Managing Hosted Unified Communication Services in the Cloud: Challenges and Opportunities

***Abstract*—In this paper we take a high level look at the advantages and challenges of migrating Unified Communication (UC) services to a cloud environment. While the advantages are clear in term of cost savings, innovation and ease of use, the challenges are often underestimated. This paper outlines the different alternatives available to deploy, license and scale UC services in the cloud.**

I. Introduction

The transition from on-premise telephony systems to hosted cloud-based Unified Communications Services is probably the most disruptive innovation to impact the telephony and communication research and market in recent years [1]. This transition has been largely fuelled by the same factors that have made cloud-based hosted applications so successful, such as:

- Reduced CAPEX: To ensure that a telephony server is sufficiently dimensioned to support current needs and future expansions, enterprises usually procure an over-dimensioned system., which increases the immediate capital expenditure beyond actual need.. By deploying a cloud-based system the costs of hardware and software ownership are substituted by monthly subscription fees based on the exact real time usage of the enterprise.
- Reduced OPEX: On-premise systems must be supported and administered. This results in annual costs such as supplier Support Level Agreements (SLAs) and/or local staff to administer the system. With a cloud based system these costs are reduced as support and maintenance are provided by the cloud-based service provider as part of the managed subscription.
- Innovation: For a long period of time communication was basically just a simple phone call with some supplementary services. With the advent of unified communication users are expecting now video, conferencing, messaging and integration of fixed and mobile devices. This has led to faster development cycles and shorter update intervals. With the services centrally managed the application provider is responsible for introducing the new features and keeping the system up-to-date.
- Mobility: With the telephony located in the cloud, it is no longer relevant if

the user is located inside the enterprise or outside of it, using his mobile phone, laptop or fixed phone.

- Security: A Telephony system located at an enterprise will have to be secured just like any other local service. This requires dedicated VoIP security and firewall components and extensive monitoring and alarming solutions to prevent denial of service attacks and detect and avoid fraud. VoIP security is, however, still a developing topic [1]. Hence, effective defence will require a high level of knowledge from the enterprise's employees and complex defence solutions. By relying on a unified communication solution that is deployed as as Software as a Service (SaaS) solution an enterprise effectively outsources the responsibility to the SaaS provider which will have the expertise and components to provide the needed protection. By offering UC services to multiple enterprises, the SaaS provider can have a broader view of possible attacks and can utilise the knowledge and experience gained from one deployment to protect multiple enterprises.

However, for all the advantages that UC cloud deployments bring, there are also some technical challenges that vendors and operators need to address to ensure the manageability and security of the solution.

Beside the complexity of actually migrating a software to a cloud environment [2] one needs to consider the introduced management overhead. In the following sections this paper will address the general possibilities for deploying a unified communication service in a cloud environment and then take a closer look at some of the issues that have to be addressed in a cloud environment, namely, manageability, scalability and software licensing.

II. Deploying Hosted Unified Communication Systems in the Cloud

Typically, Unified Communication Systems provide the following functionalities [3]:

- Media Processors (MP): Handle the media, e.g., RTP traffic [4] and provide services such as transcoding media mixing, announcements and IVR services.
- Signalling Processors (SP): Handle call signalling, user registrations, e.g. SIP signalling [5].
- Customer Management (CM): Enable the customers to configure the system and customise its behaviour in accordance to their requirements.
- Monitoring: Provides an overview of the system's performance and a way to trace operational and security issues.
- Database (DB): Unified communication solutions will usually have two types of databases. One that stores the customer's operational data such as configuration and accounting data and another for short lived data such as session state data, user location and status information.
- Security: Security is usually provided in the form of Session Border Controllers (SBC) which protect unified communication environments from attacks as well as manage SLAs and detect malicious behavior [6]
- System Management (SM): responsible for the management of the entire system and the coordination between the different components of the solution.

