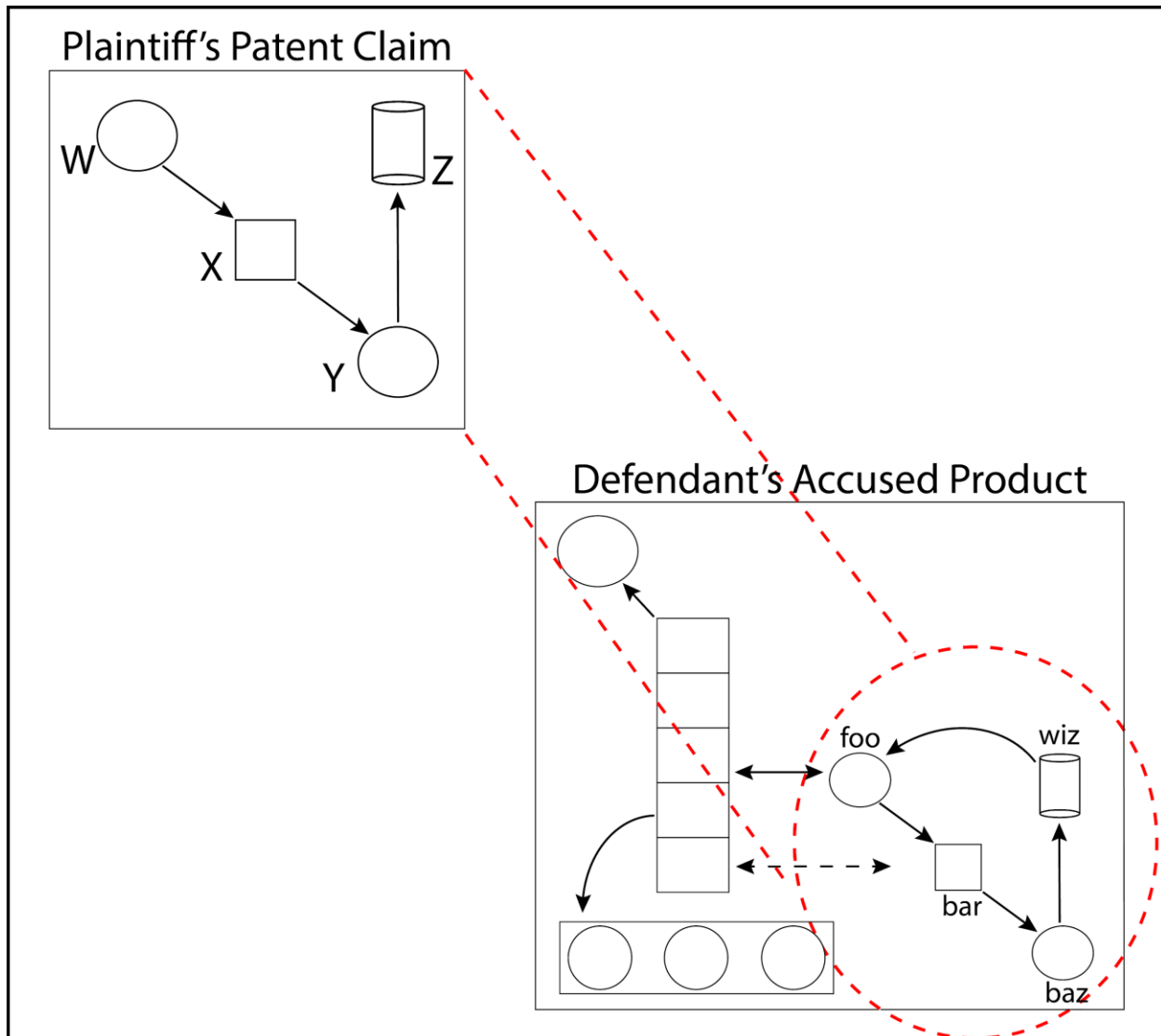


Figure 1: Using a patent claim as a template to see if everything in the claim can be found in an accused product, even if the accused product also contains additional things. All the things (limitations in the patent claim, and matching elements or steps in an accused product or method) must not only be present (in literally identical or equivalent form), but must interrelate in the same way. The patent claim and the accused product may employ different nomenclature (here, $W+X+Y+Z$ in the claim matches $foo+bar+baz+wiz$ in the accused product).

Treatises on patent litigation say little about this process, which is often assumed to be straightforward and uninteresting. As one plaintiff's attorney put it, "You just look in the defendant's stuff and find the things we're claiming" (which is somewhat like one wit's advice meant to quell test-taking anxiety: "Just get the right answers, and you'll do fine"). My forthcoming book on patent claims in litigation has an [in-depth discussion of how to structure comparisons](#) between patent claims and accused products. Patent scholars Dan L. Burk and Mark

A. Lemley in their article on “[Patent Quantum Mechanics](#)” discuss the related issue of how one determines the appropriate bite-size for limitation-by-limitation comparisons.

During this part of the process of testing for infringement, one almost tries to forget the patent as a whole, forget even the claim as a whole, and forget the accused product as a whole, and instead just limit one’s focus, with blinders on, to each separate limitation, and to what might correspond to it in an accused product. This is a disaggregated approach to comparing a patent claim to an accused product (or to a given piece of prior art). The patent claim is treated as if it were simply a parts list, bill of materials ([BOM](#)), shopping list, or perhaps -- and we have to be careful with this analogy, for reasons discussed in the next section -- as one of those checklists you see used on scavenger hunts.

... But not just a parts list

Above, I was trying to steer readers away from an impressionistic view of a patent claim as a whole, towards a limitation-by-limitation analysis that treats a patent claim as a disaggregated parts list. However, several notes of caution must be made here.

First, using a patent claim as a “scavenger hunt” checklist, while essential for researching infringement, is inappropriate for a key part of researching patent invalidity. We’ll discuss this in more detail when we get to “obviousness” in Part 5. Briefly here, it is reasonable to compare a given single piece of prior art with a later patent claim in a disaggregated limitation-by-limitation fashion. An earlier patent, publication, or product “anticipates” a patent claim if that earlier single piece of prior art adequately discloses each and every limitation of the later claim. And multiple prior-art references can be combined to show that a patent claim, while novel, was nonetheless obvious, and therefore invalid. But the combining requires care (coming in Part 5 on “motivation to combine”), and using a patent claim as a checklist to go on a “scavenger hunt” through all the prior art, picking and choosing different limitations from different prior-art references, is an impermissible use of hindsight. This is why I was careful in the previous section to refer to comparing a claim to a “given piece of” prior art, rather than to “the” prior art as a whole.

