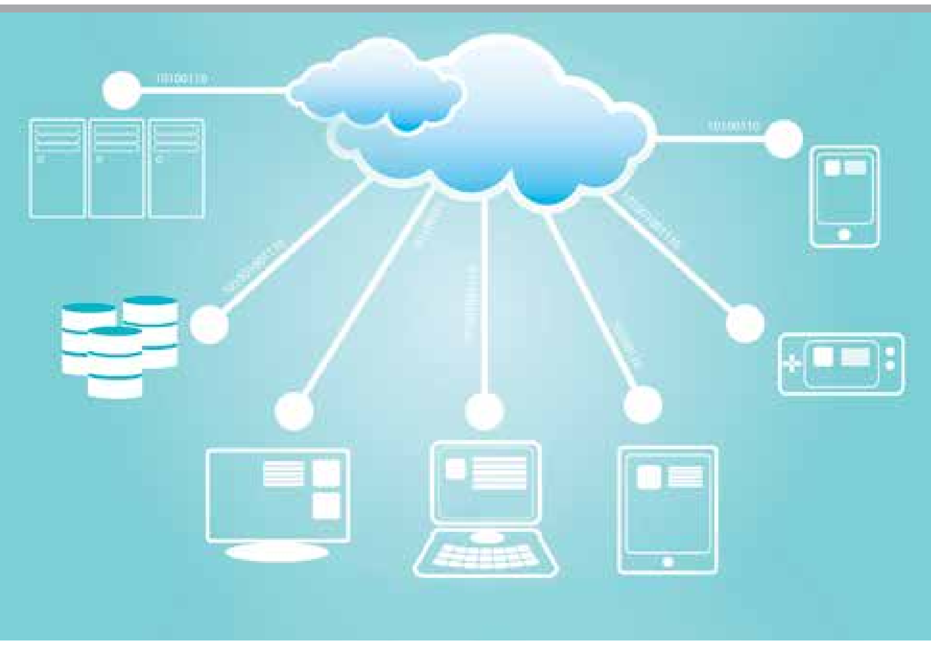
SaaS SOS

Increasingly, software companies are modifying their business models to software as a service. This change brings with it new legal challenges for the industry, **Celia Lerman** explores



Traditionally, software was sold as a product. Alongside hardware, it was sold as an additional product. Who does not remember buying an old computer that came full of diskette packs to install programs? Or even going to electronics store to buy the latest version of MS Office?

That business model has evolved along with storage technology and telecommunications. As the discs became more compact, the data transmission technology became even faster, and as the internet was adopted, software was sold online. As happened in the world of music, software began to be commercialised independently of any physical support: whoever wanted the program could pay for it and download it directly. The program, nevertheless, was structured like a product: a licence to the program was acquired, generally perpetual or of a long duration, that allowed access to minimum periodic updates, thus limiting interaction between user and software company.

2013 was a turning point from physical software to online subscriptions and shortly after, SaaS began to develop, killing off locally installable software.

Why Saas?

'Software as a service' (SaaS) is run and managed by the providing company's web server, and includes (usually web- or app-based) interfaces for end-users who pay a subscription fee. This has increasingly become the standard in the software industry. It has many benefits, especially for software businesses – although not so much for users.

Business benefits

Companies generally charge monthly subscriptions (usually with annual options at a discount), which replace the "one-time payment" of the previous product. Having an insured revenue stream on a monthly/annual basis guarantees a secure income, enabling better projection and financing of new developments. This model, which usually takes payment by credit card, enables collection when the user does not cancel during the contract period

Under SaaS, the company stores and manages all user data; whereas when the software is local the data is held by the user. This set-up provides two key benefits to the company: first, greater customer lock-in, since moving data from the cloud to a personal computer or another cloud system is usually a long, painful, and often expensive process. Secondly, by holding the documents, the company can gain deep insight into what the user is doing, provided the terms allow for this.

It is substantially easier to develop and manage web-based software as only one version is needed on the web server. However, the software must be compatible with a variety of operating systems and configurations.

Providing support is much easier, particularly as the most common issues relate to software installation issues.

In addition, a web- or app-based service allows the company to access user performance metrics. This enables businesse to better understand user behaviour and the use and issues with the product. This information is helpful when making improvements.

SaaS systems facilitate more easy and direct communication with users. For example, most SaaS systems include the easy-to-build functionality of system-wide messaging, to send notifications to users.

> **"2013 was a turning point from physical software to online subscriptions and shortly after, SaaS began to develop, killing off locally installable software."**

User benefits

As a direct result of the ease of notification, and the ability to measure user behaviour, SaaS allows a more fluid interaction between the company and the user. It allows users to leverage the advantages of cloud computing, most noteably, the ability to scale up the software as needed as well as the integration of additional platforms and services, including independent analytics, storage, and social media.

In most cases, it is substantially easier to adopt a SaaS due to the smaller learning curve for users – they do not need to go through complicated installations, nor upgrade hardware devices.

