Business users associations call for a balanced cloud market: 11 fair principles to unleash the digital potential of Europe

14 June 2022

Our associations welcome the Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on harmonised rules on fair access to and use of data (Data Act). They see the data act as an opportunity to free up inter-company data sharing and as an essential and coherent complement to other existing or planned regulations, notably: the GDPR, Platform to Business, Digital Markets Act and the Artificial Intelligence Act.

Regarding the Data Act, we encourage the Commission to:

- Consider the position of business users when drafting the Data Act and restore a fair balance in the data economy
- Ensure that business users keep full control over their data
- Improve data portability and possibilities to switch cloud providers
- Ensure fair contract terms between providers and business users regarding data
- Protect European business data from extraterritorial access

The 4 associations share these concerns and have invested time and effort to gather fair/unfair practices clauses in B2B contracts between business users and Software providers or Cloud service providers. This has given a long list of clauses. To be able to pinpoint the most important ones we have asked each association to put priorities on the list so that we have ended with the most important clauses.

11 Fair principles

Existing regulatory obligations shall apply

Whenever a company puts a product or a service on the market, it shall follow the existing regulatory obligations applicable. It is a basic principle in order to ensure a market to properly function.

Principle 1:
Vendor shall fulfil existing regulatory obligations
Description:
Vendors that work in a certain industry, geography or otherwise regulatory environment shall comply with applicable regulations. Vendors shall take responsibility for the data they manage on behalf of the customer. Furthermore, they shall provide cloud technologies in such a way that customers can comply with regulations they are subject to. When the regulatory frameworks evolve, vendors shall adapt or make available the necessary options in order for their customers to be able to fulfil their regulatory obligations and remain compliant. When a vendor cannot comply with (part of) a regulatiory obligation, it shall inform its customers; allowing these customers to assess their own compliance and giving them the opportunity to either implement controls on their side or stop the contract.
Examples:
Under the GDPR, protection of personal data shall be guaranteed by all parties involved in the processing of these personal data. The vendor must fulfil its duties as Processor and/or Controller, adhere to contractually agreed location of data processing (including hosting or remote support), etc. Services offered in the cloud might see themselves as location independent, but when offering the service to customers subject to the GDPR, they must themselves fully comply with the GDPR, including its data transfer obligations. Vendors shall adapt their offerings in order to be compliant. We know of several cloud providers

which, through Data Protection Impact Assessments (DPIA) or GDPR supervisory authorities, have been found not to comply with the GDPR.

Ensure easy switching to avoid lock-in

The organisations we represent gain their bread and butter in a variety of sectors, delivering almost all services available on the European market. Software and cloud services are important – if not decisive - production factors, next to many others.

At the consumer level, it is widely accepted that lock in happens when switching is difficult. Switching apps and regulatory intervention have been used to facilitate switching. In several markets, where economies of scale and network effects are present, such as telecommunications, this has been the case.

Given the specifics of the current software and cloud markets, even large companies are facing lock in. When switching costs are high and technical standards are diverse, customers find it difficult to switch cloud providers. Being forced to work around the resulting complexity they tend to remain with the same cloud provider and are pushed to contract even more application from the same cloud provider. To create a lively market, switching shall be guaranteed through ensuring that vendors don't create technical or commercial lock-in.

Principle 2:
Vendors must not create a technical, organisational or commercial lock-in.
Description:
Software vendors should adhere to open technical standards where such industry standard exists. Customers cannot be intentionally restricted in any commercial or technical matter to exercising their right to port their data from one vendor to another, switching providers or regaining access to their data. Interoperability between non-proprietary or comparable technologies shall be technically supported. Inter cloud data transfer should be possible, allowing a business user to shift data to other cloud provider for processing at will. In order to allow switching, the data shall be handed over in a non-proprietary form such as .txt or .csv. or human readable format. It shall be clear what type of information it contains: database back-up, images, words, figures, drawings, etc…
Examples:
When wanting to change vendor and requesting their own data, customer receives their data in a format linked to the vendor. While in theory the customer got their

data back, in practice the data are only accessible through the use of a format that is the property of the vendor, which makes the customer dependent on this former vendor even after termination of the contract and allows the vendor to still impose restrictions, increase costs and cause delays. Open standards for virtual workload in the cloud do exist, but the standards are not adopted or uniformly applied. When the standard exists, vendors shall apply it, so that switching is possible. Without advocating the standard, but as an example, cloud foundry for container workloads is such an example. As a consequence of lock-in, customers take non-productive decisions for deploying their software to avoid increased fees. The option to change vendor has become uneconomical. Customers thus stay on a legacy or non-supported platform which is a risk for operations, because changing to a newer or different environment requires unjustified software investments.

