Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2012 Proceedings | Proceedings

The unperceived boon and bane of Cloud Computing: End-user Computing vs. Integration

Jochen Hetzenecker
Information Systems III, University of Erlangen-Nuremberg, Nuremberg, Germany., jochen.hetzenecker@wiso.uni-erlangen.de

Sebastian Sprenger
Information Systems III, University of Erlangen-Nuremberg, Nuremberg, Germany., sebastian.sprenger@wiso.uni-erlangen.de

Sebastian Kammerer
Information Systems III, University of Erlangen-Nuremberg, Nuremberg, Germany., sebastian.kammerer@wiso.uni-erlangen.de

Michael Amberg
Information Systems III, University of Erlangen-Nuremberg, Nuremberg, Germany., michael.amberg@wiso.uni-erlangen.de

The unperceived boon and bane of Cloud Computing: End-user Computing vs. Integration

Jochen Hetzenecker
University of Erlangen-Nuremberg
Jochen.Hetzenecker@wiso.uni-erlangen.de

Sebastian Sprenger
University of Erlangen-Nuremberg
Sebastian.Sprenger@wiso.uni-erlangen.de

Sebastian Kammerer
University of Erlangen-Nuremberg
Sebastian.Kammerer@wiso.uni-erlangen.de

Michael Amberg
University of Erlangen-Nuremberg
Michael.Amberg@wiso.uni-erlangen.de

ABSTRACT

Cloud Computing is predicted to bring the next revolution in IT. Easy availability and the intuitive usability enable IT-professionals but also less IT-skilled people deploying IT-services. This encourages End-user Computing, where the end-user is able to solve business issues by designing the service as needed. But with end-users establishing such services, problems of integration occur due lack of IT-knowledge and insufficient skills of conceptual IT-development. By conducting a case study following the design science approach, we pointed out that with establishing cloud services the degree of End-user Computing is increased and moreover influenced by the services applied. Furthermore, we constrained that the independency from an IT-department ends up with the integration of end-user designed services in the existing IT-infrastructure. In order to solve this issue, we manually integrated IT-systems by the use of the Enterprise Application Integration approach.

Keywords

Cloud Computing, SaaS, IaaS, End-user Computing, Enterprise Architecture Integration, Design Science

INTRODUCTION

Since 2007, Cloud Computing is steadily gaining popularity by experts from science and practice (Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.; Katz, R.; Konwinski, A., 2010). The National Institute of Standards and Technology (NIST) describes Cloud Computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." Thereby, technological and financial benefits are acknowledged by many studies: Gartner places Cloud Computing at the "peak of inflated expectations" on its Hype Cycle during the last three years and devotes Cloud Computing an extraordinary Hype Cycle (Gartner Inc., 2011). Furthermore, a six fold expenditure rising from 2010 to 2015 is predicted by the Experton Group (Experton, 2010). In addition, Deutsche Bank Research forecasts a worldwide market volume of 71 billion dollars for Cloud Computing in the next five years (DB Research, 2012). Moreover, they expect an increase of usage to more than 25% of the companies in 2012 (DB Research, 2012). This development is among others enabled by the easy usage of Cloud Computing, especially Software as a Service. While installing an on-premise IT-system requires IT-skilled experts, Cloud Computing enables the installation and configuration of IT-systems even by less IT-skilled employees. In some cases, it is even possible to shift conceptual tasks to the end-user (Hetzenecker, Wiener, Amberg, 2011). This may result in higher user acceptance and less IT-dependencies, but may also lead to integration problems due to supplementary isolated applications. With the help of a case study in a hotel following the design science approach, we examined the following research questions:

- Is End-user Computing encouraged by the use of Cloud Computing?
- How can an End-user designed Cloud Computing solution be integrated in an existing IT-environment?

THEORETICAL BACKGROUND

In the following chapter, a common theoretical foundation is established. Therefore, Cloud Computing will be introduced firstly, before we describe End-user Computing and the design science approach.