Second, when testing whether a defendant infringes a patent claim, we need to ensure that all the claim limitations are present in a single infringing “instrumentality” (product or service). P can’t cobble together some Frankenstein’s infringement monster from multiple products that aren’t sold or used together -- nor even from unrelated features within a single complex product. If D’s product is sufficiently large or complex (think an iPhone or Microsoft Windows), it’s possible that one could find many claim limitations represented which, however, do not interoperate with each other, from multiple product modes or configurations that cannot be present or enabled at the same time.

Third, even when P has properly found all its claim limitations present in an instrumentality made, sold, or used by D, infringement also depends on the limitations fitting together (inter-connecting) in the instrumentality in the same way as in the claim (though, as noted in [Part 2](#), possibly serving a different purpose than that set forth in the claim’s preamble). We will see below, and in the next part, how P’s interpretation of one limitation -- and what P chooses to match it to in D’s product -- *constrains* how P can interpret other limitations.

What’s in a name: Infringement & nomenclature

Back in Figure 1, we saw that limitations W+X+Y+Z in a patent claim might “read on” elements or features foo+baz+biz+wiz in an accused product.

As one goes scavenging for infringement, the nomenclature used in the patent claim often won’t match the nomenclature used in descriptions of the accused product (or, if the product is textual like computer code, in the product itself).

What the claim calls a widget might in the product literature instead be called a gizmo or a framis. Even with such different names, one might still be looking at “literal” infringement, in contrast to a situation possibly calling for analysis under the Doctrine of Equivalence (DoE; coming in Part 4). To see why this is so, let’s revisit the simple claim for toner in Part 2:

- 1. [1pre] A toner comprising:
 - [1a] a crystalline polyester resin having an unsaturated double bond;
 - [1b] a thiol compound having a bi- or more-functional thiol group; and
 - [1c] a photopolymerization initiator.

When researching D’s possible infringement of this patent claim, P might see from a particular evidence source that D’s product includes three ingredients, two of which the evidence labels as “IRGACURE 819” and “KarencMT.” Turning to the text of claim 1, no mention is made of those. But we happen to know (see the discussion of this patent, [8,951,704](#), in [Part 2](#)) these are particularly good examples (“preferred embodiments”) of two of the claim limitations, the thiol compound at [1b] and the photopolymerization initiator [1c]. Thus, this is an exact match, even though there could be *other exact matches* (IRGACURE 184 for instance), and even though the names don’t match.

In other words, the claim limitations above serve as *variables* (which can properly be filled-in with any of a group of instances), or as *parameters* (which, to borrow from software terminology, can take on a group of “arguments”). A claim limitation will (with the exception of a narrow “picture patent”) almost never correspond to only a single thing in the real world.

The specification for the patent ([8,951,704](#)) from which the claim was extracted explains that one way of obtaining the crystalline polyester resin [1a] uses “an unsaturated aliphatic dicarboxylic acid and an unsaturated aliphatic diol,” and the sources of product information or the prior art we’re using might talk in those terms, rather than in terms of a crystalline polyester resin.

An expert will often be needed to show that, *e.g.*, P’s “framis” literally reads on D’s “gizmo.” When it’s time for P and D to present their infringement and invalidity evidence, their presentation ought to explain *how* or *why* D’s gizmo is the same as P’s framis, or conversely why a so-called “gizmo” in the prior art isn’t the same as the gizmo in P’s claim.

Absence of the word “because” is often a tip-off that such an explanation is missing. Litigants have a way of merely juxtaposing the claim language on the one hand with the product or prior-art information on the other hand, without explicitly saying how they relate, apart from saying something like “P’s framis – See, *e.g.*, D’s gizmo.” Sometimes this is because the relation is so obvious it doesn’t need explaining, but more often the litigant isn’t quite sure, and is hedging its bets. The contrast between the two types of presentation is shown below, for one limitation:

- [1b] a thiol compound having a bi- or more-functional thiol group -- see Showa Denko KarenzMT.

vs.

- [1b] a thiol compound having a bi- or more-functional thiol group -- D's accused product includes Showa Denko KarenzMT, which is a trade name for pentaerythritol tetrakis (3-mercaptopbutylate), which in turn is of the type of thiol compound specified in P's claim 1, because claim 7 (dependent upon claim 1) indicates it is one possible thiol compound, and because the '704 specification states it is a preferred embodiment.

Note how the second version actually makes an assertion or contention, whereas the first version merely says "see," which asserts nothing more than "you go figure out what the connection is."

Defense!

Following soon after what we're calling the infringement scavenger hunt, in which P looks in D's product for something matching each and every limitation of at least one claim of P's patent, D will of course also be motivated to look through its own product, with the opposite goal of showing that P's asserted match-ups are false. D in other words will be on something of a negative scavenger hunt.

D is not obligated to proactively disprove infringement in any absolute sense. P has the burden of proof to show infringement, and D merely needs to rebut P's assertions. Further, since infringement is all-or-nothing (P must show D has *each and every* limitation), it follows that non-infringement is an OR function: D need only show the absence of a *single* limitation in order to show its non-infringement. Of course, D will want to show as many missing limitations as possible, but it need only show one. Because P will have already in effect said "Their foo is our claim's W," D will generally try to show the absence of W in D's product by explaining how its foo is-not a W. It will, for example, assert that P's claim's W must be large and must connect to X in a certain way, where's D's foo is small or doesn't connect to its "bar" (which P has said plays the role of X; see Figure 1), or does so in a different way.

As also noted earlier, D will likely also try to show that P's patent itself (or at least any particular claim P has asserted against D) is invalid. While a patent issued by the United States Patent and Trademark Office (PTO) is presumed valid, this presumption can be rebutted, including by showing that a patent claim is not novel (it has been anticipated by a piece of prior art) or, even if novel, was nonetheless obvious.