Contractual terms and conditions

Employees and employers sign an employment contract with terms and conditions. A vendor that has worked on a ICT service or product, concludes an agreement with their customer, stipulating what the service is, what the payment is, etc… in the terms and conditions. The terms and conditions form the basis of the relationship they enter into.

The terms and conditions can change as the market evolves and as the technologies mature and are replaced. Customers need different services, invest or divest, etc… Terms and conditions reflect the past and current state of the market and are a force to shape the future. Mostly, they are the proofs of the relationships between vendors and customers.

Unfortunately, given the structure of the market and the asymmetric bargaining power the terms and conditions we currently see, don't reflect a thriving market, where both vendors and customers find space to evolve. Current terms and conditions are in many cases to the disadvantage of the customer. Those terms and conditions shall be changed along the different principles outlined below.

Principle 3:
Customer shall remain in control of their own data and all the data uploaded or processed by the service/solution.
Description:
The standard terms and conditions should specify the customer's right to their data, the processing and the restrictions. Data that has been processed and potentially enriched by the vendor solution and relevant meta data, shall still

remain solely property of the customer. The proprietary algorithms remain under ownership of the vendor.

Vendors shall not reuse data of the customer for their own purposes (such as service improvement or statistics).

Examples:

There are various examples where vendors consider the use of the service as the consent by the customer to allow the vendor to process and use the data outside of delivering the services

A platform to manage stock-options for customers: Based on the data the customer provides and the appropriate plans, the vendors considers that the outcome of the calculations is the property of the vendor. All data to make the calculations come from the customer, the calculation method comes from the platform, but the platform considers that the new data, the calculations, are its property.

A data mining service, which works solely with the data of the customer, considers that the outcome of the process, again solely based on the data of the customer, can be sold to other companies. For the customer, the algorithms are the property of the data mining service and the customer pays for the implementation of the service. For the service, the knowledge gained through the data mining service, is now considered as the property of the service. To make it even worse, the customer noticed that the knowledge and improvements gained through optimising the customers processes, were presented as best practices and offered to competitors of the customer. The value derived from the data shall be fairly distributed.

Here a positive example to illustrate how customers can remain in control of their data, through terms and conditions. A software provider has developed an application to manage all the functionalities required by insurance brokers to manage their business. The SaaS contract between the software provider and each broker is clear:
"The software provider will host on its servers – or those of a cloud provider – all the data of the insurance broker … The data of the insurance broker hosted on the software provider servers – or those of a cloud provider – REMAIN FULL PROPERTY of the insurance broker."
The software provider guarantees explicitly that he – or his suppliers – will not access the data and will not make any use of the data except those described in the present contract or the law. The software provider engages explicitly to guarantee the perfect confidentiality of the data to which he would have access. Each of the insurance brokers using the software remains full owner of their respective data so that there are in fact as many separated databases as there are brokers using the software and each broker decides to which user he gives access.

Principle 4:
Contractual terms and conditions shall be clear, unambiguous and not unilaterally changeable.
Description:
Clear and unambiguous Contractual terms shall be written and agreed in generally understandable and unambiguous language. Capitalized terms shall refer to an explicit definition in the agreement. If terms are uncapitalized, they will refer to a generic, non-proprietary use of the word in common language. Definitions shall be clear and concise as such that the signing parties are able to determine their obligations. Organizations conclude contracts with various vendors, but there is no consistency of contract terms between them. The same metric can be defined and calculated in very different ways across vendors. As such, uncapitalised terms shall refer to generic, non-proprietary use of the word. Minimum terms and conditions The terms and conditions should cover at least: metric definition, solution descriptions, restrictions, term and termination rights (including exit clause), anniversary and renewal notification, responsibilities and restrictions. In case terms and conditions are not explicitly included in a signed agreement, the terms and conditions at the time of signature shall apply. The vendor should not apply different terms and conditions to a contract which has been agreed prior. The vendor will ensure the customer has access, either publicly or upon request, to the terms and conditions that apply from the time of signature. The vendor should make available at least the following information for customer's contract management: name changes, support lifecycle and price lists. Not unilaterally changeable Negotiating terms and conditions has a reason. The terms and conditions are written and agreed so they are made static in the applicable contract, terms of purchase or any annex hereto. The terms and conditions can be located in a master agreement, framework agreement, license agreement, or any downstream order form, purchase document referring to the main agreement structure. The practice to refer to terms and conditions by reference to a website or online location, URL conditions, shall be agreed by both parties explicitly, as customer has no control over unilateral changes or updates to the terms and conditions. URLs are often recursive: the URL general terms and conditions refer to other URLs for service level agreements, data processing agreements, product specifications, etc. Vendors can be allowed to make changes to the language of the agreement, on the condition that there are no material nor financial adverse effects to the organisations contracting, usually referred to as the customer and the customer is notified upfront with the ability to review and reject the changes.