Cloud Computing

Although Cloud Computing has been discussed for a few years in science and practice, no general definition of the term has been established so far. But recently, the definition of National Institute of Standards and Technology (NIST) seems to find increasing attention and acceptance. NIST describes Cloud Computing as "[…] a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (National Institute of Standards and Technology, 2011). In addition, Cloud Computing is defined by means of its main characteristics "On-demand self-service", "Broad network access", "Resource pooling", "Rapid elasticity", and "Measured service" (National Institute of Standards and Technology, 2011). Furthermore, the "Pay as you go" manner is also considered to be an essential characteristic of Cloud Computing (Armbrust et al., 2010).

Architecture

Cloud Computing can be split up in three architectures. Infrastructure as a Service (IaaS) describes elements of Cloud Computing, which are closest to hardware issues. It addresses the deployment of computational power and storage (Weinhardt, C.; Anandasivam, A.; Blau, B.; Borissov, N.; Meinl, T.; Michalk, W., 2009). The next layer is called Platform as a Service (PaaS). It provides development platforms which may base on IaaS in order to develop software applications. PaaS enables the development, testing, deployment, hosting, and maintaining applications in a virtual environment. Often, the whole software lifecycle of an application can be handled with PaaS services (Weinhardt et al., 2009). The most common layer is called Software as a Service (SaaS). SaaS applications often operate on the hardware of an underlying IaaS layer and describe the provision of applications as a service. Moreover, SaaS can be split up in two subparts; on the one hand into entire software packages like CRM or ERP systems, which are delivered as a service, and on the other hand into on-demand web services such as map or stock price information (Weinhardt et al., 2009).

Deployment

Generally, Cloud Computing can be divided into three ways of deploying cloud services. Public Cloud Computing describes the supply of cloud services which are accessible for public use. But the fact that everyone is allowed to use public cloud services may raise security or data privacy issues. A solution might be Private Cloud Computing. Here, the cloud service is offered especially for one organization, which might comprise different customers, i.e. business units. Although this scenario may fix problems of data privacy and security, benefits such as flexibility and cost reduction might be reduced. Hybrid Clouds are defined as a composition of both Public and Private Cloud Computing characteristics. They can help increase user security by storing less critical data in a Public Cloud, and personnel or critical in the Private Cloud. Furthermore, NIST suggests the community Cloud Computing approach, which in our opinion is a subcategory of the previous mentioned. (National Institute of Standards and Technology, 2011)

End-user Computing

In former times, End-user Computing (EUC) was defined as individuals who interact directly with the computer (Rockart, J. F.; Flannery, L., 1983).

Further, the end-user facility task group of the CODASYL Systems Committee provides a more detailed definition of End-user Computing. They distinguish between "direct" end-users who actually use hard- and software devices, "intermediate" end-users who specify business information requirements, and finally "indirect" end-users who use computers only through third parties (Lefkovits, Henry C., 1978).

In contrast, Davis and Olson point out two different user roles concerning End-user Computing:

- A primary user, who makes decisions on the base of the system's output.
- A secondary user, who interacts with the software, but does not use the output directly.

They conclude that in End-user Computing the two described roles are combined. (Davis, G.B.; Olson, M.H., 1985)

Until now, the application of EUC was constrained by technological restrictions. Administrators and the IT-department were responsible for adoptions and extensions of IT-systems. With the new paradigm of Cloud Computing, where services are loosely coupled via service-oriented architecture (SOA) (Vouk, M. A., 2008), end-users are enabled to design own solutions by combining services without the need of programming knowledge (Hetzenecker et al., 2011).

Design Science

The fundamental goal of the Design Science approach is "that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artifact" (Hevner, A. R.; March, S. T.; Park, J.; Ram, S., 2004).

According to the Design Research methodology developed by Vaishnavi and Kuechler, the general process of designing an information system is made up of five steps with each producing a definite output. First, the awareness of a real-world problem must be established. After understanding the problem, a suggestion for solving it must be developed. In the development phase, the artifact is constructed and implemented. This phase iteratively alternates with the evaluation of the artifact until a conclusion can be stated. This process can circulate repeatedly. The five steps of the process will therefore serve as the structural reference for the main part of this paper to ensure a comprehensible outline. The general focus is to provide a definitive result which corresponds with the respective outputs defined by this approach in the end (Vaishnavi, V.; Kuechler, B., 2011).

