Investment Protection in ICT Sourcing

Michael Isler and Oliver Kirchner

Biographical Information

Title: Dr

Name: Michael Isler

Position or Title: Senior Associate

Firm or Place of Business: WENGER PLATTNER

Address: Seestrasse 39, CH-8700 Küsnacht-Zürich

Phone: +41 43 222 38 00

E-Mail: Michael.Isler@wenger-plattner.ch

Primary Areas of Practice: IP, ICT, Life Sciences

Education:

- University of Zurich and Leuven (lic.iur.)
- University of Lucerne (PhD)

Recognition: Who'sWhoLegal Switzerland 2013 (Technology, Media and Telecommunications)

Membership in Associations, Committees, etc.: ITechLaw, Institut für Gewerblichen Rechtsschutz (INGRES)

Name: Oliver Kirchner

Position or Title: Legal Director EMEA

Firm or Place of Business: Citrix Systems International GmbH

Address: Rheinweg 9, CH-8200 Schaffhausen

Phone: +41 52 635 7703

E-Mail: oliver.kirchner@citrix.com

Primary Areas of Practice: IT Law, In-house generalist

Education:

- Eberhard-Karls-Universität Tübingen, First State Examination in Law
- OLG Stuttgart, Second State Examination in Law
- University of London (International Programme, UCL & Queen Mary), Master of Laws (LL.M.)

Membership in Associations, Committees, etc.: DGRI Inhouse Lawyer’s Committee

1. Introduction

Procurement of ICT systems, software and associated services entails considerable long-term investment expenditure for the buyer, whilst product life spans tend to decrease in view of rapid technical evolutions and market dynamics. The ensuing risks are frequently undervalued, and discontinuation notices or changes of feature sets soon after contract closing are not uncommon. As a result of such product discontinuation or modification, the customer may have to deploy a substitute of the discontinued product or a new release, ensure compatibility with the existing system environment, evaluate data center capacities, and safeguard business continuity. The associated cost and trouble may seriously put at jeopardy the business case that was drawn-up at the time the product was purchased. If the customer decides to stay with the phased-out product, it may lose vendor support for it.

The problem is mitigated at first sight in case software or entire enterprise platforms are procured as a service (XaaS or Cloud offerings). The up-front investment is considerably lower and software operation is maintained on the service provider's infrastructure. On the other hand, the user has no control over the installed software. If the service provider changes the application, there is no possibility of rolling back to the previous version, and integration of the provided service into the onshore ICT systems and business processes remains a vulnerable point, too. Moreover, since the customer disposes of corporate data that is processed or generated on the external vendor platform, the business continuity aspect in case of service discontinuation becomes pivotal.

From vendor's perspective, addressing investment protection in contracts is equally important. Portfolio strategy, product phase-out processes and release policies are key factors for a vendor's business success. No supplier wants its customers to be entitled to stay with a frozen product, thereby absorbing scarce resources for maintenance and support of "dinosaurs" that should rather be allocated to innovative tasks.

Both in traditional ICT infrastructure sourcing and new forms of service procurement, investment protection seems to be a blind spot in law and pertinent legal literature. Hence, if the contract remains silent, the outcome will be unpredictable. This paper helps in raising awareness and providing best practices on how to negotiate investment protection in contracts, both from vendor's and buyer's perspective.

2. Product, Release and Service Life Cycles

The product portfolio is any ICT vendor's most valuable asset. Hence, portfolio management is a strategic task that requires constant review. As resources are limited, they must be allocated to the maintenance and further development of the products and services that promise most profitability and growth, whilst portfolio elements with decreasing demand or limited growth potential should be phased-out. There are other factors that may influence the portfolio decision, such as mergers or acquisitions, where redundant portfolios need to be streamlined, phase-out of database or platform standards, which require the depending products to evolve accordingly, or intellectual property disputes that force a vendor to adapt its product in order to make it non-infringing.[1]

The life cycle of a software product, a specific product release or a certain Cloud service usually iterates the following milestones:

- General availability (GA): The product / release or service is made available for purchase;
- End of sales (EOS): The product / release or service cannot be ordered any longer;
- End of maintenance (EOM): standard maintenance and support availability for the product / release or operation of service ceases;

[1] Cf. JANSEN / POPP / BUXMANN: The Sun also Sets: Ending the Life of a Software Product, <http://slingerjansen.files.wordpress.com/2009/04/sunsetting.pdf>, 2.