Examples:

"use" vs "Use"

1. The uncapitalized word "use" refers to the common definition putting something such as a tool, skill, or building to a particular purpose. The customer has a benefit of a certain asset or service from the vendor. The capitalized word "Use" can refer to a verb or definition which is more broad than "use". "Use" can refer to the ability to use a service or software or the availability of a certain capacity which has never been put to the benefit of the customer.

   Typical wording that extends the definition of use beyond what is reasonable accepted and thus should be capitalised: *"[use] every device that has the capability to execute the program*" In this case, even not used, but with the capacity, it is counted…"

   The change towards the possibility that something is used was included intentionally. A customer that gave the possibility of several administrators to access a server, for example for reasons of ensuring continuity when the responsible administrator would be impaired, had to pay for that possibility, even when the customer could prove that only one administrator had accessed the server.

2. Typically and intentionally listing possible scenario's to make the definition of the common understanding of use broad, and redefining it as 'Use': "*[use] Customer's installations, deployment, access of or provision of access to, or use of each Product*". Agreeing to the definition of "use" in this case has a negative effect on the calculation of fees. Provisioning something in this case lead to higher than expected fees. For a specific vendor, it seems the business model is based on making terms unclear, with the result that customers are charged more than the customers expected. The vendor buys software that is losing market share. Through a change of the definitions combined with an aggressive control and audit policy, the vendor put pressure on the just acquired customers to pay more.

Agreement to commercial documents lead to change in terms and conditions

When a contract requires extension, for example a yearly subscription comes to expire, vendors often include their latest terms and conditions by referring to the URL conditions, in the renewal documents. When signing the renewal documents, the initially reviewed and agreed terms in the frame agreement which formed the basis of the initial contractual relationship are overridden and voided. The new terms and conditions are not reviewed and agreed and may have a negative effect on the obligations of the customer under the initial, fully negotiated agreement.

Principle 5
Contractual terms shall not restrict or discriminate for customer's choice of cloud provider, outsourcing partner or hardware platform.
Description:
Customers having purchased or purchasing software, shall have the possibility to deploy and use the software on the platform of choice or with the cloud provider of choice. The terms and conditions, including commercial conditions, shall be non-discriminating and uniform between running workload in the cloud, on premise or in any hybrid setup with comparable workload and performance. If the technology and workload is comparable and common, the customer shall have freedom of choice. Customers who are moving on premise workload to cloud providers should achieve a cost neutral shift that safeguards the investment on comparable performance and workload. Cloud infrastructure has the same or comparable computing power and has the same limitations as on premise workload, with the sole exception that the underlying hardware is owned by a third party.
Examples:
Various examples exist of restricting choice: Bundling software and infrastructure Software vendors create a self-maintained list of authorized public, private or hybrid cloud infrastructure on which customers are entitled to use their purchased licences. More concretely, a major vendor prohibits the use of a software on the infrastructure of a major IaaS provider, while 2 others are allowed, although under certain conditions. The list is subject to change and only contains a subset of common cloud providers in the industry. As the list is self-maintained, a customer runs the risk that at a future point in time, the current infrastructure cloud provider, can no longer be used. Through this practice, customers are at the mercy of the vendor for what they can use. On top of that, cloud providers bundle their services and give enormous rebates. Customers are free to run a service elsewhere, but they are uncertain of the future and forgo enormous rebates. Limiting support for a software when changing infrastructure cloud providers Next to including software, also the practice of stopping the support contract when moving to a competing cloud provider. The software vendors prohibit the usage of licenses on public cloud (IaaS), by restricting the use rights under the support agreement.

While the customer can transfer the licence to an IaaS environment, the support is not included anymore. So, even if customers maintain an uninterrupted support stream, the eligibility for running the same licenses in an IaaS environment stops at the sole decision of the software vendor itself, through the use of the support contract. One dominant IaaS provider only offers licence mobility within their own products, unrelated to the IaaS service.