The Design Science approach states seven guidelines which must be fulfilled. Hence, an artifact that solves a relevant problem must be developed. The artifacts efficiency must be evaluated and should provide contributions on the relevant research areas. The development must follow an iterative search process and base on rigor research methods. The result must be communicated in scientific literature (Hevner et al., 2004).

CASE

In order to examine the research questions, we conducted a case study with a hotel which is introduced in the scenario section. Afterwards, we show two iterations of the design science process. First, we introduce the migration from an exchange server to Google Mail without professional IT consulting; the second iteration illustrates the implementation of a CRM system and the integration of the modules with the help of IT consulting.

Scenario

As an associate we choose the Austrian hotel "Steinerwirt 1493", a small and innovative company, which started out to use Cloud Computing and has been mentioned in various publications (Manta, 2011; Moutafis, 2011; Smirnova, 2010). The long-term goal of the managers was the migration of the entire on-premise IT-infrastructure to Cloud Computing, starting with the email-system and the customer relationship management system. The small company was not only able to successfully implement these changes due to its financial resources; it also provided flexibility and motivation for Cloud Computing. This allowed rapid responses and adoptions of changed behaviors and factors.

The hotel "Steinerwirt 1493" is located in Zell am See, Austria. Since 1892, the small business is family-owned in its fifth generation by the family Schwaninger. Besides today's managers Gunda and Johannes Schwaninger, the hotel Steinerwirt employs a staff of around 20 and has 28 rooms at its disposal. The hotel reaches an average utilization of 70% per year.

Until 2008, most parts of the hotel's IT-infrastructure consisted of on-premise solutions for email transfer and hotel management. The IT-server of the hotel was running on the one hand Microsoft Exchange Server and on the other hand GMS Felix, a comprehensive hotel management program developed by the GMS Hutter GmbH & Co. KG, based in St. Michael, Austria. Being specifically designed for small and medium sized hotels, the system includes comprehensive modules to manage guests, bookings, and accounting information. Additionally, numerous lists and reports, such as upcoming arrivals, housekeeping schedules or turnover overviews can be generated to support daily operations.

End-user Computing

To reach the long-term goal of migrating the whole IT-infrastructure to Cloud Computing, the two managers decided in 2008 to address its email server firstly. In daily use, the hotel's Microsoft Exchange server caused many failures and had to be fixed by hotel employees. The IT-infrastructure was also maintained by the personnel of the "Steinerwirt". This resulted in a loss of time and costs spent for IT issues instead of the hotel's core competences. In addition, a lack of professional IT knowledge encouraged the managers to search for efficient alternatives dealing with the hotel's email system.

In the managements view, the new possibilities of Cloud Computing seemed to be able to solve the mentioned problems. Hence, maintenance and responsibility was shifted from the costumer to the provider. Experiences from private use and the lack of ready-to-use alternatives led to the decision of Google Apps.

In 2008, Google Apps, especially Google Mail and Google Calendar, was implemented. After an on-premise exchange-server was migrated, Google Mail represented the email client and server. To manage contacts, Google also offers the tool Google Contacts. It allows the user to save contact details like name, email address or phone number out of emails automatically. In addition, Google Documents was established, because the hotel administration system GMS Felix was not able to monitor pre-payment processes of the hotel's visitors sufficiently. Google Documents supports the creation, editing, and administration of documents in an environment similar to Microsoft Office. Even though the establishment of Google Apps yielded several advantages, numerous drawbacks of the encountered IT-system were identified during the evaluation process:

Data redundancy

After the customer confirmed a reservation, the respective booking information was added to a Google Documents spreadsheet as well as to GMS Felix. As the booking data was also the basis for the occupancy overview, inconsistent data could impact other ongoing reservations and inquiries.

Once a new fixed booking had been recorded, the customer received an email with the respective pre-payment information. In addition, the due payment was added to Google Docs as a future reference. Once the payment was received, the record was updated in the spreadsheet and the amount needed to be deducted from the final invoice was presented to the customer. As this process involves financial data, inconsistencies or misunderstandings prove to be especially delicate.

While Google Contacts were registered first, because initial contact was done via email in most cases, the respective data needed to be manually copied to GMS Felix and the Google Docs spreadsheet, once a booking was made. In order to keep data consistent, new information required retrieving and modifying data in all of the programs.

