Patent Drafting: Software as a Service (SaaS)

by Robert Plotkin, BlueShiftIP, with Practical Law Intellectual Property & Technology

Law stated as of 04 Nov 2024 • USA (National/Federal)

An Article discussing an approach to patenting Software as a Service (SaaS) inventions in the US Patent and Trademark Office (USPTO). It outlines the challenges in patenting SaaS, best practices, strategic considerations, and drafting claims and specifications to maximize commercial value including for enforcement. This article also addresses patent eligibility, functional and structural claim language, and single entity infringement.

The SaaS Model

Software Patent Eligibility

Challenges of Patenting and Enforcing SaaS Patents

Divided Infringement

SaaS Best Practices

Claim and Specification Drafting Strategies

Drafting Claims for Single Entity Infringement

Comprehensive Claim Sets

Functional and Structural Language

Proprietary **software** is a critical asset for many businesses. Whether a company licenses its software to others, uses it to deliver services to customers, or employs it for internal operations, proprietary software can provide a significant competitive advantage that is safeguarded through legal protection.

Software as a Service (SaaS) has emerged as a dominant model for delivering software solutions. Before SaaS, vendors and customers bought and used increasingly powerful computers to run increasingly powerful software locally. Users often purchased this software on physical media, such as CDs, USB drives, or as a downloadable package, that could be installed on their local devices. This traditional model typically involved a single customer transaction for a perpetual license.

SaaS presents unique challenges due to its distributed, dynamic, and transitory nature. Despite these complexities, obtaining strong and enforceable patents for SaaS technologies is critical to protect and leverage these innovations in a SaaS-dominated market. By understanding the specific hurdles, such as the need to avoid divided infringement and the importance

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.

of crafting claims that remain relevant over time, patent counsel can strategically obtain and enforce SaaS patents that offer robust protection.

For more information on patents generally, see Practice Note, Patent: Overview.

The SaaS Model

SaaS is a software delivery and licensing model through which providers remotely host, manage, and support software applications for customers over the Internet or a private network. SaaS comprises a one-to-many access method derived from the application service provider model. This allows a vendor to host the same version of a standard software product for multiple customers to use, typically via a subscription, rather than purchased and installed on individual computers.

SaaS involves a combination of:

- The software and the technical mechanisms by which the software is delivered to its users, such as distributing the software across a client and a server.
- The legal constructs that govern ownership and use of the software, such as an end user license agreement which grants the customer rights to use the software in specified ways.
- The software's payment terms, such as a one-time fee for perpetual use or, more commonly, a recurring subscription fee.

For detailed information to consider when negotiating and drafting SaaS agreements, see Practice Note, Software as a Service (SaaS) Agreements and Standard Document, AI SaaS Agreement.

Software Patent Eligibility

Patent requirements are notably more stringent than those for other intellectual property, such as **copyrights**, and have become increasingly so due to the federal courts' growing skepticism regarding the patent subject matter eligibility of software-implemented inventions. Applicants for patents on these inventions typically claim them as novel methods, systems, machines (such as pre-programmed, special-purpose computers), machine-readable physical media, or a combination of these technologies. However, courts rigorously examine the patent eligibility of these inventions since patent claims often recite or involve abstract ideas, such as mathematical formulas and algorithms, without adequate limitations to restrict the claims to specific applications.

To increase the likelihood of drafting an application with patent-eligible subject matter, patent counsel should draft the patent application to support the claims by:

- Describing the system and processes from a technological standpoint.
- Discussing improvements to the functioning of the computer, with explanation detailing how the invention improves on the prior art. Consider analogous decisions to identify improvements likely to be considered patent-eligible.

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.

- Using definitions and technology to facilitate an understanding of the invention.

- Describing the specific and unique characteristics not previously used in the claimed subject matter.

For more information on drafting to support patent-eligible subject matter, see Practice Note, Patent Drafting: Claiming for Proper Interpretation in Computer-Implemented Inventions: Drafting Considerations and Challenges for Computer-Implemented Inventions.