One software vendor provides an incentive of running the customer's perpetual licenses on his IaaS cloud in favour of all others, by giving more use rights per license if selecting his IaaS cloud service. A customer runs the software of a vendor on premise, in its own data warehouse. For the use of the software, the customer has paid a licence fee and pays a maintenance fee, a scheme which was and is standard. When moving to the cloud and looking for a IaaS provider, the vendor of the software allows for up to 90% conversion of the initial fees and all the maintenance of the software, when moving the vendor's software to the vendor's IaaS solution. If moving it to a competing IaaS solution, the customer would have to pay the full price. The benefit here is disproportionate and restricts freedom of choice.

Principle 6:

Contractual terms for licensing and subscriptions shall be free from geographic and entity restrictions.

Description:

Customers purchasing licenses, subscriptions or services should have the ability to act on behalf of the enterprise they are part of. Customers undertake mergers and acquisitions, have subsidiaries in different countries and need to be agile, including the software they need. Licences need to be able to travel with the production or sales between different sites, independent of location.

Vendors should not restrict the usage to only the contracting entity and customer should have flexibility to use the purchased subscriptions within the same enterprise, group of companies that form a holding or group of entities with the same objectives or otherwise connected. All current and future entities should be covered, as long as they are majority-owned. Vendor should accept the 'customer' definition in accordance with the intended use and contracted scope of the services/solutions for the contracting entity and all affiliates under the same definition. Customers do accept that such entities are still subject to laws and regulations, such as import and export restrictions, embargoed and sanctioned countries or other legislative restrictions.

Geographic and entity restrictions work against central purchasing or IT departments who are a separate legal entity for purchasing and distributing licenses within an enterprise or group. Some of the benefits of structuring

companies are simply annulled by limiting the use of licences base on geographic and entity restrictions.
Examples:
Sometime, terms explicitly restrict the use to a single legal entity: *""Licensee" means (a) the company or other legal entity on behalf of which (name vendor) are acquired* […] *For clarification, "You" refers only to a single, specifically identified legal entity or individual, and does not include any subsidiary or affiliate of any such legal entity or individual or any other related person."* Or legal terms with entity restriction simply in it: *"We grant you a […] licensee to deploy the Software within the Territory"* and *"Territory means the country or countries in which you have been invoiced, except as otherwise provided in the Product Guide. If the Territory for your Software includes any European Economic Area member states or the United Kingdom, you may deploy that Software throughout the European Economic Area and the United Kingdom."* This term is a publicly available part of a contract terms of a major software provider and shall not hold, as they restrict the use of the licence based on territory. Global deployment rights are unilaterally revoked by the supplier, the customer will only get them back if an amount is deposited in a 'fund' from which additional licenses can be purchased later (regardless of whether the user needs them or would like to purchase from the supplier).

Principle 7:
Contractual terms shall allow customers to use progressive or innovative technologies and deployment models.
Description:
Software terms and conditions are always based on underlying technologies. But, when underlying technologies progress faster than the software terms, a disconnection between the usage, now based on the advanced technology and the software terms is created. The software terms still refer to an old concept of calculating the required entitlement, while the new technology has progressed to a different licencing concept. Vendors use software terms and conditions in their favour if they can be interpreted in such a way that the customer needs incremental investment to cover the same functionality, while the technology allows for better use. Therefore, software terms should within reasonable timeframe adopt licencing rules that give the benefit of technological progress to the customer. More granular and efficient management of compute resources should be recognized and supported by a vendor offering to run the software on such advanced technologies.

Customers notice that technically, the vendors are up to speed and have embraced the new development. When contacting the vendors support staff, for which customers pay through maintenance contracts, the vendors are able to solve the issues. But, when the contracts are discussed, the vendor ignores the technological change and works as if the software is still running in the previous environment.

Apart from increased efficiency and thus potentially less revenue for the vendor, new technologies also provides more certainty in terms of business continuity and a more agile organisation. These are advantages where the vendor is not impacted, but come to the benefit of the customer. In these cases, the vendor applies the old framework, which leads to a more difficult business case or even the customer not adopting the new technology.

Examples:

Optimisation of computer power by virtualisation and the use of containers

1. Processors: Software terms refer to processor licences as traditional physical on premise workload. As such, they ignore the technologies that create more advanced levels of virtualised workload. Customers and infrastructure providers moved from physical compute power, to virtual compute power and more recently to containers, while the software vendors create increasingly complex licence calculation to increase the required entitlements. A software licence calculation can require multiplications and divisions of different new criteria to finally end up with the elevated licence requirement. Instead of ensuring that the benefit of the technological progress is split between the users of the new technology and the providers of the technology, the software companies change their contractual terms to ensure the benefit is more difficult to archive -more overhead, less certainty. In extreme cases, some clients set-up isolated compute environments in order to full fill to licencing requirements while benefitting from the technology, going against best practices for disaster recovery and back-up. A processor definition in the current context is no longer a physical electronic circuitry that executes instructions, but often a virtual and shared unit of processing with abstraction of the physical device or environment.