There is, however, a downside in that there is no longer a one-time payment, as subsciptions and licences need to be renewed.

Other downsides for users might be the loss of control over data files, the risk of the loss of access to data and the possibility of other parties (eg, government bodies via subpoenas) gaining access to files.

Legal challenges

SaaS brings with it various legal challenges.

When software was sold as a product (not SaaS), it was offered under a licence agreement. With SaaS, the key document is the terms of use or terms of service (ToS). This document, which does include a licence for use, also includes many other clauses adapted to the needs of the service's particular dynamics.

ToS clauses usually include:

- **Payments:** structure and terms, including various fees and discounts for longer-term subscriptions.
- **Limits to use:** the SaaS allows the company to monitor more closely the activity of the software user, enabling more efficient enforcement of limits on use.
- **Early termination:** in line with the payment clauses, ToS establishes termination only in certain periods and after specific periods have lapsed. These clauses are often hidden and can be confusing to users, resulting in higher charges.
- **Periodic modifications:** These can be introduced with prior notice given to users, which provides greater flexibility for the business.

The rise in privacy concerns is closely related to the mass adoption of the SaaS model. By interacting more closely with users and collecting their data, companies must inevitably take into account personal data and privacy regulations. Now that the General Data Protection Regulation (GDPR) has come into effect, a new global standard of privacy is emerging.

Copyright

It is best practice to register for copyright of a service in the US before launching it. This allows the owner to obtain protection over the code, and the look and feel. It also allows the owner to consolidate international protection of the copyright: with a single registration.

Although copyright does not protect the functionality of the software, such protection is often not necessary, either because the software is not innovative enough to obtain a patent, or even if a patent is obtained, the company may not be able to allocate the necessary resources to defend said patent.

The disadvantage of the copyright registry for SaaS is that the registry is, by definition, static, but SaaS is a digital entity in constant evolution. The registry becomes a mere "photo" of the SaaS at that particular moment, not necessarily illustrating its evolution or its implementation in different contexts, such as different software platforms or hardware devices.

In these cases, the best practice is, on the one hand, to register the code "engine" or code common to all the platforms, and independently register the different implementations. On the other hand, it is convenient to make periodic records, especially when substantial improvements are made to the technology. It is customary that at regular intervals (eg, every three or six months) the legal team and the technical team come together to evaluate the technology, the substance of the changes, and the need to record the developments.

Patents

Since the precedent-setting decision of the Supreme Court of the US in *Alice Corp v CLS Bank* (2015), it is increasingly difficult to use patents to protect software. This does not mean that applications for software patents have dropped. Although it is true that many patents have been subsequently invalidated, there are still no clear standards on what makes an abstract idea become a suitable patent application.[1]

Most frequently, software companies do not focus their IP strategy on patents. Many software companies are start-ups, with limited resources, for whom the value offered by the patent does not justify the expense it imposes. In the US, for example, the entire process of a patent application costs approximately US$30,000. Furthermore, with the cost of enforcement of the rights added, these are legal expenses that few start-ups can afford.

In addition, patents take a considerable amount of time to be obtained, which is often greater than the boom period of said technology. If the patent is only granted when the market for that technology is on the decline, then the patent will not be valuable. With this in mind, many software companies build their patent portfolios not because of their intrinsic value, but because of their competitive advantage to hinder their competitors.

Patents are generally not a central part of the software strategy of small and medium-sized companies. In larger companies, software patents are complemented by patents linked to hardware and other tangible technologies.

> "SaaS systems facilitate more easy and direct communication with users. For example, most SaaS systems include the easy-to-build functionality of system-wide messaging, to send notifications to users."

Trademarks

Trademarks continue to be a key legal tool. They protect not only the name, but also the colours, logos and all other distinctive features of the software. Trademarks are crucial to defend the use of the company's name in the domain names and social networks.

The change from software as a product to SaaS normally involves new registrations for key signs in different classes. Software as a product is normally protected in international class 9 (downloadable software), while SaaS generally involves class 42 (software services) and may involve class 38 (telecommunications) and class 41 (non-downloadable publications), together with other classes that are specific to the problem that the software addresses.

Summary

SaaS, as a business model, is useful for both large and small software companies. It enables them to grow and consolidate their businesses. However, it also creates new legal challenges. Key documents that provide protection under SaaS are the terms of use and the privacy policies, accompanied by intelligent strategies for copyright, patent and trademark registrations.

Footnote

[1] See Stroud, Jonathan and Kim, Derek, Debugging software patents after *Alice* (2017). 69 *South Carolina Law Review* 117.

Author

Celia Lerman is a partner at Lerman & Szlak and a lawyer in both California (California bar & master in the science of law, Stanford University) and Argentina (Universidad Torcuato Di Tella, valedictorian; LLM, Intellectual Property, from Universidad Austral, with honours).