- End of Life (EOL): customized maintenance and support availability for the product / release at elevated fee is eventually being discontinued.

If a customer that avails of a perpetual license in locally installed software faces a discontinuation notice, it may prefer using the stable product without receiving maintenance and support services, instead of switching to a new substitute product or release. This is different in SaaS offerings though, where the vendor promises to make available a specific functionality during the term of the contract for a multitude of users. In contrast to the individualized operation of on-premise ICT infrastructure, in a multi-tenant environment all customers collectively will either have to take the upgraded service or not, unless the service provider runs parallel versions of the application, which is rarely the case though, or a dedicated (private) cloud for a given customer, which is more expensive and not available in all circumstances.

3. Negotiating Investment Protection Clauses in Infrastructure Procurement

3.1 Product Life Span and Maintenance Periods

The decision of a customer to invest in a new software product comes with the expectation that this product remains part of the vendor’s product portfolio for a certain period, in order to be able to profit from improvements, purchase additional licenses and be assured of the general availability of maintenance and support services. The vendor, on the other hand, desires maximum flexibility in the determination of its portfolio strategy, which should not be influenced by various individual customer demands and contractual commitments.

In a typical scenario, a BUYER may be expected to propose a clause along the following lines:

> *VENDOR undertakes, during a period of no less than ten (10) years from the Effective Date (the “Life Span”), (i) to keep the Software generally available, (ii) to permanently observe the evolution of technical innovations and market requirements, and (iii) to develop, on own motion, competitive improvements to the Software reflecting then current technical standards and market demands.*
>
> *VENDOR shall notify BUYER of the End of Sales Date of the Software at least eighteen (18) in advance. Should VENDOR discontinue the Software before the expiry of the Life Span, VENDOR shall, without prejudice to BUYER’s other rights and remedies under the Agreement, migrate the Software to a new, functionally equivalent technological basis at no cost to BUYER.*

> *VENDOR further undertakes, during a period of no less than five (5) years from the end of the Life Span, to generally make available maintenance and support services for the Software at competitive terms and to hold available necessary technical and personnel resources with respect thereto.*
>
> *This clause shall survive termination of the Agreement and shall apply irrespective of the terms of any maintenance and support agreement regarding the Software.*

The VENDOR, if acceptable to product life span commitments at all, will have three main areas of concern:

i. The life span, maintenance availability and notification periods;

ii. The consequences of breach of the life span commitment;

iii. The independency of the life span commitment from the uninterrupted effectiveness of a maintenance agreement.

In view of the above concerns, the VENDOR, in the unlikely event of engaging in a negotiation based on BUYER's contract template, may likely provide the following mark-up to BUYER:

> *VENDOR undertakes, during a period of ~~no less than ten (10)~~ five (5) years from the Effective Date (the*

“Life Span”), (i) to keep the Software generally available, and (ii) to use commercially reasonable efforts to permanently observe the evolution of technical innovations and market requirements.~~, and (iii) to~~ Further, it may develop, in its own discretion, competitive improvements to the Software ~~reflecting~~ taking into account then current technical standards and market demands.

VENDOR shall notify BUYER of the End of Sales Date of the Software at least three (3) ~~eighteen (18)~~ months in advance. Should VENDOR discontinue the Software before the expiry of the Life Span, VENDOR shall~~, without prejudice to BUYER’s other rights and remedies under the Agreement, migrate the Software to a new, functionally equivalent technological basis at no cost to BUYER~~ be offered a succeeding or different Software by VENDOR at special commercial conditions, to the extent such Software is available.

VENDOR further undertakes, during a period of no less than ~~five (5)~~ one (1) year~~s~~ from the end of the Life Span, to generally make available maintenance and support services for the Software at ~~competitive~~ generally available terms ~~and to hold available necessary technical and personnel resources with respect thereto~~.

VENDOR shall be entitled to change the Life Span and time frames defined in this section in its sole discretion if this is required based on business needs or technical risk for customers.

This clause shall survive termination of the Agreement ~~and~~ but shall ~~apply irrespective of the terms of any~~ always be subject to the uninterrupted effectiveness of a maintenance and support agreement regarding the Software during the entire Life Span.