2. Desktops: Software terms refer to a ‘desktop’ or ‘seat’ as a traditional physical device which remains at the office and don’t take into account the current way of working in the modern workplace. Mobile devices, multiple devices per user, virtual devices accessible from any location are common practice. Interpreting the traditional metric in the current environment, with virtualisation, leads to a very complex licence calculation. Vendors try to reflect how in a world without virtualisation, the calculation would have been. One user accessing a virtual desktop can be counted as multiple installs since there is no physical limitation on the individual’s usage or location.

In order to make this more concrete, imagine we have a machine with 2 cores. The licence would be needed for those 2 cores. When going virtual in order to ensure more business continuity, a break-down of the machine would now not lead to a

stop in the available computing power, the need is still for 2 cores, but now virtual ones. Nevertheless, the vendor would argue that all physical cores (which in a server environment supporting the virtual machine would easily climb up to hundreds of physical cores) need to be licenced, even though at no single point in time, more than 2 cores would be working.

Many vendors apply such terms and conditions and force customers to licence more cores when combining it with virtualisation, making the business cases harder. Several customers have accepted to install software that tracks the use of the number of cores at work, meaning an additional workflow for the customer.

Furthermore, every vendor has invented different rules and procedures, different metrics depending on the type of core. Starting from the physical cores, going through the type of machines and versions, one has to take all the rules into account to come to the number of virtual cores that the customer has to licence. To make it worse, in some cases, when a system comes to the end of its supported live, but for whatever reason the customer can't decommission the system, some vendor switch back to counting the system by its physical core again. This has led to an unnecessary increase in complexity.

Below are some examples of the practices of software vendors.

1. One vendor starts with making the difference between on premise and cloud for the licence model. But, as technology progressed, several other options have become available: co-location, private cloud or managed datacentres. Do these newer options fall under on-prem or cloud?
   If considered as cloud, their licencing model include an extra administrative fee for the customer, in order to be able to move the software the customer already paid for to the cloud. It is an extra burden, which we also see as unfair, see principle 10. But, on top of the fee, the vendor claims the right to review their licencing if technology improves. This is a clear example of how the vendor tries to capitalise on improvements it has not contributed to.
   If considered on premise, several variables come into play: virtual or physical, dynamic allocation or not, fixed partitioning or not, multithreading with a 0,5 factor, but not in virtual setting, only physical, etc…

2. Other vendors start with the difference between physical capacity and virtual. For physical capacity, the calculations already have two metrics. Within the metrics, the customer has to take into account:
   - About 40 different multipliers depending on the name of the processor, the server model, the number of sockets and processor model number.
   - A matrix with different multipliers for the first thousands, the next thousands and so on of the resources licenced.

For their software used in virtual systems, the vendor has imposed a whole range of conditions. If the customer doesn't comply with the conditions, the virtualisation doesn't count and all physical capacity counts. If the customer has 1 virtual machine with 2 virtual cores, but working on a larger cluster with say 400 cores, it

will be the 400 cores physical capacity that will be counted. The conditions amongst others are:

- The vendor decides if a software can work on a virtual machine.
- The vendor decides if the virtualisation technology is eligible. The vendor has for example excluded older versions of virtualisation technology, forcing the clients to update the virtualisation technology. In principle, the software of the vendor has no link with the virtualisation technology.
- Some systems become excluded, depending on the end-of-life decisions of other vendors, while the newest versions of certain systems are not eligible yet. Again, those systems have no relation with the vendor, but it is the vendor, through its licencing policy, that influences the uptake or not of certain systems.
- The vendor also determines which processor technologies the customer needs to use. With a growing number of companies deciding to develop their own processors, this might create an major barrier for those companies and for technological evolution.
- The vendor also enforces a measuring tool to be installed to verify the use at the customer side. In many instances, the software doesn't work very well, creating a possible major problem for the customer. If during several years something goes wrong, the vendor might annule the virtualisation, leading to the counting of the 400 cores instead of the 2, as given in the example in the beginning.

3. One of the clearest examples of not allowing customers to take advantage of the technological progress consists of always having to take into account the physical capacity, even if everything runs on virtual machines. Like others, the vendor also imposes multipliers depending on the type of processor. When the vendor imposes that for the whole cluster and even all related clusters, a licence is needed even if the software is not running on the cluster. Even if the customer installs a measuring tool and can proof through the logs that the software has not turned on any of the other cores, the vendor still imposes complete licencing of all the cores.