Showing this will require that D search the prior art, looking for a single reference (such as an earlier patent or published patent application, a published article or book, or a product) that anticipates P's claim or, failing that, a combination of prior-art references, or common knowledge in the field, that would render P's patent claim obvious, because an ordinary non-inventive practitioner in the field at the time would have had some motivation to combine them (this is basically how patent law defines obviousness).

There is a risk to P in choosing to exercise its patent: by suing D for infringement, P is also motivating D to try to invalidate the patent. Further, a finding of invalidity in one case will likely preclude any further attempts to use the same patent claim in future cases against other Ds; thus, P needs to win the invalidity battle every time.

When P points to D's specific elements or features of D's product as infringing the patent, this creates a *specific* risk to P. Part 2 noted that infringement and invalidity are in some ways mirror images: "that which infringes if after, invalidates if before." P's position on what infringes therefore also becomes P's acknowledgement of what would constitute an invalidating piece of prior art. Put these together, and when P accuses D's product X, D is now motivated to find X in the prior art -- even if it otherwise wouldn't have thought of X as particularly good prior art. By accusing X of infringement, P has put X-like prior art into play. P's desire for a broad infringement reading may burden it with a broad invalidity reading.

In addition to anticipation or obviousness, D may also try to show that P's patent didn't sufficiently disclose the full scope of what it's claiming ("enablement" -- this can be a problem even for patents on great inventions, such as [Morse's telegraph](#)), or that D tripped itself up by selling or publicly using the invention more than one year before it applied for the patent. Why patent law cares about such a time lag is a topic for when we discuss the so-called on-sale and public-use bars, but for now, simply note that patent law tends to reward disclosers of how to make and use inventions, rather than inventors as such.

P will of course want to respond to D's invalidity assertions with its own validity rebuttal, just as D responded to P's infringement assertions with a non-infringement rebuttal.

A simple software patent claim

It is now time to look carefully at a more complex patent claim than our toner example. This one is still relatively short and was also arbitrarily selected, but unlike the earlier toner example, it has explicit interconnections between its limitations, and some other interesting features, including an unusually-named limitation and negative limitations. We will see the claim, like many patent claims, has a surprising number of subtle but important nuances.

It is claim 1 from a software patent, US [7,472,398](#), granted to Hewlett Packard (HP) in 2008 (long after HP's application in 2003). While there is public debate in many countries over whether patent systems should cover computer software, which many feel are too abstract to be a proper subject for patents, it is still hard to imagine a modern patent system without coverage of something as important as software, and even those with a gut-level opposition to software patents should still know what it is that they are upset about. Here is claim 1, with (as is common in patent litigation) square-bracketed letters added for each limitation, including "pre" to designate the (often non-limiting) preamble:

- [1pre] 1. A computer system comprising:
 - [1a] a central processing unit (CPU);
 - [1b] a memory unit coupled to the CPU;
 - [1c] an application stored in the memory unit and executable by the CPU;
 - [1d] a facade server stored in the memory unit and executable by the CPU; and
 - [1e] a program stored in the memory unit and executable by the CPU,
 - [1f] wherein the program creates an interface between the facade server and a web-browser for exchanging data associated with the application,
 - [1g] wherein the facade server hosts the application without utilizing network protocols and without opening network ports.

I have split the claim into limitations along seams represented by the semicolons and "wherein" clauses. When asserting infringement, P will generally desire fewer limitations, and D to rebut

will generally desire more (as this provides more places for rebuttal); these stances are (for reasons given in [Part 2](#)) generally switched for invalidity.

As one reads through this claim, it is helpful to temporarily set aside the CPU and memory (which we'll get to soon), and focus on the interconnections among the other limitations, which are spelled out entirely in the two “wherein” clauses [1f] and [1g], which can be further parsed and abbreviated:

- program (1e) creates interface (i)
- i between facade-server (1d) and web-browser (wb)
- i for exchange data associated with app (1c)
- 1d hosts 1c without utilizing net protocols
- 1d hosts 1c without opening net ports

P will use the set of limitations as a checklist to find infringement in D's product (what we earlier called a scavenger-hunt checklist, and what Part 2 called a device for testing infringement). D will use it to show non-infringement, and also to find prior art that invalidates the patent claim.

Some nuances of the claim will likely not be evident until P actually begins to compare the claim with one or more target products that it thinks might infringe. Some initial infringement analysis is needed to drive claim construction in litigation, that is, the process of understanding what the claim limitations mean. Even if the patent specification defines a “facade server” (see [1d] above), for instance, the boundaries of this term may not become clear until the claim is compared with an actual accused product that P asserts contains such a facade server. That might in turn affect which prior art might invalidate this patent claim. This is one example of how the up-front patent examination at the PTO cannot catch some possible invalidity issues that litigation (with a specific infringement target) can catch.

At the same time, to use the claim as an infringement checklist, P first needs to start with a reasonable understanding of the claim itself. Certainly, both sides will need at least a preliminary understanding of what the facade server in [1d] might be. This initial need to understand the claim, before getting into infringement or invalidity, is an example of performing claim construction before infringement or invalidity analysis, as will be discussed in Part 4. We're now

going to do this, for each and every limitation that comprises this sample claim, and we will not be delving into an infringement analysis of the claim, comparing it to a defendant's product, until Part 4.

Walking through claim limitations

We've said that a patent claim is similar to the "metes and bounds" of property, which in turn represents a walk around the perimeters of the property. As noted in Part 2, the perimeter description is accurate, because a claim should not contain the full set of elements or steps that implement the invention, but only those elements or steps necessary to differentiate the claim from other technology (especially the prior art).

Let's now take that walk around the perimeters of claim 1 of the facade-server patent. Below, we'll jump around a bit amongst the different limitations, but we'll end up having something to say about each of them. It may seem that we're getting into the weeds here, spending too much time discussing the details of one arbitrarily-chosen patent claim, but this will provide a reasonable feel for the initial reading of a patent claim. Had we picked a different patent, we would have entirely different issues, but the same *type of* issues, and the same basic process.

Some of the claim limitations, like a **CPU** [1a] and a **memory unit** [1b] appear to be gimmies. It's a software product, of course the software will be loaded into memory, and will contain instructions executable by the CPU.