While all Unified Communications solutions will consist of these functional components, one can distinguish between two

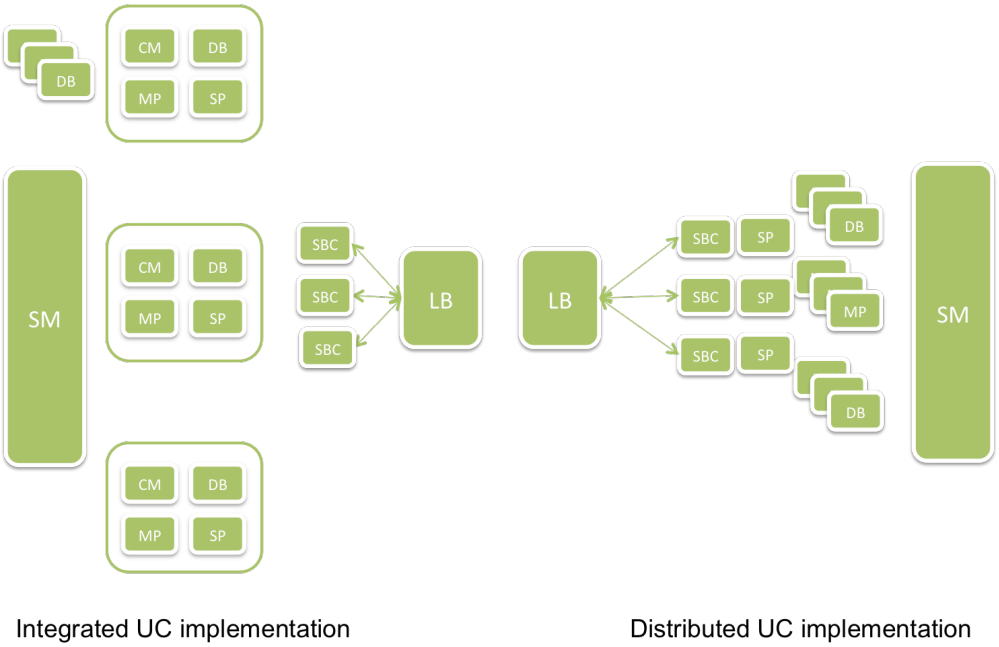

Fig. 1. UC implementation models in the cloud

possible implementation architectures: integrated and distributed, see Fig. 1.

In an integrated UC solution, all the functional components are implemented as part of the same process or as separate processes that communicate with each other over an Inter-Process Communication (IPC) method. Usually, this type of solution is scaled by adding a load balancer in front of a cluster of servers that run the UC solution. As the session border controllers are often provided by a different vendor than the UC vendor, the SBCs are often deployed in their own cluster.

With a distributed implementation, each functional component might run on its own server. A popular approach is to have servers dedicated for the customer management component, others for the signalling processing and have the media processors running on dedicated hardware. This approach has the advantage that it is possible to scale the different components independent of each other and possibly use dedicated hardware for certain tasks such as video processing. In order to scale such solutions load balancers are needed to distribute the signalling traffic among the available signalling processors. The signalling processors then decide how to distribute the media traffic.

A hosted communication system differs from an on-premise UC solution in two points: it is (i)mutli-tenant capable and (ii) is run by an operator on behalf of the customer. The customer, usually an enterprise, can access the solution and configure its own services and required behaviour. The operator of the hosted UC service runs the solution over a set of servers either managed directly by the operator or located in a data centre.

Unified communication systems deployed by an enterprise or as a hosted system fulfil

their performance requirements by deploying sufficient hardware. Such systems are scaled out by adding more hardware. In a cloud environment, unified communication solutions use multiple virtual machines and are scaled out and down by adding or removing virtual machines.

In general one can distinguish between two possible approaches for deploying a hosted unified communication service in the cloud: system oriented and customer oriented, see Fig. 2.

In a system-oriented UC cloud deployment a hosted UC solution is deployed over virtual machines in a cloud environment. All customers share the available resources. With a distributed UC implementation, the operator deploys a number of virtual machines dedicated to each functionality. That is a cluster for singling processing, another for the media processing and so on. Compared to a traditional hosted UC service, a system oriented deployment can be scaled out and down by adding or removing virtual machines whereas a traditional hosted UC solution is scaled out by adding more hardware.

In a customer-oriented deployment, each customer is allocated a virtual infrastructure dedicated for processing the customer's traffic. This infrastructure consists of virtual machines running the signalling and media processors as well as session border controllers for security. Further, the customer relies on the operators' infrastructure for customer management, billing, database as well as life cycle management -e.g., software updates and patches.

By allocating a virtual infrastructure per customer, the operator can achieve the following:

- Separate customer traffic and hence avoid the situation in which one customer overloads the processing resources and reduces the quality of service (QoS) for other customers
- Granular per customer configuration: Enable a per customer configuration that is customised to the exact needs of the customer without adding too much complexity to the overall service's configuration scheme
- Limit effects of failures: In case some bug or call scenario causes a system failure then this failure would be limited to the customer that has encountered this bug or call flow. Other customers will not be affected as long as the same error cause was not encountered by them as well.
- Limit effects of security attacks: In case a certain customer is attacked then in the case of a traditional hosing solution in which all customers share the same infrastructure of the hosting provider all customers might be affected. By dedicating a separate virtual infrastructure for each customer the denial of service attacks or security breaches would only have effects on the targeted customer.