Principle 8:
Service levels and product specifications shall be explicitly listed and take into account the context of the customer.
Description:
Software vendors offering cloud solutions need to ensure and explicitly list the responsibility they take regarding their services. In a cloud solution the customer makes abstraction of the lower layers of the technology stack. For example, in a Software-as-a-Service solution, the customer does not have access to the physical infrastructure, the network and the storage, but instead only consumes the front-end application or website. The agreement should specify service descriptions,

service level KPIs, the consequences for not meeting the service levels (credits, termination rights, etc.), response and resolutions times.

Vendors shall define maintenance windows and excluded downtime so customers can align their critical business processes with the provided service continuity.

Vendors shall accept consequences which are sufficiently material compared to the impact of such failure and aligned with how critical the solution is, as well as the scope of the service for the customer. If the solution is critical for the business continuity, the vendor should not disclaim its responsibility and offer the service which is aligned with the industry expectations.

The foregoing also applies to IP infringements, regulatory compliance and waivers. A SaaS solution can use open components but also proprietary components, protected by IP. The vendor remains responsible for indemnifying and holding the customer harmless. In the case of waivers, the vendor shall also prove that it has taken alle necessary measures to ensure the regulatory compliance is guaranteed.

Examples:

The use of software is verified in some cases with licence keys. A vendor started with using a separate server to verify the key. As the customer had it server on premise, this 'extra' verification went smoothly. Afterwards, the vendor moved the verification server to the cloud. Given the many integrations, the customer could follow the move to the cloud. As the verification could not be done anymore, and notwithstanding a perpetual licence for the software, it couldn't be used anymore.

Cloud providers often do not consider themselves business critical from an operational perspective and refuse to provide strong commitments on service levels. Nevertheless, they charge a premium fee for their services. As the services are in many cases business critical, organisations do contract the service. A major cloud provider won't offer any service levels, which makes it impossible for the customers to have a predictable environment to work in. When the services break down, there is no view on how long it will take before the service is restored.

In other cases, maintenance windows are often not aligned with customer expectations or upgrade cycles are too fast for customer to perform regression testing before the cloud upgrade is required (testing the impact of the upgrade on al processes and other software connected to the software that is upgraded). Some cloud providers release a new version twice a year.

A small software solution generates serial numbers and writes these in the central ERP system. Without the software solution, the entire production of the customer would stop. Therefore, the service level for this software solution needs to be high, as it is critical for the business. But, the vendor of this particular software solution doesn't offer a high service level, because in their opinion the software only fulfils a limited function in the production process.

Commercial models

The changes in commercial models are related to the move to the cloud. Before the move to the cloud, software was hosted on-premise: Software which runs at a location that is fully part of the customer, on servers that are owned, managed and hosted at a customer site. With the move to the cloud, the software runs on servers that are located at a data warehouse that is owned, managed or hosted by a cloud service provider.

Currently, organisations are moving, have moved or are forced to move all, many or some of their applications from on premise to the cloud. The extent to which they have of will move to the cloud depends on, amongst others, the organisation's chosen strategy and the regulatory environment. The typical commercial model for on-premise is a onetime payment combined with recurring maintenance and update fees. The typical cloud model is 'renting' the application with subscription payments.

Vendors have also embraced the cloud model, terminating their offerings for on-premise software. As a consequence, organisations have to move to the cloud if they still want to use the applications. Another result is an increased dependence on the supplier of the software, as the supplier could just shut down access to the applications if a conflict arises with the customer over payments, audit results or contract renewal negotiations, instantly bringing the operation of the customer in peril.

Principle 9
Commercial models shall not be changed unilaterally and adhere to an active 'opt-in' principle.
Description:
How vendors change their commercial models Vendors offer several software applications for distinct functions. Productivity software such as calculation, word processing, email and communication, operating systems, database management software, software for sharing and storing information, collaboration tools, virtualisation software or security are all very different software applications. Yet depending on the vendor, several of the applications are owned by the same vendor. They repackage their commercially available software and make name changes and update functionality on a frequent basis. While customers are not against improvements and updates to existing commercial offerings, challenge exists when a functionality or element is removed or customers are forced to incrementally invest into new products to follow the vendor's release cycles. The changes to the commercial models or functionalities should not have an adverse impact on the functionalities and commercial conditions initially selected and agreed. Vendors should continue to offer at least the same functionality under the same contract terms and commercial conditions.