3.2 Release Support Policies

Software is in constant evolution. In order to exploit its resources in an efficient and commercially viable manner, a software vendor is interested in maintaining only the most recent software releases, thereby providing an incentive to customers to migrate to the new release. Users on the other hand wish to retain the possibility to skip new releases and keep the currently installed release, especially in the event that a new release does not add value to their business or even discontinues a previously available feature. As a minimum requirement, they want to be sure that the currently installed release is maintained for a certain minimum period and does not need to be replaced shortly after its deployment.

In a VENDOR's standard contract template, the software release policy may be phrased as follows:

> *Maintenance for Software will be available for each Release until the date a second subsequent Release becomes generally available or until the date separately notified by VENDOR.*

The BUYER's main concern will be that the software release cycles will not force it to install new releases within short intervals. Hence, it will require a minimum period during which an installed software release remains subject to VENDOR's maintenance obligation. Further, end of maintenance of the installed software release should not be coterminous to the general availability of a subsequent release, because the time lag in having such subsequent release installed and integrated must be taken into account. This leads to the following mark-up by BUYER:

> *Maintenance for Software will be available for each Release at least until six (6) months after the date a second subsequent Release becomes generally available or ~~until the date separately notified by VENDOR~~ for a period of two (2) years following general availability of each Release, whichever is longer. End of maintenance of a Release must be announced in writing at least six (6) months in advance.*

3.3 Compatibility and Capacity Requirements

The deployment of new releases may have repercussions on the compatibility of the product with customer's surrounding IT environment and may entail additional capacity requirements to operate the software.

The following clause provided by BUYER may be a typical starting point for negotiations:

> *VENDOR undertakes that the Software remains compatible with BUYER’s peripheral systems in terms of architecture, specifications and provided capacity during the entire Life Span, without detrimental effect to the then current functionality or other performance parameters of the Software.*
>
> *Should VENDOR fail to comply with the requirements set forth in this clause, VENDOR shall, without prejudice to BUYER’s other rights and remedies under the Agreement, upgrade BUYER's peripheral systems without loss of functionality or performance and at no cost to BUYER in such way that the committed compatibility requirements can be met.*

The VENDOR confronted with such a requirement will only be willing to negotiate if it is a customized software solution and VENDOR has had the possibility to assess the BUYER’s peripheral system before the Effective Date. Vendor will further consider such requirement as inacceptable in case a new software release comes with additional value-adding features or enhanced performance. It will also try to carve-out from its commitment any dynamics that are beyond its control:

> *VENDOR undertakes that the Software remains compatible with BUYER’s peripheral systems in terms of architecture, specifications and provided capacity as existing at the Effective Date during the entire Life Span, without detrimental effect to the ~~then current~~ functionality or other performance parameters of the Software as at the Effective Date.*
>
> *In case BUYER changes the specifications of its peripheral systems, or in the event that enhanced functionality of the Software requires modifications to be made to BUYER's peripheral systems or influences the performance parameters of the Software, such new specifications of BUYER’s peripheral systems, new Software functionality and new performance parameters will serve as new point of reference for the remainder of the Life Span.*
>
> *Should VENDOR fail to comply with the requirements set forth in this clause, VENDOR shall~~, without preju-~~*

> *~~dice to BUYER's~~ at its option and to the exclusion of any other rights or remedies ~~under the Agreement~~ available to BUYER, either (i) upgrade BUYER's peripheral systems without loss of functionality or performance and at no cost to BUYER in such way that the committed compatibility requirements can be met, or (ii) re-deploy and continue supporting the previous compatible Release.*

The customer in its further reaction will be reluctant to upgrade its surrounding systems if it is forced to deploy new features that it does neither desire nor actually use. So it is well advised to insist on a qualification that the enhanced functionality of the software will only count as a defense if such functionality is ordered and actually used.

4. Negotiating Investment Protection Clauses in XaaS – What is Different?

4.1 Facilitating Change vs. Maintaining Status Quo

XaaS stands for "Everything as a Service" and may entail, among the most frequently used business models, Software, Platform or Infrastructure as a Service.

Contrary to the traditional procurement of ICT infrastructure, the customer does not make an investment in a *product*, but orders a *service* that is operated by the provider on an external platform. The only infrastructure required by the user is a browser or other virtualized desktop installed on the client in order to receive the service. The provided service application as such is not reliant on a customer-run operating system; only where interoperability with other customer applications is needed, integration issues may remain.