However, it's actually not so simple: if patent owner P sues software vendor D for infringement, P is going to run into an issue here, because a software vendor is likely not making or selling the CPU or memory. Recall that infringement requires that the defendant make, sell, or use each and every limitation. The CPU and memory are likely not part of what any D will make or sell. The CPU and memory only come into play when the software is *used*, generally by D's customer rather than by D itself. P can't wish these two limitations away as trivial, since P itself included them, and presumably did so to tie the claim to a physical machine, thereby avoiding abstract subject matter which is unpatentable under 35 USC 101 (restricting patents to "any new and useful process, machine, manufacture, or composition of matter"). The inclusion of CPU and

memory makes this into a claim for a *machine*. There is a separate claim for a *process*; see claim 6 below.

Perhaps D sells some turn-key computer systems (this would match the “computer system” preamble [1pre]), and it is almost certainly using the software in-house for testing and tech support. So, these are computer systems with CPU and memory. But when D sells its software to consumers (and such sales are likely the basis for the reasonable royalty or lost profits P will seek as monetary damages), infringement will not occur at least until the consumer installs the product on a computer with a CPU and memory. At that point, D is perhaps an *indirect* infringer, for having induced or contributed to the consumer’s direct infringement.

To also capture consumer end-use of the product, not just D’s in-house use, P would either have to sue the end-users for direct infringement, and/or sue D for indirect infringement, in which case it would still have to show end-user direct infringement, but also show D’s inducement, which in part requires (in contrast to what was said in Part 1 about patent infringement strict liability) D’s intent and/or knowledge of the patent.

This patent also includes computer-readable media (CRM) claims. Claim 11 of the same patent begins, “A computer readable media storing instructions executable by a computer system, and when executed the instructions implement a method comprising ...,” and this would get around the CPU and memory issue. P need not assert every claim, and usually won’t, and P could choose to assert claim 11 but not claim 1. But let’s stick with looking at claim 1.

As to claim 1, as for all patent claims, we should ask ourselves *when* infringement would occur. This is not a method claim, and so infringement of this claim likely doesn’t occur during the actual execution of the software. Contrast claim 6, for “A computer-implemented method comprising ...,” infringement of which requires actually *using* the method, in contrast to claim 1, for which infringement could also be shown by *making* or *selling* the product. Method claims, and the “steps” that constitute their limitations, were noted in [Part 2](#).

The claim refers to a **program** [1e] “executable by the CPU” -- note the wording: executable, not necessarily executed. Working with patent claims requires careful attention to differences such as -able vs. -ed; the executable vs. executed distinction has been featured in at least one

[case](#). Here, infringement doesn't require execution of the program, merely that it be susceptible to execution (*i.e.*, it contains instructions).

On the other hand, the claim also states that the program is “stored” in memory, not merely storable, so this appears to require that the program be loaded into memory, even if not yet running. On the third hand, in [1f] the program “creates an interface ... for exchanging data.” Clearly, the data need not (yet) actually be exchanged for infringement to occur, but the interface does actually need to be created; perhaps it can be registered during one-time installation of the software. We have to think about install-time vs. load-time vs. run-time. We're focusing on small word differences; while an unnatural exercise for many engineers, this careful attention to words flows from the patent system's use of human language to describe inventions.

The presence of “**a web-browser**” in [1f] does not necessarily require that D make or sell a web browser (though perhaps D bundles a browser with its software), but one must be present (like CPU and memory). It is at least a *precondition*, if not a limitation (precondition vs. limitation is an important distinction we'll get into later in this series). Per [1f], the browser must be connected to the facade server, via the data-exchange interface created by the “program.” Incidentally, if we decided that an infringing product must include a web browser, could a defendant “design around” the patent simply by assuming the consumer already has a web browser, or by providing one for download on its web site, but not as part of the product? A patent claim should not be easily evaded with mere contrivances; this is part of the role of the doctrine of equivalence we've been mentioning.

The reference to **data “associated with”** an application in [1f] sounds rather vague. Patent claims are surprisingly replete with such seemingly-loose connections, including “based on”, “related to”, “connected to”, and the like. However open-ended such connections may be, there still must be *some* connection, and this will constrain how the claim is mapped to a product. Data “associated with” an app at least requires something about the data that would not be true, were the app not present (this is a but-for test). This also shows how proper reading of claim limitations becomes more constrained as one proceeds through the claim: here, having data associated with an app and exchanged with a web browser implies that whatever constitutes the app [1c] in an infringing product cannot be just any app, but must at least have associated data that will be exchanged with a web browser, per [1f]; limitation [1f] *constrains* [1c].

More about the **application** [1c], and its associated data [1f]: an infringing product is likely going to enable the user to put some sort of web-browser interface on an existing application. The application is very likely going to be something that the users themselves supply: perhaps a legacy app for which they want a spiffy front-end, without rewriting the app. An infringing product likely comes with some sample applications, but this is probably not where the revenue generation for damages is going to be (unless P can show that customers wouldn't purchase the product without the sample apps). Using a claim to identify infringement should take into account how that infringement will be counted and monetized; one choice of an infringing element might be more valuable, or at least more countable, than another otherwise-good choice.

D's customers will likely be combining their own app with D's product, and then using it in-house, or perhaps re-selling it downstream to further end-users. Much of the "associated" data is likely generated by those end-users. Such considerations will further impact P's choice of the best targets for an infringement suit, and what types of infringement these targets are allegedly committing (directly making, using, selling; or indirectly inducing or contributing).

The "application" [1c] and "program" [1e] are two distinct entities in this claim. While we must separately map each claim limitation to an attribute of the product, there is no patent-law rule requiring that each separate claim limitation have a one-to-one correspondence with a *separate* divisible piece of an infringing product. One of P's limitations might be found in what D regards as two separate pieces. This still counts toward infringement, so long as D's pieces are not merely unrelated grab-bag items. Conversely, two of P's limitations might be embodied in, or carried out by, what D regards as a single piece. We might find both the app and the program residing in a single executable file in D's product, for instance, or in a single .java file in its source code.

So far, we've just touched on the relatively simple-looking limitations: the CPU, memory, application, and program, and already things are not so simple. Much of the complexity lies not in difficult technical terminology (of which there is little here, in contrast to the toner patent claim), but rather in the use of fairly ordinary words (apart from "facade," which we'll get to in a moment) to describe what was at the same time supposed to be new and non-obvious technology.