Customers should be able to rely on stable conditions for cost forecasting and application portfolio management purposes.

The practice of increasing the functionalities and increasing the prices under the same licence name, combined with creating a new package with different functionalities, creates the need to constantly follow updates and when necessary, to change to the new package, in order to just remain with the same functionalities. If not done, the customer moves to the more encompassing, more expensive package.

Maintenance and support should be part of the software terms and commercial conditions. The period of the maintenance and support lifecycle and the right included therein should be transparently communicated upfront and commercial conditions should not be unilaterally changed.

Active opt-in, not opt-in by use

Commercially available features or changes that are subject to additional fees should not be enabled by default. The customer should be transparently notified before deployment and usage, or alternatively have the ability to actively opt-in. The customer should be able to verify and prove non-usage of such commercial feature if active but never intentionally used. The customer should be able to apply corrective actions if such usage invoked a licenced feature without intent.

Customer employees, or individuals working on behalf of the customer, are typically technical resources who do not have the mandate to sign off and commit to additional fees trigger by usage of the software.

Examples:

A major software vendor increases prices under the claim of increased or improved functionality. However they do not take into account the customer's usage or alternative packages to rationalize the costs. While it might be correct that the package contains more functionalities, these are not customer driven. Furthermore, the old package is not offered anymore or the customer has to change to a package with a different name, but with has the same bundle as before and the same price. Instead of an active opt-in, the vendor applies an active opt-out.

Other vendors enable add-ons or options to a base functionality by default. The add-on and options do trigger additional license costs. Customers should have the ability to review and granularly select the functionalities to deploy and use.

A customer invests in licences for its employees to use several applications that are offered in a package or suite. The package-license allows all employees to use these applications and they do. At a certain moment the supplier decides to divide the applications in the package over two different packages, each needing a license to use the applications. The customer now needs to re-invest into new licenses in order to make the same applications available to its employees. There is no added benefit, but twice the cost for the customer.

Principle 10:
Commercial models and offerings shall be consistent and reasonable, not combining different models for the benefit of the vendor's revenue.
Description:
Subscription model or perpetual licence model, not both Commercially available products under a subscription model (i.e. cloud) should warrant the commercial flexibility and scalability as marketed. Subscriptions should therefore be a flat fee, non-committed, flexible and pay-as-you-go. Vendors should not require upfront investment or production usage fees during development and deployment. Customers shall have the flexibility to reduce the number of licences after each commitment term expires. During the committed term customers shall have the ability to scale up at the entitled price and eligible tiered pricing if applicable. Software offered as a perpetual right to use (i.e. licenses with maintenance) are a frontloaded investment, can be considered an intangible asset. This provides benefit for both customer and vendor upfront. The subsequent maintenance should be an agreed and predictable fee for the customer. Customer shall not be forced from a perpetual license model with maintenance to a subscription model during the same commitment term and without material changes to the solution's functionality. When purchased under a perpetual license model with maintenance and provided the customer uninterruptedly continues the maintenance stream, the software shall remain eligible to be used by the customer, including all benefits of the maintenance (e.g. upgrades). Vendors shall not be allowed to change the commercial model, when the customer has indeed kept all the maintenance. Mixing both models is detrimental for the customer value: Changing from perpetual to subscription is in the benefit of the vendor. The customer has done the initial investment in the perpetual licenses, after which the vendor switches to higher subscription charges. In addition, vendors requiring upfront investment under a subscription model attempt to frontload the investment, while the subscription does not represent a perpetual right to use (or intangible asset). Applying different licence models in the same organization creates inconsistencies and complexity in pricing, naming and application management. The customer has limited alternatives if the product is only continued under a subscription model: either run the software without maintenance and support or follow the vendor change to subscription model. Licencing changing the basis for the calculation within the terms Not only a change from licence and maintenance towards subscription model happens, but also from a licence per device towards a licence per user for the

same software, running on the same machines. It is clear that the use case needs to be completely re-analysed if such a change is imposed.

Examples:

A vendor obliges all existing customers with perpetual licences to move to a subscription model. Before a specific date, the conversion is without costs. Afterwards, there is a 30% price increase. The vendor frames forced move as a case of 'application modernization program'. Customers nevertheless want to remain with the previous version.

The roll-out of digital meters has suffered from the change in metrics. An ERP vendor has announced that not only internal users, but also users connecting through the internet are now defined as users for which a licence is needed. When rolling-out digital meters, that facilitate use by avoiding having to go physically check all the millions of meters in a country, all of a sudden, the utility companies saw its users increase with millions of units, obliging it to acquire millions of licences. This change to include 'indirect' use, has let to an enormous protest, but in the end, business users had to increase the number of licences and therefore costs.