Retrieving and updating additional customer information

Knowing whether a customer had been a guest at the hotel before or was intending to visit for the first time makes a significant difference in how they should be addressed in emails or phone conversations. In addition, segmenting customers according to their individual interests provides a great opportunity to leverage cross selling opportunities.

Even though GMS Felix already provided the possibility to link customers to custom created segments, the process of recording and retrieving proved to be very complicated. Working with GMS Felix involved a long-lasting process of connecting to the server via VPN and launching the application. In order to consequently update and use segmentation data, the process needed to be executed several times a day and therefore represented a significant bottleneck.

The current lack of integration proved to be very time consuming and complicated. Even though most of the information was available in at least one of the systems, users spent a lot of time on switching back and forth between Google Apps and GMS Felix while trying to connect the dots. Hence, a manual integration of this heterogeneous system landscape was needed. Figure 1 outlines the described process of implementing Google Mail and illustrates the problem of lacking integration which had to be solved within a next iteration.

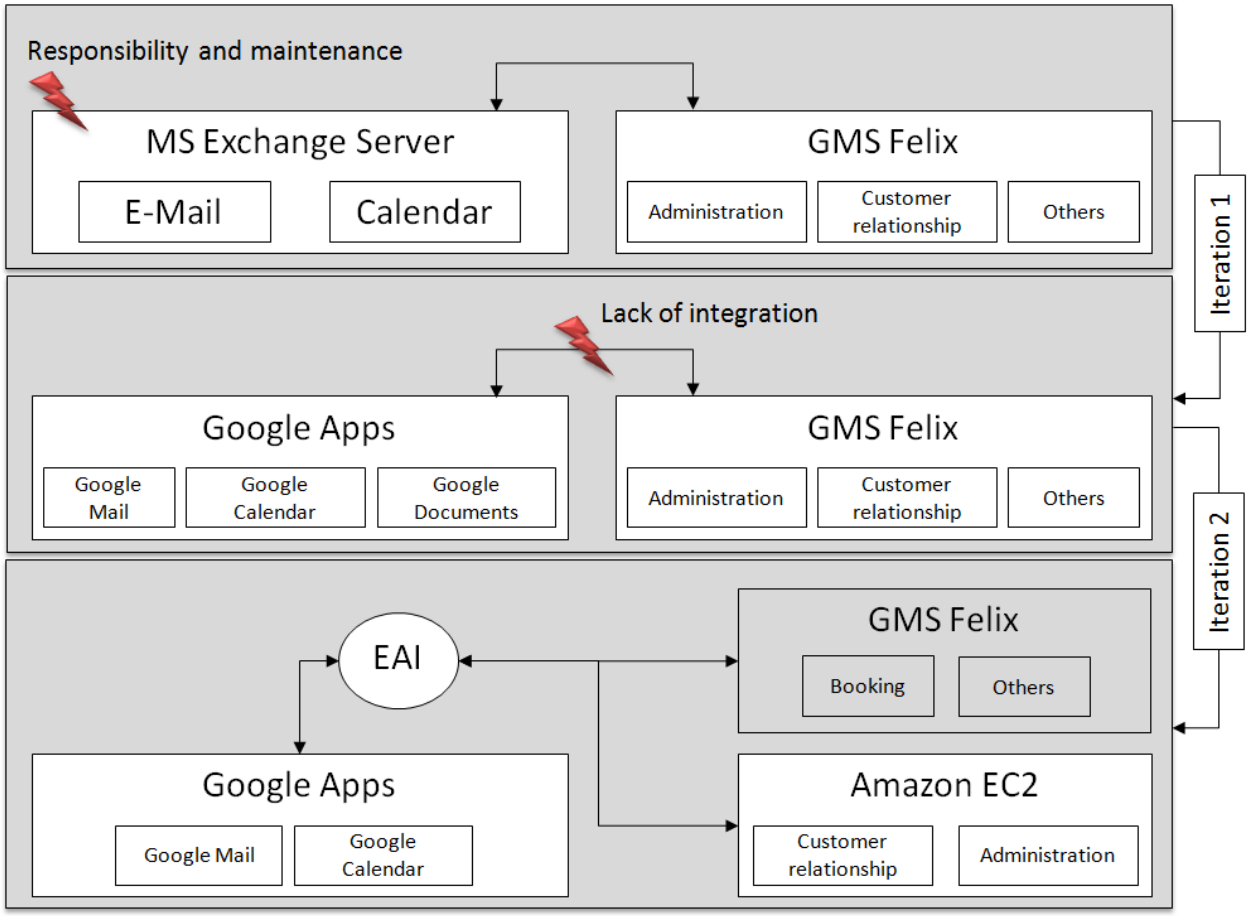

Figure 1: End-user Computing and Integration at the hotel "Steinerwirt"