III. Scaling Cloud-based Unified Communication Systems

One of the basic merits of a SaaS offering is that the amount of resources needed for the service can be adapted to the exact needs of the service by adding or removing virtual machines. So when a cloud based web service sees a surge in the number of incoming requests, the service can be scaled out by adding additional servers. When the load goes down then servers can be turned off and used for other services.

This is not as simple with a unified communication system due to what is known as „session stickiness“ [7]. When the overall load of a UC system goes down and the

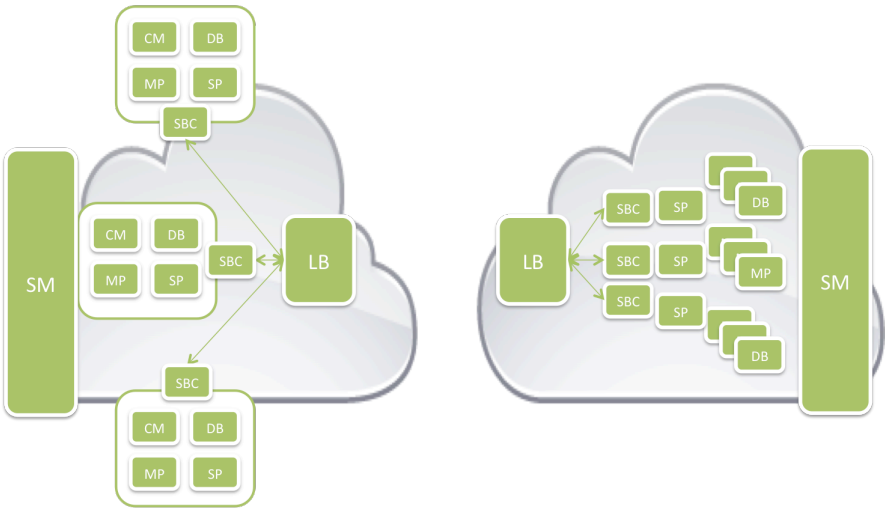

Fig. 2. UC Deployment models in the cloud

number of used servers could be reduced it is not possible to just turn a server down while active calls are being processed. Further, when the system is scaled out it must be ensured that all messages that belong to a call will still be assigned to the server that is currently processing that call.

Using SIP-aware load balancers that support session stickiness, that is a load balancer that keeps track of which server is responsible for which SIP session, is the usual solution for the case of scaling a service out. When a new server is to be added to the cluster of UC servers, the load balancer will include the new server in its load balancing considerations but route only new calls to that server.

Scaling down is a bit more complicated. In the typical web-based cloud application, a session consists of multiple TCP connections. These connections don't have to be processed by the same server. When a server is removed from the cluster the user's session will continue on another server. In the case of unified communication a session consists of signalling messages and media packets that are processed by a server that keeps state information about that session. Therefore, servers can only be taken out of a cluster once all sessions on a server have been terminated. Depending on the chattiness of the involved users the duration of a call can be anything from a couple of minutes to a couple of hours. Hence, it is not always feasible to wait for the sessions to terminate before shutting a server down as cloud infrastructure providers often charge for using the servers by the minute. Some solution options are:

- Transfer the call state to another server: Call state information could be saved in a central database accessible by all servers in the cluster [8]. All traffic arrives into the cluster through a session aware load balancer (LB) and leaves the cluster through the same or another session aware load balancer see Fig. 3. When a server in the cluster is