Through making the best offer, a vendor of drawing software increases its client base. Three years after entering into the agreement, the vendor changes its model from perpetual licences towards a subscription model, giving its clients 6 months to change to the new model at a reduced price. Many companies changed to the new model. Unfortunately, another 12 months later, the vendor changes the model again, towards a licence per user. In one particular case an organisation had 30 licences and at no single moment it had more than 30 employees using the software at the same time. But, in total, 300 employees at the organisation used the software; although mostly to consult drawings, not to make drawings. Although the software vendor proposes to convert the licences 1 to 2 towards named licences, the organisation now only gets 60 names, while before 30 licences were enough. The organisation would need 5 times more licences under the new commercial model.

A vendor supplies a connector between two major software packages. It is a perpetual licence, combined with a maintenance contract. Without a move to cloud or other underlying technological or architectural change, the vendor changed its commercial model to a subscription only model. The vendor charges a subscription fee which exceeds the original maintenance costs. Consequently, the customer using the software simply has to pay more.

A vendor changed its licencing model from a device based towards a machine-based model. Depending on the use case, such a change has enormous consequences for the business of a company. If various people use a stock of machines together, peak moments can be coped with by increasing the number of people, without having to worry about the licence. But when a licence per user is now needed, having more people working with the same machines, the cost

increases. As such, the licence model determines the business case of the customer.

Audit

Some vendors use the change from on premise to the cloud to reduce the number of audits because they have a direct view on the usage. For various other vendors, for which the cloud didn't necessary allow a better view on the use of their software, audits are logically happening and necessary, but remain with the same difficulties as before.

The organisations we represent are well-established, they don't organise software fraud on a large scale. All in all, they cooperate and comply. If anything irregular happens, it is almost always related to unclarity in the relationship and entitlements due to changes in licenses, names of products/packages, mergers or divestments of organisation entities or unclear software asset administrations either with the customer, supplier or service partner or all. The organisations feel that the audits shall be better defined.

Principle 11:
The scope, execution and intended outcome of an audit shall be clearly defined in the contract.
Description:
The audit right in itself is not unfair if customers formally agree to it in a contract. However, the outcome, timing, objectivity and intended purpose goes often beyond the nature of a factual audit. Often there is an additional aim of selling more licenses, new products and services or pressuring the customer to a new platform or business model. Audits may not be misused for such ends. If possible the software should be self-regulating so it does not lend itself to misuse or overuse, resulting in reduction of effort for the customer to prove compliance in case of an audit. The move to cloud has facilitated audits in some cases. During an audit, customers should not be held liable for software installed by default, but never used nor activated, for example by a license key.
Examples:
In some cases, vendors use audits to pressure customers to pay for the increased use they were not aware off or were seen as a consequence of the definition of the word 'use', see principle 4. In other instances, audits happen close to or during contract negotiations, fuelling the perception that audits are used to put the customer under pressure.

A large software-supplier wants to conduct an audit but cannot provide a correct list of current licenses. Licenses are attached to non-existent legal entities, no longer operational legal predecessors and/or are not allocated correctly to entities after carve-outs.

After establishing some incompliance during an audit, a substantial claim is filed by the supplier. This will only decrease if new licenses are purchased for the product for which the most targets have now been placed within the sales organization, regardless of whether the customer needs it.

The supplier doesn't accept the cost calculation method agreed by the intermediary/implementation partner with the customer, even after that calculation has been accepted by the supplier in writing at an earlier stage. This leads to an incompliancy claim being filed, due to an almost doubling in the use of the application compared to the contract. This claim is not approx. 2x but 20x the original cost.

Two examples of increased use, through providing automatically licences when certain functions are used:

A collaboration software gives users automatically licences when they are able to install it using a company email. It is clear that those employees who organise meetings need a licence, and those simply joining a meeting don’t. In case a customer has a Bring Your Own Device policy for external consultants, who temporarily get a company email, this becomes a crucial element. External consultants don’t need to organise meetings and should only participate, but through a back door in the licence policy, the company still will have provided them with a licence if they use the company email. When auditing or verifying through the cloud use, customers are notified that they have much more licences than they provisioned for.

Another vendor of a document sharing software requires a licence to send, but not to receive documents. However, if a person receiving a documents comments on the document, the vendor requests a licence, which is automatically provided. Again, when auditing or verifying through the cloud use, customers are notified that they have much more licences than they provisioned for.