For an overview of patent subject matter eligibility, see Practice Note, Patent-Eligible Subject Matter: Overview.

For more information on patent protection for software, see Practice Note, Legal Protection of Software: Patents.

Challenges of Patenting and Enforcing SaaS Patents

The advent of high-speed, wireless, always-available internet connectivity shifted this model towards cloud-based software execution, with our local devices acting mainly as "thin clients" that interface with remotely executed software. This connectivity eliminates the need for physical distribution and local installations, offering a more dynamic and flexible approach to software delivery. However, certain functions, such as highly sensitive encryption and low-latency tasks like conversational AI, still need to be performed locally. This results in a complex and fluid division of labor among distributed systems.

SaaS's technological and commercial benefits appeal to vendors and customers, underscoring the need for robust patent protection for SaaS innovations. However, the features that make SaaS attractive also introduce unique challenges in obtaining and enforcing strong, broad, and defensible patents. These challenges include single entity infringement, patent-eligible subject matter, and sufficiently flexible claims. Moreover, the distributed nature of SaaS, intricate client-server interactions, and diverse legal constructs governing its use add complexity to patenting this technology.

For an overview of the cloud and distinctions between the cloud, **cloud computing**, and cloud services, see Practice Note, IT Basics: The Cloud: Overview: The Cloud Defined.

Challenges of patenting SaaS include:

- The transitory nature of SaaS. SaaS is accessed over the internet rather than installed locally on devices. This transitory nature poses challenges for patenting. Products and processes qualify as patent-eligible subject matter more easily when they take a tangible and non-transitory form. Inventions directed to transitory propagating signals:

    - do not fall within any of the statutory categories of process, machine, manufacture, or composition of matter; and

    - are not patent-eligible subject matter.

    (*Mentor Graphics Corp. v. EVE-USA, Inc.*, 851 F.3d 1275, 1294 (Fed. Cir. 2017).)

    For a collection of resources on patent-eligible subject matter, see Section 101 Patent Eligibility Toolkit.

- The distributed nature of SaaS (see Divided Infringement).

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.

- The dynamic environment. The environment in which SaaS operates is highly dynamic and constantly evolving with frequent updates and changes in how the software is deployed and used. This fluidity can make it difficult to draft sufficiently flexible patent claims that remain enforceable while capturing future software improvements.

Divided Infringement

SaaS systems are inherently executed in a distributed manner, often involving multiple servers and clients that interact over a network. A fundamental challenge of effectively patenting SaaS innovations is that drafting patent claims to avoid divided infringement, whether in method or product form, where a single party directly infringes by performing the method or makes, uses, or sells the product is complicated (see Practice Note, Patent Infringement Claims and Defenses: Patent Infringement Claims).

For information on drafting claims, including avoiding divided infringement, see Practice Notes:

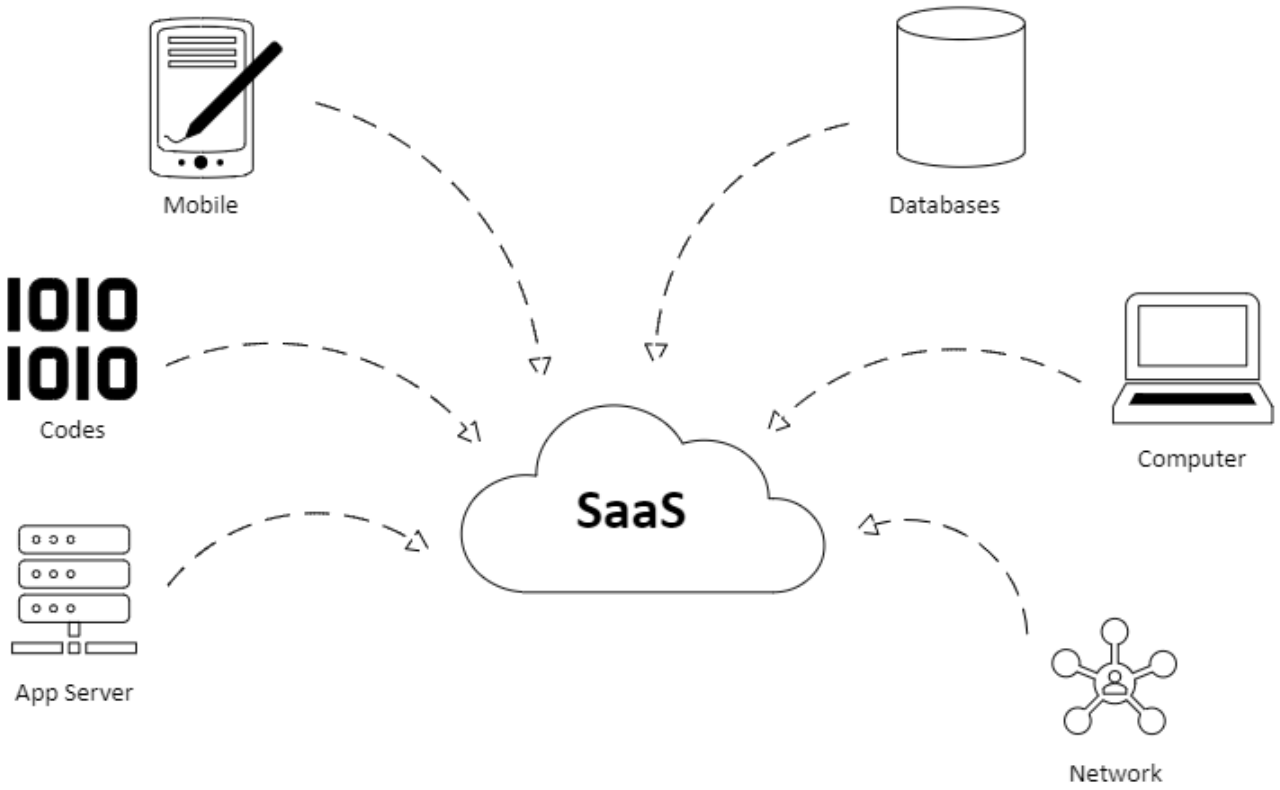
- Patent Drafting: Claiming for Proper Interpretation in Computer-Implemented Inventions.
- Patent Drafting: Chemical and Pharmaceutical Claims.
- Patent Litigation: Infringement: Divided Infringement.

Consider the example of a traditional consumer product, such as a mousetrap. A new type of mousetrap is typically straightforward for a skilled patent practitioner to capture in a product claim because the entire invention is embodied in a single tangible object. When someone makes, uses, or sells the mousetrap without permission, it is clear to identify when a single entity directly infringes the patent. Although these products may be sold as kits or in other ways that pose infringement challenges, they are relatively discrete and self-contained units that patent counsel can map onto individual patent claims to prove infringement.

In contrast, SaaS's inherently transitory, distributed, and dynamic nature means that different parts of the software and its functions are often spread across multiple entities, complicating the identification of a single infringing party. This transitory characteristic makes it challenging to obtain patent claims for SaaS inventions that are directly infringed, requiring careful and strategic claim drafting to address these complexities. For information on direct infringement, see Practice Notes, Patent Infringement Claims and Defenses: Direct Infringement and Patent Litigation: Infringement: Direct Infringement.

An example of a SaaS system with functions spread across multiple entities is shown below:

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.



Although other kinds of software can be transitory and dynamic to some extent, non-SaaS software typically executes on a single computer, making it easier to identify and prove direct infringement. Enforcing patents against SaaS implementations carries with it all of the challenges of enforcing traditional software patent claims, plus the added complexities associated with the distributed nature of SaaS.

For a detailed discussion of drafting challenges for computer-implemented inventions generally, see Practice Note, Patent Drafting: Claiming for Proper Interpretation in Computer-Implemented Inventions: Challenges for Computer-Implemented Inventions.