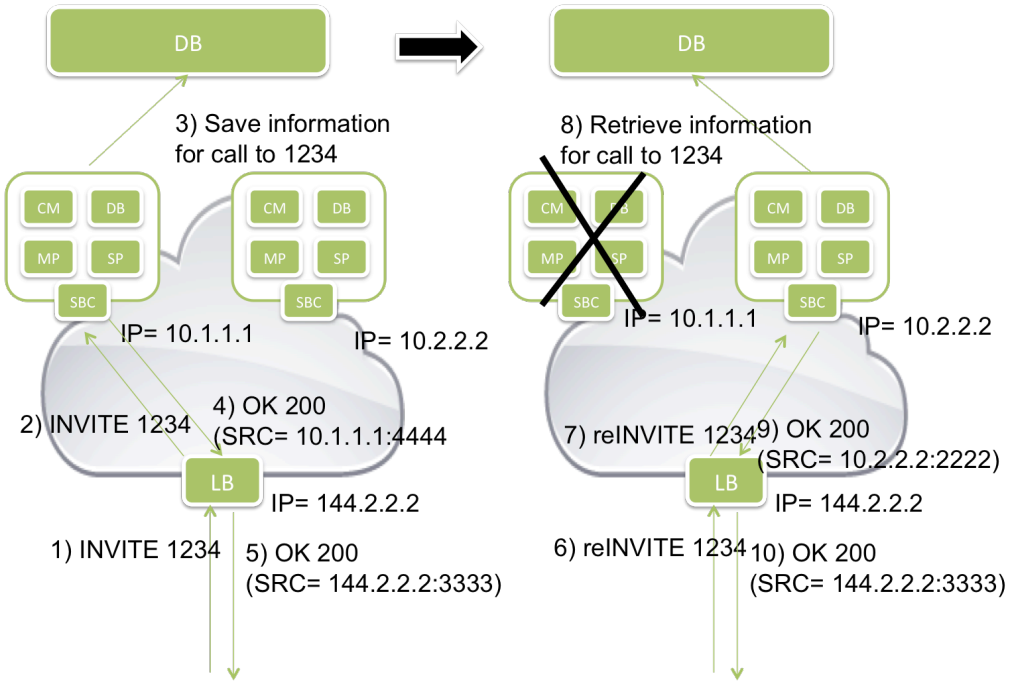

Fig. 3. UC caling-down with Call state transfer

to be removed then the ingress LB is notified and will send all traffic relating to the sessions that were active on that server to other servers. When a UC server receives data for a session which it was not processing it will retrieve the session data from the central location. Changes in the IP addresses of SIP and RTP messages might cause issues at other SIP instances beyond the UC servers, e.g., end clients, SIP servers of other operators or enterprises. Therefore, any SIP and RTP traffic leaving the UC cluster will have to go through an egress load balancer. The egress LB will also be notified of the session shift and will correlate the SIP and RTP messages to an already established session regardless of the used source IP. Thereby, scaling down a service can be accomplished by a tight coordination between ingress and egress session aware load balancers as well as keeping session data in a central database. However, this approach requires support from the UC instances themselves which need to be able to save and retrieve state information from a central database and use it to process the calls..

- Call termination: Here, the orchestrator responsible for scaling the servers down and out notifies a session aware load balancer about the server designated for termination. The LB then stops sending new session establishment requests to that server. Further, after a certain period of time, the LB will terminate all active calls to that server. While this will cause some calls to be dropped, they will be terminated in an orderly manner and the users can

restart the calls. A major advantage is that the UC servers don't have to be aware of the orchestration process or support handling of mid-call transfers which is not usually supported by a current UC software. The needed logic is contained in the load balancers and orchestrator.

IV. Managing Cloud based Unified communication systems

Cloud platforms have their own routing, networking and load balancing technologies. A vendor adapting its UC solution to a certain cloud technology will have to accommodate these technologies and ensure the added complexity is hidden from the operator of the UC service. From the point of view of the operator, the major difference to a hosted solution will be in the scaling strategy of the service. In the case of the system-based deployment, the operator will define the proper scaling strategy so as it fits its own needs. For the case of the customer-based deployment, see Sec. II, additional features will be needed as part of the service management:

- Per customer scaling strategies: With different customers having different expectations in terms of performance, traffic and costs the service provider will have to support different scaling strategies for different customers. So beside the ability to configure the service itself the customers will have the possibility to configure their auto-scaling plans as well.
- Per customer infrastructure management: Depending on the customer requirements and configuration the virtual infrastructure dedicated to them can be different. The number and type of components used by a customer that is utilising conferencing services to a great extent will differ from the infrastructure required by a customer using only audio and video calls. So the customer can design and plan the virtual infrastructure by specifying the needed service and the operator will have be able to deploy the required infrastructure. This requires the services to include a mapping of the customers' specifications to the components to be deployed.

V. Licensing Cloud based Unified communication systems

When considering the aspects of licensing of UC services in the cloud one needs to consider two aspects: How are the services offered to the subscribers and how are licenses enforced by the vendor of the unified communications platform?

The usual approach for offering cloud services is based on a monthly subscription model. For a fixed price the customer receives a service with certain features and capacities.