Integration

The previous section revealed in the imperative, that a further iteration is necessary in order to integrate the Cloud Computing solution Google Apps with the existing infrastructure. So the stated problem was named by the deficient integration of Google Apps and the existing hotel management system GMS Felix in terms of data and functionality. That led to insufficient usability and efficiency due to redundant data and process breaks. Furthermore, the adoptability and the extensibility were strongly restricted. As a second problem, the sparse support of GMS Felix in terms of customer relationship management was mentioned.

Hence, the first step was to proof the further applicability of all components. In detail, the level of integrity of the components and functions in terms of the IT-environment and the processes were analyzed. The most desirable solution was the re-use of functionality or at least synchronization of data. In the other cases, a replacement and replication of the functionality was necessary. Google Apps provided a huge amount of APIs and interfaces which allowed easy integration in the IT-environment. By contrast, GMS Felix, the administration tool of the hotel, was very sparse of interfaces. Only manual triggered import and export of data was possible. For that reason, the replacement of GMS Felix seemed inevitable. Due to the extent of this software, an incremental migration based on functional modules was suggested. These modules should be designed highly integrated with the target system Google Apps. As first module for migration, the required CRM-functionality was suggested.

The first decision in the development phase affected the kind of integration. Traditionally, the integration of new IT-components led to a re-design of existing components or interfaces in order to establish the communication between the new and the existing component (Ruh, W.A., Brown, W.J., Maginnis, F.X., 2001). Because this requires regular and huge adoption effort and is often not possible in Cloud Computing, we applied the Enterprise Application Integration approach. In this concept, a middleware serves the necessary interfaces and data transformation functionality to communicate extraordinary with each, new, and existing components (Lee, J, Siau, K, Hong, S., 2003). This leads to fewer dependencies, re-use of legacy systems, and easier further integration.

The main objective of Google Apps was the handling of the email communication. Related, contacts and contact groups were organized within. Google Calendars were used for event scheduling and Google Documents for prepayments. Google Apps provides also the user-management functionality. On the other side, GMS Felix was responsible for the hotel-management processes. Bookings, customers, and billing were handled by GMS Felix. Due to system-overlapping processes, redundancy and application breaks were constraining the process efficiency. The first module selected for migration was the CRM-functionality. It constitutes no core functionality in a hotel, but provides sufficient importance to evaluate the migration and integration. Furthermore, the CRM-functionality in GMS Felix was realized very sparse. The CRM-module should be able to manage the customers of the hotel with all data and, in addition, the segmentation data like hobbies or interests.

The middleware was implemented on Amazon's IaaS solution EC2 running Fedora Linux with MySQL, GlassFish Community Server using Java with the objective to connect new and old components. The alternative PaaS solution Google App Engine was declined because of issues in portability to other systems and restrictions concerning port usage. The email correspondence was synchronized by use of the correspondence pull service, where the CRM-system pulls the correspondence as needed, as shown in Figure 2. Also the contacts and contact groups were synchronized, in order to avoid data redundancy. Google provides a contextual gadget, which allows the integration of content within Google Mail. Dependent from the sender's address, the relevant guest data, the segmentation of the CRM, the booking history, and a quick link to the guest detail website in the CRM-module was displayed. The user management was not re-implemented; instead the Google Apps user authentication was used. With the help of the OAuth protocol, users were forwarded to the Google Apps login page, which grants the access and the permissions to the CRM-system. This design allowed the easy integration of further modules, because only the mediation Layer had to be adopted.