The claim requires in [1g] that the facade server host an application "**without utilizing network protocols** and **without opening network ports**." We earlier noted that D in patent litigation will

be looking for absences, *i.e.*, for P’s claim limitations that are *not* present in D’s product. But here, the roles are reversed: P will be looking for this required absence, and D would like, if it must admit to having a relevant server, that at least this server hosts apps *using* network protocols and/or with *open* network ports. D may for example argue that even a localhost or [local loopback](#) still uses a network protocol. However, these limitations require a *specific* absence. For example, it won’t help D that the web browser makes HTTP requests using port 80, because “without opening network ports” need only apply to the facade server.

Patent claims often have such required absences, under various guises. Consider Amazon’s famous [one-click patent](#) claim, which in part required “in response to *only a single action* being performed, sending a request...”. To show infringement, Amazon would in part need to have shown the absence of a second action before a request was sent. Similarly, “whereby the item is ordered *without using* the shopping cart model,” at least if considered a limitation (there is substantial case law around the words whereby vs. wherein), likely means both a shopping-cart model must be present, *and that it not be used* for the one-click operation.

“Facade server”: interpreting a term in a patent claim

Finally, we come to the elephant in the room, or at least what seems like it: the “**facade server**” limitation. A software engineer reading the claim would reasonably think they know what all the other limitations mean. At least think they know, because a patent owner is allowed to be its “[own lexicographer](#),” and use ordinary-seeming technical terms in idiosyncratic ways, so long as its idiosyncratic definitions are disclosed in the patent specification. While claim-construction rules are complicated, the default position is “plain and ordinary meaning.” Software engineering has many terms which use ordinary words in specific ways with a plain and ordinary meaning for an ordinary software engineer: stack, shim, thunk, wrapper, and so on.

But what is the plain and ordinary meaning of facade or facade server? The word *facade* refers to the external front of a building, and also suggests an external appearance that hides or masks a less-appealing inside. The term “[facade pattern](#)” appears in software design. But facade server?

This is not a standardized term of art, and yet the presence of a facade server is required to find infringement (or invalidity) of P’s claim. It is unlikely that a component of any infringing or

invalidating software is going to come pre-labelled as a “facade server”; see our earlier discussion of nomenclature. At the same time, an instance of a facade server, whatever someone else calls it, is likely going to be a fairly specific thing; it must be more specific than a plain server. Adjectives constrain claim scope, and limit it. Any product that infringes this claim must contain not any server; it must specifically be a facade server.

Looking at the patent specification, how does it describe the facade server? It does so by referring to numbered item **110** in a drawing which appears as Figure 1 of the patent, reprinted here as Figure 2:

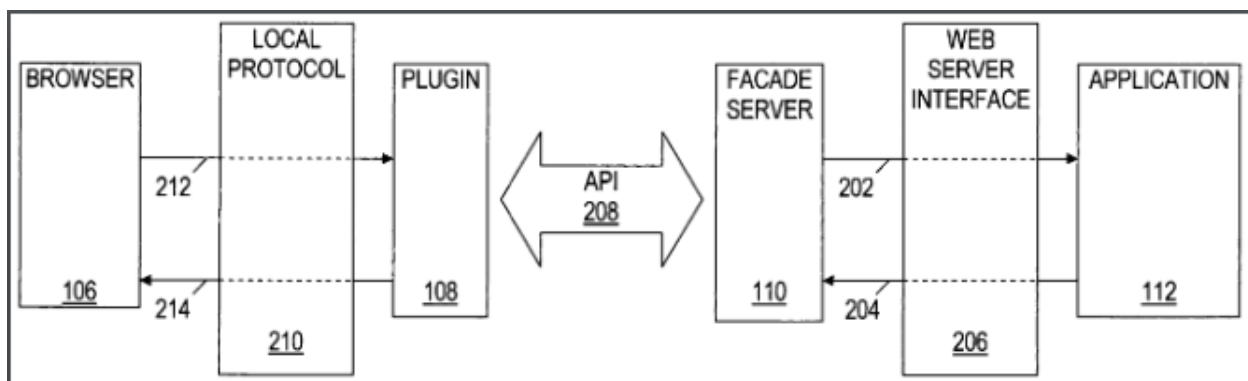


Figure 2: A drawing from the facade server patent. The local app at 112 talks indirectly to the browser at 106. While the drawing at 108 includes a “plugin,” no claim in the patent reflects this.

Before getting to what this specific patent specification says about the facade server in item 110 in the above drawing, a few words about patent drawings generally. Note in Figure 2 that this drawing includes a “plugin,” but that no claim in this patent includes such a plugin. Patent drawings are not perfect illustrations of the patent claims. A patent’s claims and its drawings often diverge, because while claims are amended during examination at the PTO -- the government’s grant of a patent is not “take it or leave it” -- the rest of the patent application often won’t be amended, and often can’t be, without introducing so-called “new matter” that would interfere with a desired earlier priority date for the patent. Even though patents are often portrayed using their drawings, the drawings (like the patent title, abstract, and description) are secondary to the claims. And while the claims must have sufficient support in the specification, the specification (including the drawings) will often contain additional material that isn’t in the claims. Such disclosed but unclaimed material, as the plugin here, is viewed as “dedicated to the

public,” meaning that the patentee can’t point to it as an infringing equivalent to what it has claimed.

But while a patent drawing is not in the patent itself, nor synonymous with its claims, it can still be helpful to view the drawings while reading the claims. Here, the facade server [1d] needs to be defined by reference to the patent specification, which in turn refers to numbered parts of a drawing:

“The facade server **110** may interact with the application **112** by creating one or more interfaces ... such as application programming interface[s] (APIs) and common gateway interfaces (CGIs), that may ordinarily be created by a web-server. The interfaces allow the application **112** to communicate with the facade server **110** in a substantially similar manner as the application **112** would communicate with a web-server.”

In other words, the facade server is like a web server, but isn’t one. As reflected in the patent’s “file wrapper” (see Claim Construction in Part 2), HP distinguished the facade server limitation from something close in the prior art by stating that the facade server “does not use any network protocols.” A patent applicant often must distinguish its claim from something *close* in the prior art, by making fine distinctions.

Here, the facade server is a web server that does not use any network protocols. This is not explicitly stated in [1d], but is implicit from the patent specification. Additionally, the required absence is broader than that of [1g], which only applies to how the app is hosted.