SaaS respectively XaaS offerings mean no or only small initial investment costs and generally also significantly lower ongoing operating expenditure. At the same time customers benefit from high security standards, feature variety and capacity reserves, which an on-premise infrastructure in most cases could not offer at similar price level. The high flexibility, usability and quality of such services at a minimum investment can be seen as an investment protection in itself. If the customer is unhappy with the provided solution, the financial hurdles for a provider swap are almost negligible, at least compared to the ensuing swap costs in traditional ICT sourcing.

However, these apparent advantages do not come without downsides:

- First, the customer is deprived of any flexibility in skipping releases. In a multi-tenant environment, there is just one service provided in the current release to a multitude of users. In case a core feature is discontinued in a subsequent release, there is no option in reverting to the previous scope.

- Second, the customer is giving away control over its business data, that are hosted and processed by the service provider. If the service is discontinued, business continuity is at stake immediately. Contrary to locally installed software, which keeps on running for a while if the licensor is facing insolvency or otherwise closes its business, the operation of the outsourced platform may be stopped overnight. While the customer may reverse-engineer locally installed software or have the source code released under a possible escrow if vendor fails in remedying defects, there is no physical access to the software code in a XaaS setup.

With these preliminary observations in mind, there is a shift of perspective from which investment protection is looked at: In traditional ICT sourcing, the focus is on the product, i.e. the deliverables of the vendor the investment in which should pay-out over the long term. In XaaS, the focus is on the customer data and business continuity. To cope with the downsides of XaaS, rather than implementing a scheme that helps in retaining the current solution for as long as possible, ways should be explored to facilitate and accelerate migration of hosted data and applications.

Given these conceptual differences, it makes little use to apply the same concepts: negotiating investment protection in XaaS *is* different. There may be minimum initial contract terms, termination and notification periods to observe before a service provider is entitled to discontinue or change a service, which reminds of the previous discussion on product lifecycle and release support. However, these seem to be ancillary points. The key questions are:

1. Is the service provider obliged to release hosted customer data, and if so, in what format, at what cost, and in what timeframe?
2. How does the customer get access to data if the service provider is incapable or reluctant of releasing the data?
3. How does the customer ensure business continuity if the data is released, but the provided service functionality is unavailable?

These questions show that the access to data and data portability is the core of ensuring business continuity in a XaaS world. In case the service provider does no longer provide the services for any reason and the customer can no longer access and use its business critical data, business continuity is seriously at risk. The release of data is equally relevant if the customer wants to change to a different solution, be it by switching back to traditional self-operated applications or by migrating to another XaaS provider. If this possibility to change the solution does technically not exist or only at unreasonable cost, the customer would end up in a so-called "Vendor-Lock-in".[2] It is not surprising that this aspect has been also identified by the lawmakers as a potential weakness of Cloud solutions. The EU Commission's Cloud Initiative highlights data portability, interoperability and reversibility as key points.[3]

However, even if data portability was properly safeguarded, this may result in a halfway achievement only. Possession of data does not endow the customer with a seamless continuation of the business critical service application functionality on the basis of the same or compatible business processes, logics and workflows. Taking business continuity seriously would also involve reflections about application portability on top of data portability – a concept not yet widely advocated (if at all) in XaaS contract negotiations.[4]

[2] ROTH-NEUSCHILD: Cloud Way out. Exit-Strategien bei der Nutzung von Cloud Services, ITRB 2013, 213–217, 215.

[3] European Commission: European Cloud Computing Strategy, <https://ec.europa.eu/digital-agenda/node/10565>.

[4] HON / MILLARD / WALDEN: Negotiating Cloud Contracts: Looking at Clouds from Both Sides Now, in: 16 Stan. Tech. L. Rev. 79 (2012), 79–129, 116.

4.2 Statutory and Contractual Remedies

After having outlined the criticalities of investment protection in XaaS settings, the available legal tools to support the objective of safeguarding business continuity will be examined. Without asserting completeness, access to business data may be accomplished either by way of invoking statutory data access rights or by way of reliance on contractual remedies.