The CRM-system was also implemented on an Amazon EC2 instance, running the same configuration as described above. It consists of the functional modules guest management, booking, and prepayments. In addition, an administration area was implemented. The guest management contains all guest data such as name, surname, phone numbers, email addresses. Furthermore, it contains the segmentation data like sports or interests of the guest, which are relevant for CRM-functionalities such as guest categorization, newsletters or personal suggestions. The booking module managed the bookings related to the guests. This included the day of arrival and departure, the attendants, the room, and the additional goods and services such as breakfast or a ski pass. In connection, a room management was added. Prepayments were managed in the corresponding module and were related to guests and bookings. The permissions for the modules were stored in the administration area and could be set user- and module-based.

The comprehensive described IT-architecture is illustrated in Figure 2 and represents the Google Apps systems on the left side, the CRM-system on the right side connected through the middleware or mediation layer.

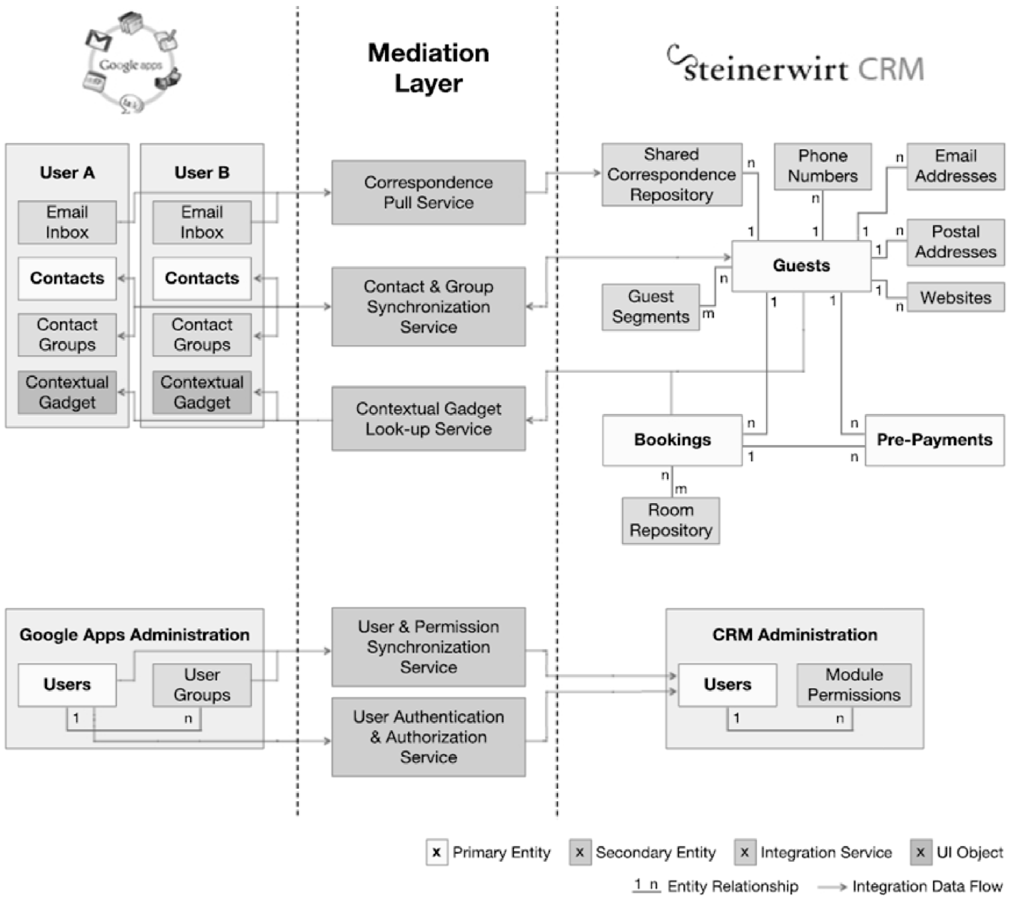

Figure 2: IT-architecture

To measure the success of the CRM-module and the integration by EAI, the key indicators usability, efficiency, and extensibility were evaluated. Therefore, interviews with the users of the systems were conducted and process times and clicks were traced. Because of the manual iterative implementation of the CRM-module, the usability was constantly adjusted to the users' needs. Hence, the user interface was designed similar to the Google Apps layout and structured intuitive. New programming techniques like AJAX led to a satisfied user experience. The integration through EAI avoided breaks through manual application switches and integrated data as required. Efficiency was measured by time and necessary clicks which are required by a process. The time, for example for the lookup for user segmentation, could be cut from 92 seconds to 13 seconds and from 9 clicks to 2. That resulted in an increased efficiency. The claimed extensibility and adoptability of the modules are fulfilled through the integration via EAI. This enables the integration of new modules without changing existing ones. Instead, only a connection between the middleware and the new component has to be established via interface.