SaaS Best Practices

Obtaining enforceable patents for SaaS requires:

- Strategic claim drafting (see Claim and Specification Drafting Strategies and Practice Note, Patent Drafting: Claiming for Proper Interpretation in Computer-Implemented Inventions).
- Extensive specifications (see Structural Support in the Specification and Practice Note, Patent Drafting: High-Technology and Life Science Invention Specifications: Computer-Implemented Inventions).
- A deep understanding of how SaaS can be implemented in a variety of technological and commercial settings.

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.

These foundational elements are useful for writing SaaS patents that are as broad and flexible and can be asserted against a wide variety of ways in which a SaaS invention may be implemented across devices and infringing parties.

Claim and Specification Drafting Strategies

Patent counsel can also use key strategies to increase the likelihood of direct infringement by a single party and to maximize the ease of enforcing these patents. These strategies include:

- Drafting claims to cover a single entity (see Drafting Claims for Single Entity Infringement).
- Writing comprehensive claims sets provide broad protection and different aspects of the invention (see Comprehensive Claim Sets).
- Using functional and structural language in the claims and the specification to cover various technology implementations (see Functional and Structural Language).

Drafting Claims for Single Entity Infringement

To minimize the risk of divided infringement, patent counsel should craft patent claims to ensure that all steps of a method or all product components are likely to be performed or used by a single entity. Typically, it is most desirable for that entity to be the SaaS provider because this entity is more likely to infringe on a large scale than individual customers of the SaaS solution. Patent counsel should consider claims:

- That focus solely on server-side operations, rather than including both server-side and client-side operations in a single claim. An independent claim that is likely to be infringed solely by the server will reduce the chances of divided infringement.
- That would be infringed solely by the client, even if they are less commercially valuable than server-side claims. In some cases, the client side of the system may be implemented by large infringers.

For information on direct and divided infringement, see Practice Notes, Patent Litigation: Infringement: Divided Infringement and Patent Infringement Claims and Defenses: Patent Infringement Claims.

Patent counsel should also consider pursuing claims that include both the server and client. Although these claims are more likely to encounter divided infringement, they may be more easily patented, whether due to the ease of demonstrating patent-eligibility for a multi-computer system or the ease of proving nonobviousness for a method that is distributed across multiple computers in an unusual way. In any case, if the server and client are owned or operated by a single entity, divided infringement may not be an issue.

Comprehensive Claim Sets

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.

Patent counsel should draft claims as a coordinated set, including:

- The desired invention subject matter.
- Appropriate statutory claim types, such as method, product, and system claims, with differing claim scope.

A combination of method and product claims in a SaaS patent can provide broader protection and capture different aspects of the SaaS technology. Different claim combinations may include:

- Method claims to detail the processes performed by the SaaS platform. Ensure that these claims are structured to avoid divided infringement by focusing on steps likely performed by the SaaS provider.
- Product claims to cover the software architecture and components, including client-server interactions. This helps in targeting the SaaS provider who controls the overall system.
- System claims that encompass the entire SaaS platform, including both client and server components. These claims can address the interaction between different parts of the software, providing a holistic view of the innovation.

While it might seem that method claims are inherently preferable over system claims for SaaS inventions because of the service-based nature of SaaS, system claims also have high infringement value if a SaaS developer sells its solutions to other parties but does not provide the SaaS-based solution as a service itself.

For information on the claim drafting process, see:

- Practice Notes:
  - Patent Drafting: The Specification: The Claims;
  - Patent Drafting: Claiming for Proper Interpretation in Computer-Implemented Inventions; and
  - Patent Drafting: Chemical and Pharmaceutical Claims.
- Patent Application Preparation and Filing Checklist: Draft the Claims.

Functional and Structural Language

As part of a comprehensive strategy, patent counsel should consider obtaining claims that employ functional and structural language in the context of SaaS inventions (see Functional Language and Structural Language). Using functional and structural language in SaaS patent claims provides a powerful and comprehensive approach to securing robust patent protection.

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.

Functional language offers flexibility and broad coverage, capturing various technology implementations and adapting them to rapid advancements. Structural language, on the other hand, ensures precision and clarity, highlights specific technical innovations, and helps the identification and enforcement of infringements.