As explained earlier, the file wrapper or “prosecution history” contains the correspondence between the patent applicant and the PTO examiner. For purposes of claim construction, it is regarded as “intrinsic evidence” to the patent. In other words, it is basically part of the patent. While it is a public set of documents, however, it is not as easily-accessed as the basic patent document. Until 2015, Google [hosted file wrappers](#); they are less easily accessed at the PTO’s “[Public PAIR](#)” web site. Armed with our patent’s application number (10715250), which is shown in the patent, we can [download a large zip file](#) containing images of the file wrapper. Google does not OCR these image files, much less index them. A company called [PatDek](#) is working on making the full text of file histories searchable.

Looking in the file wrapper, we see that HP provided the following explanation to the PTO examiner: “The facade server 110 derives its name from the fact that, despite being in software form, it is able to perform at least some of the functions of a conventional, hardware server.” That’s a fairly baffling explanation, until we read on that “The disadvantages of hardware servers (*e.g.*, security/vulnerability issues stemming from the use of network protocols) are mitigated by the facade server, because the facade server is a software program and, as such, does not use any network protocols.” The meaning of the raw “facade server” terminology in the claim is amplified by the file wrapper. A patent applicant’s explanations to the PTO during examination are *binding* on the patent owner after the patent is granted. HP could not explain the facade server in one way in order to get the patent granted, and then in an inconsistent way during litigation (see the “nose of wax” in [Part 2](#)).

Thus, a facade server, as this patent uses the term (and that’s all we care about when using the claim to identify infringement or invalidity), is a web server with only non-network, *i.e.*, local, access. The facade server would host apps which display in a web browser, just like web sites and web apps do, but the apps and the browser would reside on the same computer. This understanding is reinforced by dependent claim 4, which reads: “4. The system of claim 1 wherein the application, the facade server, and a web-server interface by which the application exchanges data with the facade server all utilize a common address space.” (Though this also means that claim 1 *could* involve disjoint address spaces, perhaps with the elements communicating via non-network remote procedure calls or RPC.)

Generally, then, the claim relates to using a non-networked web server to put a web-browser front-end on an application (which is likely an already-existing legacy app that can’t or won’t be modified). Instead of having the browser make network requests of the web server, all these pieces reside on a single computer, communicating without network protocols or ports. The claim is basically for a local version of the web -- not local in the sense of an intranet, but rather the web browser (using a non-networked web server) provides a front-end to a local application, acting as its display engine.

At a high level, this sounds a lot like what was once called “local CGI.” We’ve just seen that the patent refers to the Common Gateway Interface (CGI). CGI is an old mechanism for web servers to run an app that doesn’t know about the web (an old command-line program for instance), and

generate HTML from the app’s output. The resulting HTML is displayed in a web browser. The idea of local CGI was that a web browser could reside on the same computer as the web server, and be used as the “display engine” for old (including legacy) apps, without rewriting those apps to generate HTML, nor to do I/O via HTTP. Here, the facade server will take care of that. Of course, this is simply a high-level summary, and we must stick with the claim language, albeit the claim language as understood in the context of the entire specification and the file wrapper.

Having gone through all the limitations of claim 1, we can do a quick back-of-the-envelope drawing of how the limitations (apart from CPU and memory) interrelate; see Figure 3 below. Unlike the patent drawing in Figure 2, this one reflects the final contents of claim 1, granted by the PTO.

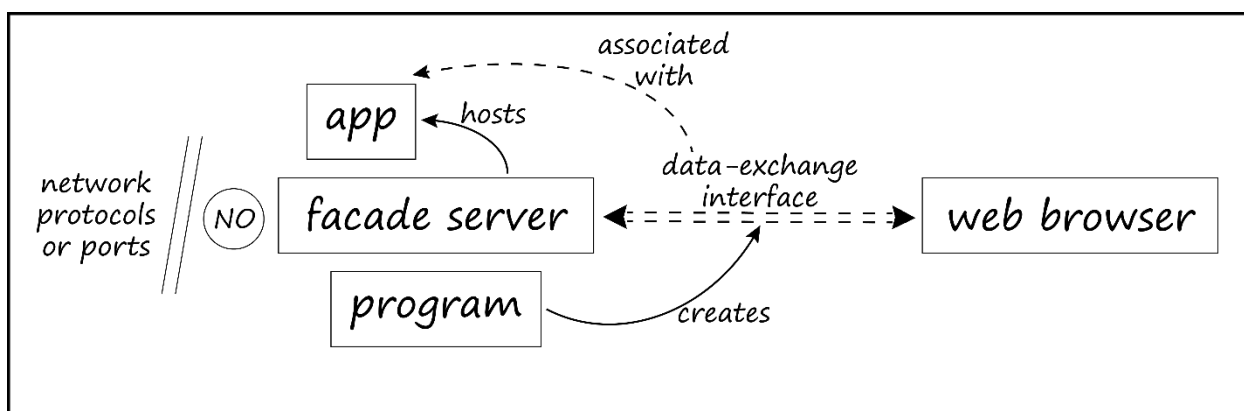


Figure 3: A back-of-the-envelope drawing of claim 1 of the facade server patent, without the CPU and memory limitations.

We’re done with our initial walk-through of claim 1. It’s almost time to try to use claim 1 to find infringement (Part 4) and invalidity (Part 5).

The “core” of the claim may not look like much

One last point, however: I referred to the facade server as “the elephant in the room,” hoping to convey that this was in some regards the hardest to pin down, or most debatable, limitation in the claim. It also seems to be what the patent is “all about”: the term facade server also appears in the title of the patent (“Method and system for hosting an application with a facade server”).

However, it’s often a mistake to view a particular limitation as more crucial than any other. It is certainly a mistake, called “[patent profanity](#),” to use such words as “critical,” “crucial,” or

“particularly important” in the patent itself. Remember, *all* limitations must be found to declare infringement or invalidity (“I love all my children equally”). Often the parties in litigation will end up fighting, not over what seems the heart or core of the invention, but rather over something seemingly more banal. Litigants in a facade-patent case might spend more time fighting over the program [1e], for example, than over the facade server [1d]. Similarly, it might turn out that something other than the facade server was the one thing that P needed to add to its patent application to distinguish it from the prior art, and have the patent application approved. That one thing may appear quite narrow (though it still must have been non-obvious).