In summary, the two main goals were fulfilled. The CRM functionality was migrated from the sparse on-premise solution in GMS Felix to a cloud-based, implemented solution using Amazon's EC2 Cloud. Also the integration of the modules Google Apps and the CRM system was successful, whereby the integration of GMS Felix lead to problems due to missing interfaces at GMS Felix. To solve this issue, GMS Felix should be replaced by cloud solutions step-by-step. After the easy usage of Google Apps by the end-user, the individual CRM and the integration middleware had to be implemented by hand from programmers. This enabled an individual problem solving and process improvement, but also led to dependencies to IT-consultants or the IT-department.

FINDINGS AND DISCUSSION

The goal of this paper was to examine on the one hand, if Cloud Computing encourages End-user Computing, and on the other hand, how end-user designed solutions can be integrated in existing IT-architectures. Therefore, we conducted a case study following the design science approach in an Austrian hotel. In this paper, we introduced two relevant iterations in the design science process. Firstly, we showed the migration of an on-premise email solution to the cloud-based Google Mail application. Secondly, the integration of a CRM module with the help of Enterprise Application Integration was presented.

Within the first iteration, we were able to observe End-user Computing in terms of system migration by less IT-skilled managers of the hotel. This was possible because of the easy and guided installation and configuration of the SaaS solution Google Apps. The current hype of Cloud Computing and huge marketing and publicity actions of the providers in combination with the availability of well-engineered products result in an increased awareness of the Cloud Computing alternative. The benefits of this approach such as easy installation and usability, and the flexible, modular design of the applications promote End-user Computing and results in an increased independency from the IT-department. This can be transferred to many other SaaS providers as well; even IaaS providers try to enable the end-user through easy and intuitive installation and usage. Only PaaS is limited mainly to programmers due to the necessity of programming skills.

Hence, the end-user is able to create a working cloud-based application. Rarely, these applications are running isolated; therefore, they have to be integrated in the existing IT-architecture.

To examine the integration of end-user designed cloud-based applications in the existing IT-architecture, we conducted the second iteration of the case study. The main goal was to integrate the Google Apps application with the existing on-premise hotel management software GMS Felix. Because of sparse interfaces provided, GMS Felix was not suitable for the integration with other software. Hence, it should be replaced modularly by cloud-based applications. Starting with the CRM functionality, the integration issue still existed. Therefore, a cloud-based middleware, following the EAI approach, was implemented. This middleware serves interfaces and data transformation tools extraordinary for every IT-system. For further integration, only the connection between middleware and new IT-system would be necessary. The end-user is hardly able to create such an integration middleware for several reasons. Firstly, he usually has no or only sparse knowledge about the existing IT-infrastructure and conceptual skills in building enterprise IT-architectures. Furthermore, cloud-based applications normally have many, but less standardized interfaces and APIs to work with. That requires a deep understanding and knowledge of the different implementation possibilities. Finally, the end-user acts problem-centric and creates isolated solutions to solve current issues; for integrating applications, a system-centric approach regarding the whole IT-architecture is necessary.

In summary, Cloud Computing encourages End-user Computing in a new dimension, but results in a new issue regarding the integration of isolated applications. The end-user is hardly able to solve this issue without IT-support. Cloud-based EAI can act as an integration middleware, fitting interfaces for data synchronization assumed. Hence, the end-user becomes more independent from the IT-department concerning service design; for integration issues, the IT-department is still needed.

Restricting factors are the support of End-user Computing of the deployed cloud solution and the IT-skills of the end-user. Depending on this, the observation of End-user Computing may shift from conceptual help and motivation to control in order to avoid shadow IT and unstructured growth. Future research could for example deal with the development of a more guided process for End-user Computing that sets boundaries and reduces the need of control. Hence, further research should examine how such a guided process could be designed and what agreements from providers' side are necessary.