By combining these approaches, patent claims can effectively protect the core functionalities of SaaS technologies while also detailing the unique configurations and components that make them innovative. This dual strategy maximizes the scope of protection, enhances the defensibility of patents, and ensures they remain relevant and enforceable in a dynamic technological landscape.

Functional Language

Functional claim language describes what the invention does rather than what it is (*Swinehart*, 439 F.2d 210, 212 (CCPA 1971)). This approach offers several benefits in the context of SaaS. For example, SaaS technologies are often implemented in various ways across different platforms and environments. Using functional language:

- Helps cover diverse implementations without being tied or limited to specific technical details. For example, a claim for "managing user authentication" can encompass various authentication methods, such as passwords, biometric, and two-factor, and technologies, such as cloud-based, on-premises, and hybrid systems.
- Ensures that the patent remains relevant and enforceable even as the underlying technologies evolve. The SaaS landscape is dynamic, with frequent updates and advancements in technology. For example, a claim for "real-time data synchronization" remains applicable whether the synchronization uses traditional database methods, distributed ledger technologies, or future innovations.
- Provides broader coverage, making it easier to capture infringers who perform the same overall function, even if they use different technical approaches. For example, SaaS involves distributed systems where different components interact over a network. Functional claims can simplify enforcement by focusing on the overall function rather than the specific distribution of tasks across components.
- Offers flexibility, broad protection, and adaptability, which are crucial in the fast-evolving and diverse SaaS landscape. Functional claims help capture the essence of the service, simplify enforcement, and align with the dynamic nature of SaaS, ensuring robust and lasting patent protection.

For information on functional claiming, see Practice Note, Functional Claiming for Computer-Implemented Inventions.

Structural Language

Structural claim language specifies the particular components and their arrangements within the invention. A structural claim term should either:

- Itself provide the structure for the term because it is generally known in the art to designate structure.

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.

- When read in light of the specification, inform a **person of ordinary skill in the art** (POSITA) of the invention's scope with reasonable certainty.

(MPEP § 2173.)

Using structural language can provide a variety of benefits for SaaS-based inventions, including:

- Helping to clearly define the technical aspects of a SaaS invention, which can be useful for demonstrating patent eligibility. For example, claim limitations that refer to specific types of hardware components (for example, a particular GPU) can be useful for demonstrating that the claim is not directed merely to an abstract idea. Even software-based structural limitations can be useful for this purpose, such as "A system comprising a client device configured to transmit user data, a server configured to receive and process the data, and a database for storing the processed data."

- Demonstrating novelty and nonobviousness by overcoming more generic **prior art** (for information on novelty and nonobviousness, see Practice Note, Patent: Overview: Novelty and Nonobviousness). For example, an encryption algorithm may use well-known steps such as hashing, which may prove difficult to distinguish over the prior art even if the claimed method makes a new use of well-known steps. Structural limitations, whether in the form of hardware or software components, can more clearly distinguish the claims (for information on prior art references, see Practice Notes, Prior Art: Determining the Status of a Reference and Prior Art: Public Use, On Sale, Knowledge, and Availability).

For an example of structural claim language, see Practice Note, Functional Claiming for Computer-Implemented Inventions: Box, Structural Claim Language.

Structural Support in the Specification

Patent counsel should include a description of the claimed invention's physical components in the specification. Too often, patent applications for software-implemented inventions lack reference to a processor or other physical component. Even more often, they fail to tie together the inter-reliance between the physical components and functional elements.

To maximize the value of claims to a SaaS invention, the specification should include:

- Detailed text that describes the physical components of the invention.

- Figures showing those physical components.

For more information on drafting patent specifications, see Practice Note, Functional Claiming for Computer-Implemented Inventions: Include Structural Description in the Specification.

For an example of structural language in the specification, see Practice Note, Functional Claiming for Computer-Implemented Inventions: Box, Physical Structure Description in the Specification.

 © 2024 Thomson Reuters. No claim to original U.S. Government Works.