Indeed, a comparison of claim 1 in the patent, with that in the [original application](#), shows that P added two limitations during the PTO’s examination of the application. These two additions show up in the file wrapper for this patent (if you’re following along from the earlier file-wrapper discussion, open up the Google-hosted file-wrapper zip file and extract “10715250-2008-11-04-00005-NOA.pdf”); see the underlined portions of Figure 4 below:

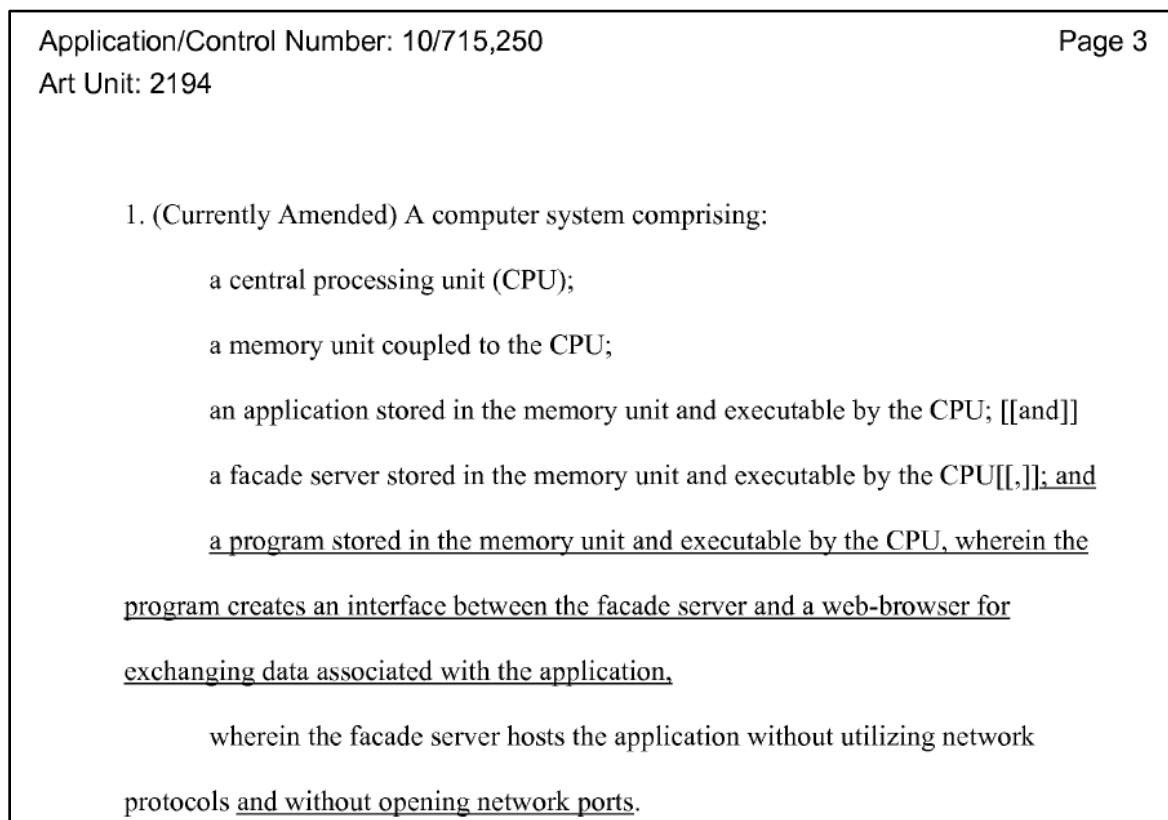


Figure 4: Patent owner’s two amendments to its original application for claim 1.

In Figure 4, we see that before its patent was granted, HP added the “program” (which creates a data-exchange interface between facade server and web browser) and “without opening network

ports” limitations. Given the context available in the file wrapper, this was done to avoid prior art. In the original application, the program appeared in separate dependent claim 2. HP narrowed claim 1 by making the program a limitation, *i.e.*, an absolute requirement in claim 1, rather than an optional dependent in claim 2.

The facade server itself, seemingly the star in this claim, is something the patent examiner found already in the prior art, in the form *e.g.*, of an “interactor” in [6,717,593](#) that can download XML and JavaScript via inter-process communications (IPC) rather than via HTTP.

Why did finding a facade server (though named differently) in the prior art not result in immediate refusal of the patent application as not novel? Because the claim was not just for the facade server, but for this plus the other limitations we’ve been discussing. All those would need to be found in the prior art to anticipate the patent claim (in a single piece of prior art for anticipation, or in multiple references for obviousness).

The patent examiner believed he had found all the limitations of the [original applied-for claim](#). So, to create sufficient distance between its claim and the prior art, HP added the program and added the absence of open network ports. The patent was then granted. Adding these limitations was not decorative. HP did this to get the patent claim granted. Their addition makes the claim a more finely-filtered trap for infringement; *i.e.*, it catches *less* infringement than it would without them, but it is more likely to be a *valid* trap.

From this perspective, we could say that the program and “without opening network ports” are the stars of this patent claim. That possibility is certainly not apparent from looking only at the granted patent, without benefit of the original application or file wrapper. This is one reason why the file wrapper, seemingly not part of the patent itself, is regarded as intrinsic evidence for claim construction. Keeping the file wrapper separate from the patent, and difficult to access, certainly doesn’t help make the claim “patent,” in the sense of something that is open and overt, and that provides full notice to potential infringers (the notice is a bit like that provided by the [local planning office](#) in Douglas Adams’s *Hitchhiker’s Guide*). Patent litigation treatises often discuss claim amendments (as reflected in the file wrapper, and in the difference between a granted patent and the original application) in the narrower context of something called “prosecution history estoppel,” which is a carve-out to the doctrine of equivalence. But claim amendments

have wider importance than that, and one should compare the granted patent claims with those in the original application (which is generally published 18 months after the application date).

At some point in the litigation, P and D will reach agreement on what are the “most significant” claim terms requiring the court’s claim construction at a “Markman” hearing. Which does not at all mean that the parties will agree on the meaning of those terms. Rather, they will agree on which terms have meanings that are most in dispute. But even this agreement on what really matters for the litigation will have followed an intense period in which P and D wrestle with P’s claim and with how it maps onto D’s product and onto prior art found by D.