The integration via EAI works well, but results in dependencies to the IT-department. Furthermore, the IaaS virtual machine must still be administrated and maintained. One solution might be the Integration Platform as a Service (IPaaS) which enables the EAI approach on a PaaS for the purpose of integration. Implications for further research are the provision of automated EAI modules that enable the end-user to integrate cloud-based systems in the existing IT by himself.

REFERENCES

1. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.; Katz, R.; Konwinski, A. (2010) A View of Cloud Computing. *Communication of the ACM*, 53(4), 50-58.
2. Davis, G.B.; Olson, M.H. (1985) Management Information Systems: Conceptual Foundations. Structure, and Development, McGraw-Hill Book Co., New York, 532-533.
3. DB Research (2012) Cloud Computing – Freundliche Aussichten für die Wolke http://www.dbresearch.de/PROD/DBR_INTERNET_DE─PROD/PROD0000000000283604/Cloud+Computing%3A+Freundliche+Aussichten+f%C3%BCr+die+Wolke.pdf, retrieved on 02/22/2012.
4. Experton Group (2010) Cloud Computing startet in Deutschland durch – Ausgaben und Investitionen in 2010 bereits über eine Milliarde Euro. http://www.experton-group.de/press/releases/pressrelease/article/cloud-computing-startet-in-deutschland-durch-ausgaben-und-investitionen-in-2010-bereits-ueber-eine-milliarde-euro.html. *Experton Press*, retrieved on 02/15/2012.
5. Gartner Inc. (2011) Gartner's 2011 Hype Cycle Special Report. http://www.rolotec.ch/blog/archives/2011/08/gartners_2011_h.html, retrieved on 12/18/2011.
6. Hetzenecker, J.; Wiener, M.; Amberg, M. (2011) User-Designed Software as a service – can conceptual work be shifted to the end-user? *Proceedings of the Americas Conference on Information Systems*, 129.
7. Hevner, A. R.; March, S.T.; Park, J.; Ram, S. (2004) Design Science in Information Systems Research. *MIS Quarterly, 28* (1), S. 75-105.
8. Lee, J.; Siau, K.; Hong, S. (2003) Enterprise integration with ERP and EAI. *Communications of the ACM*, 46, 54-60.
9. Lefkovits, Henry C. (1978) A Status Report on the Activities of the CODASYL End User Facilities Commitee (EUFC), SIGMOD Record.
10. Manta, C. (2011) Der Steinerwirt hebt ab, *CIO.de*, http://www.cio.de/subnet/google/anwender-und-praxis/2264292/, retrieved on 12/20/2011.
11. Moutafis, J. (2011) Wie ein Hotel durch die Cloud seine Server los wird, *chip.de,* http://business.chip.de/videos/Wie-ein-Hotel-durch-die-Cloud-seine-Server-los-wird-Video_48701475.html, retrievd on 12/20/2012.
12. National Institute of Standards and Technology (2011) The NIST Definition of Cloud Computing. http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf, retrieved on 06/11/2011.
13. Rockart, J. F.; Flannery, L. (1983) "The Management of End User Computing," *Communications of the ACM,* 26, 10, 776-784.
14. Ruh, W.A.; Brown, W.J.; Maginnis, F.X. (2001) Enterprise Application Integration: A Wiley Tech Brief: How to Successfully Plan for EAI, Wiley.
15. Smirnova, J. (2010) Revolution aus den Wolken *Welt Online*. http://www.welt.de/print/welt_kompakt/webwelt/article10147880/Revolution-aus-den-Wolken.html, retrieved on 12/20/2011.
16. Vaishnavi, V.; Kuechler, B. (2011) *Design Science Research in Information Systems*. http://desrist.org/design-research-in-information-systems/, retrieved on 12/08/2011.
17. Vouk, M. A. (2008) Cloud Computing – Issues, Research and Implementations. *30th International Conference on Information Technology Interfaces*, 31-40.
18. Weinhardt, C.; Anandasivam, A.; Blau, B.; Borissov, N.; Meinl, T.; Michalk, W. (2009) Cloud-Computing – Eine Abgrenzung, Geschäftsmodelle und Forschungsgebiete. *Wirtschaftsinformatik 51(5)*: 453-461.