When it comes to licensing unified communication software it gets a bit more difficult. The usual approach for licensing UC systems is for the operator to buy from the vendor a number of licenses that support the expected worst case demand and cover certain services, e.g., audio calls, conferencing, IVR, etc. These licenses are bound to certain servers and hardware platforms. While this approach has been adequate for on-premise deployments or even for hosted UC solutions it is not adequate in a cloud environment.

At times in which the operator of the UC solutions buys computing resources such as CPU, memory and bandwidth in the cloud based on a usage model and is accounted on a minute or hour basis, requiring the operator to pay for the worst case scenario de-

mand might seem outdated. Further, with the operator of the UC service deploying virtual machines that are scaled down and out depending on the traffic load the basic assumption of traditional licensing solutions, namely to allocate certain licenses to certain servers that are well known in advance, no longer holds. The vendor of the UC solution will have to consider the following limitations:

- The number of virtual machines running the UC service changes over time
- The IP and mac addresses which are usually used to identify the servers are not known in advance
- In the spirit of cloud deployment, it should be possible to offer the operator the option to acquire licenses on demand and not require the operator to buy sufficient licenses to cover the worst case traffic

The simplest approach to support such usage scenario would be to have the vendor of the UC solution to run a licensing server. Whenever a new virtual machine is started by the operator, this new instance contacts the licensing server and requests a bunch of licenses. If traffic increases then the VM can request additional licenses. These licenses would be time limited and the VM needs to renew the lease regularly. In case the VM is terminated then the lease can be either terminated explicitly or it times out. This approach enables the vendor to collect exact usage information, charge on time base and provide the operator with accurate bills. For the operator this has the advantage that it can scale the service in real time and only pay for the licenses used. This simple this approach will usually not be accepted by larger operators of UC services. Having to rely on an external entity means that in case this entity is not reachable the entire UC service can fail. Further, having a vendor collecting exact information about the usage model of the operator raises privacy concerns and is usually not acceptable by the operator. A variation of this model is to have the licensing server be part of the UC solution itself. That is, the licensing server is run by the operator of the UC service. In this case, the UC server needs to contact the vendor on a regular basis to report about the used licenses by the operator. The licensing server itself would have a time limited license that needs to be regularly renewed by contacting the vendor. This could be done in a time uncritical manner by setting the interval to a month for example and renewing the license already a couple of days before it expires.

Another approach for licensing is to bind the licensing process with licensing and billing components of the cloud platform itself. In this case the vendor packages its own software with virtual machines of different sizes (e.g., different CPU, memory and bandwidth capacities) and offers them through the cloud operator, e.g., Amazon AWS market place [9], and indicates the pricing model, per time, bandwidth or request for example. The UC operator can then buy the right to use the software through the cloud operator and is billed by the cloud operator. While this approach relieves the vendor from having to develop a dedicated licensing technology there is a monetary penalty as cloud operators charge a percentage of the generated revenues for this service.

Hence, the challenge here is to develop a licensing model that provides the flexibility needed by a cloud deployment but still considers the security and high availability needs of the UC operator.

VI. Summary

Migrating unified communication services to the cloud promises cost savings and faster innovation cycles. However, the migration process poses also various technical challenges. In this paper we discussed some of these challenges in the context of scalability, manageability and licensing. Especially in the context of scalability new components and updates to both the cloud platform and the UC software are needed to ensure safe scaling down and out. Also in the context of licensing new models are needed that accommodate the dynamic nature of cloud environments and the changed expectations of the customers.

References

[1] T. Erl, R. Puttini, and Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2013.

[2] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 142–157, 2013.

[3] W. Flanagan, *VoIP and Unified Communications: Internet Telephony and the Future Voice Network*. Wiley, 2012. [Online]. Available: https://books.google.de/books?id=pgsc4hyMrgUC

[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), Internet Engineering Task Force, June 2002, updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[6] B. Penfield, J. Hautakorpi, M. Bhatia, A. Hawrylyshen, and G. Camarillo, "Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments," RFC 5853, Apr. 2010. [Online]. Available: https://rfc-editor.org/rfc/rfc5853.txt

[7] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 45–52, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1925861.1925869

[8] J. Sherry, P. X. Gao, S. Basu, A. Panda, A. Krishnamurthy, C. Maciocco, M. Manesh, J. a. Martins, S. Ratnasamy, L. Rizzo, and S. Shenker, "Rollback-recovery for middleboxes," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 227–240, Aug. 2015. [Online]. Available: http://doi.acm.org/10.1145/2829988.2787501

[9] Amazon, *Launching an AWS Marketplace Instance*, 2017.