To wrap up this third part of the six-part series, we’ve looked at:

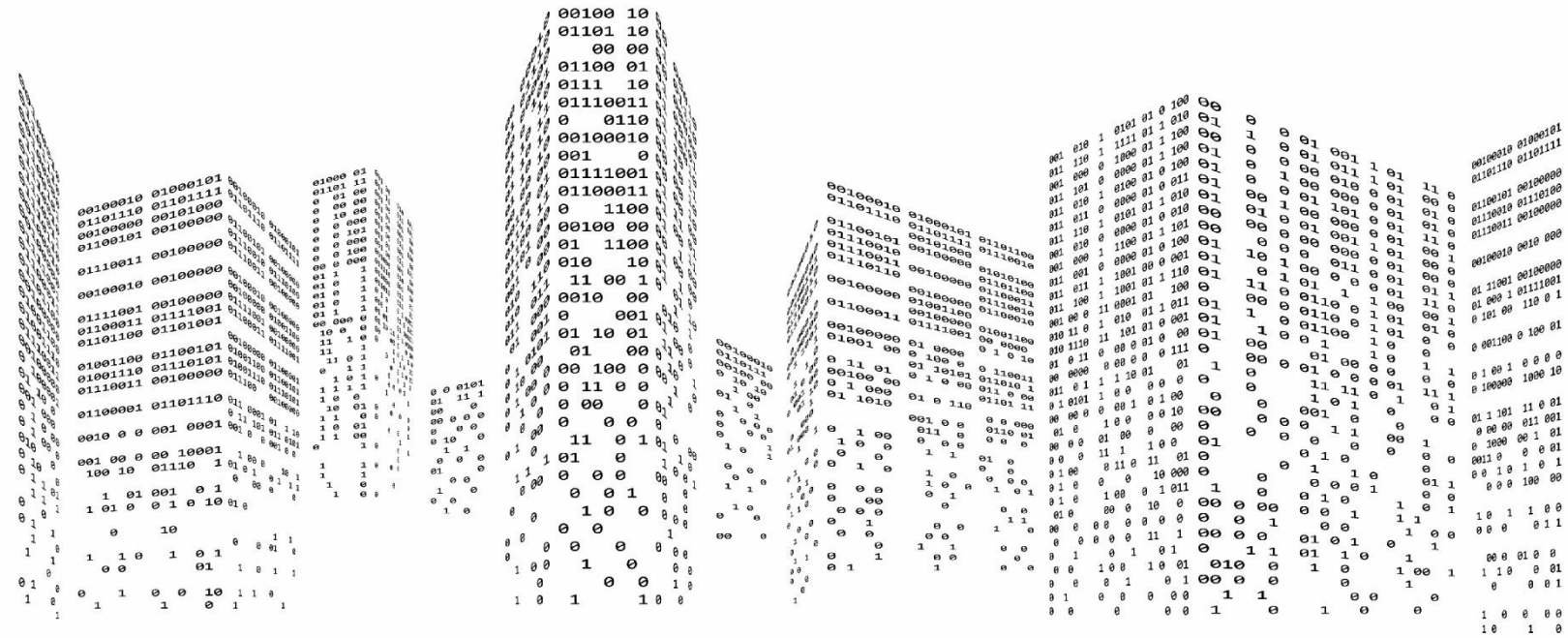
- Using a patent claim as a parts list or checklist to investigate infringement, by searching for products that include the claim limitations.
- Combining the disaggregated checklist approach to claims with a look at the claim as a whole, and at interconnections among the limitations.
- Defending against a patent owner’s assertion of infringement, by showing the absence of one or more claim limitations, and/or by showing that the patent claim is invalid (*e.g.*, by finding each and every limitation in a piece of prior art).
- P’s accusation of infringement against product X puts X-like prior art into play, when D tries to show P’s patent is invalid; this is one example of how the up-front PTO examination can’t catch some invalidity issues that litigation can catch.
- A simple but exemplary software patent claim, looking at nuances such as:
 - negative limitations;
 - preconditions vs. limitations;
 - inter-connections between limitations; and
 - small wording differences (such as the difference between “executed” and “executable”).
- Nomenclature: the names the patent owner uses for claim limitations, including the ability of a patent owner to be “its own lexicographer.”
- Differences between nomenclature used in a patent claim, and the names used for matching elements found in accused products or in the prior art.

- Adjectives in a patent claim generally limit the claim’s scope, *i.e.* how much infringement it can catch.
- Changes made to patent claims during examination by the patent office (*i.e.*, the difference between claims in the granted patent and in its original application), and the importance of such changes for the use and understanding of patent claims.
- The file wrapper for a patent is important intrinsic evidence for what a patent claim means.
- Why the drawing on the front page of a patent isn’t necessarily a good representation of a patent claim.
- Why the seeming core of a patent claim, what is highlighted in the title of a patent, or what otherwise seems like what the patent is “all about,” may not be the focus of patent litigation.

Coming up in Part 4, we’ll look at:

- Investigating infringement of the facade-server claim (we’ll investigate prior art and invalidity in part 5);
- Brainstorming search terms when using a patent claim to find infringement;
- Testing a possibly-infringing product for infringement;
- Conducting a limitation-by-limitation comparison of a claim to an accused product (we’ll see in Part 6 that Local Patent Rules in key federal districts *require* this limitation-by-limitation comparison early in the litigation);
- Examining “means for...” and other forms of functional claiming; and
- Examining the Doctrine of Equivalence, and the function/way/result test for equivalence.

Andrew Schulman is a Senior Software Litigation Consultant at [DisputeSoft](#). He focuses on software patent litigation, pre-litigation investigations, and source-code review. Mr. Schulman is also the founder and principal of [Software Litigation Consulting](#). As a software engineer, he edited and co-authored several books on the internal operation of Microsoft operating systems, and is an attorney with an LL.M. in Intellectual Property.



If you are in need of a software patent dispute expert, we invite you to consider [DisputeSoft](https://www.disputesoft.com).

Contact Information

Jeff Parmet, Managing Partner

301.251.6182

jparmet@disputesoft.com

12505 Park Potomac Ave. | Suite 475 | Potomac, MD | 20854