Even if data privacy legislations in many jurisdictions endow the data subect with the right to get its personal data returned or deleted, it is not recommended to rely on such statutory provisions for various reasons:

i. Data privacy laws in Europe do – save for some exceptions – only address the right of accessing personal data of private individuals and do not embrace data pertaining to legal persons;

ii. the right of accessing own personal data does not extend to any third party business data that may be hosted by the provider;

iii. .the information and business relevant data stored by XaaS customers is not limited to personal data (e.g., statistical market analysis data);

iv. data access rules under privacy laws are not designed to ensure business continuity – the process takes too long.

In view of the above, invoking data access rights under statutory privacy legislation is not a practical solution to the problem.

It is submitted that the service provider's obligation of retaining and releasing hosted customer data is an implied term in any XaaS contract and sometimes even laid-down as a principle in the applicable national civil codes.[5] However, given that XaaS contractual practices have not yet matured, the parties are advised to invest considerable time and effort in negotiation data portability issues. In this respect, the following checklist may be helpful:[6]

i. Scope of data: it is important for the customer to not only get its original data back as it was uploaded or entered by the customer into the service provider's application, but also have ownership of data that was produced in the XaaS system by processing customer's input data in the course of the provision of the services.

[5] Cf. Article 400 para. 1 of the Swiss Code of Obligations: *"The agent is obliged at the principal's request, which may be made at any time, to give an account of his agency activities* ***and to return anything received for whatever reason as a result of such activities****"* (emphasis added), or sec. 539 para. 2 of the German Civil Code (to the extent XaaS is deemed as lease): *"The lessee is entitled to withdraw a contribution that he has provided the leased property with."*

[6] Cf. also HON / MILLARD / WALDEN (footnote 4), 117.

ii. Data formats: it is recommendable to clearly define contractually that the service provider has an obligation to provide such data to customer using common interface standards and data formats.

iii. Timeframe: the time period during which data must be retained after contract termination should be defined. The right to have the data released should further be capable of being asserted in anticipation of termination, not only at the time the contract is expired. Especially for long-term contracts the customer may also want to consider requesting a data portability right on an on-going basis and not only in the event of termination of the agreement.

iv. Assistance and fees: scope and fees of service provider's termination assistance, e.g. for possible data format conversions, should be addressed in the contract.

v. Right of retention: whether the service provider has a right to suspend the services in case of non-payment of service fees, or even a right of lien in the data upon termination, is another matter that may be subject to intensive discussions between the customer and the service provider, but should in any case be regulated in a contract.

The risk that a service provider is no longer able to provide the services, in particular due to insolvency, is more difficult to address. Whereas for normal software purchases source code escrow may be an option (although of questionable practical relevance and not supported by many software vendors), this solution seems more difficult in a SaaS environment. Interestingly there are SaaS escrow solutions being offered in the market, purporting to hold available a fully-fledged redundant application that may substitute the principal service platform within an instance. To our knowledge, such offerings have not yet stood their reality check. It will have to be seen if XaaS service providers are prepared to allow such escrow service providers to operate the XaaS provider's software and systems to make this a viable solution.

The most obvious risk mitigation strategy is of course, like for the selection of every other business critical service provider, to chose of a trustworthy and stable provider. Technically hybrid solutions, where customer stores its data in the cloud and simultaneously on an on-premise infrastructure, or regular data backups with another service provider or on own infrastructure may be another possibility.

4.3 Conclusion and Outlook

Whereas traditional software that is installed on an own infrastructure will most likely remain the preference for many cases, Cloud and XaaS offerings are a fast growing market and attractive alternative for many customers from a technical and commercial perspective. Like every commercial construct or technology there is no risk free “one-size-fits-all” solution that is “right” for every customer. And even if there are unsolved legal questions with regards to XaaS (as for traditional ICT projects), lawyers should help their clients to identify, understand and mitigate these risks and enable them to make the right business decision for the solution that meets their requirements best – and not just see problems in new and innovative business models only because they are more comfortable with the old world. The conclusions drawn herein, derived from the insight that investment protection in XaaS means facilitation of change rather than preservation of an installed base, may serve as an impulse in that direction. Innovation in technology and business is the basis for innovation by lawyers – and should therefore be seen as a chance and not as a threat.

Michael Isler and Oliver Kirchner
Zurich and Schaffhausen, 30 September 2013