User-controlled data sovereignty in the Cloud

Marc Mosch
Faculty of Computer Science Faculty of Computer Science
Technische Universität Dresden
01062 Dresden, Germany
Email: marc.mosch@tu-dresden.de
Supervisor: Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill (alexander.schill@tu-dresden.de)

***Abstract*—Cloud computing plays an increasing role in the IT-market. Promised scalability and flexibility attracts enterprises as well as private end users. The loss of data sovereignty is seldom addressed but nevertheless a serious threat to users. The FlexCloud[1] research group proposes an approach that enables users to profit from public cloud scalability without losing data sovereignty. The proposed dissertation derives from the work within the research group. The goal of the PhD thesis is an architecture design of a gateway for the secure outsourcing of sensitive data.**

***Keywords*-Cloud Computing; Gateway; Storage; Data Security; Privacy; Data Sovereignty**

I. Introduction and Motivation

In this paper I introduce my PhD intention and propose a way to benefit from cloud computing advantages without suffering from the drawbacks. In this context the term ”data sovereignty” in the title refers to the user's self-determined control over his or her data. It means to be sure that important data—once outsourced to distant servers—is still only under user's control and secure from manipulation by third parties. So the meaning of sovereignty is not to be mistaken as sovereignty in terms of geolocating like it is for example understood in [1]. Such territorial assumptions are not of importance for the presented approach.

After the motivation and an introduction of the Personal Secure Cloud (π-Cloud) a scenario shows how the ability to benefit from cloud's scalability without losing the data sovereignty enriches daily life. Afterwards the concept of my thesis points out problems to be solved, lists the extracted research questions and my solution approach. Then related work is discussed followed by a short summary and outlook.

What is cloud computing? There is no unique definition that everybody agrees on. A lot of definitions for example the one by NIST [2] are complex. Cloud computing is nothing completely new. It is a combination of long existing technologies. In short, cloud computing can be understood as a distributed service approach which involves virtualisation of physical servers and their rental, or the rental of services running on them. Because the focus of the thesis is on data security, only storage services are taken into consideration. A distinction can be drawn between four different kinds of clouds: private clouds, where the user is the owner of the infrastructure and in control of the cloud; community clouds, a kind of merged private clouds where the infrastructure is shared among several users with shared concerns; public clouds, which are not owned by the user and which are open to paying customers; and hybrid clouds, which are a combination of the stated types of clouds. End users as well as companies are attracted by the fancy term “Cloud” and the marketing promises associated with public clouds—unlimited scalability and availability. Companies at the management level see cost savings that arise from outsourcing bulky high-maintenance server farms. For end users cloud computing promises a unified database—reachable any time from any place—without the need to manually synchronise smart phone, notebook, tablet, personal computer and other devices. As a welcome side-effect backup-strategies are a matter of cloud providers—no need to care about. What most end users and companies disregard is the high price they pay for the comfort of public cloud services—they pay with the loss of their data sovereignty. Sensitive data once outsourced—whether stored or processed by public cloud services—are exposed to loss, abuse and manipulation. To benefit from public cloud approaches without losing the data sovereignty an open source solution with transparent client side encryption for confidential data has to be established. This is what the FlexCloud project is focused on.

II. Introducing the π-Cloud idea

A central mediating instance—hereinafter referred to as π-Box (Personal Secure Box)—should encapsulate the variety of devices and thereby form a π-Cloud (Personal Secure Cloud) as it is illustrated in Fig. 1. The π-Box consists of a service platform based on SPACE [3] and the π-Gateway. SPACE is a service broker architecture that features propagation, booking and deployment of services as well as Service Level Agreement (SLA) negotiation.

The communication to the outside is coordinated by the π-Gateway in order to protect user's privacy. This is done by distinguishing between private (sensitive) and public (non-sensitive) data and encrypting the former one in a way that only authorized users are able to access it. The architecture should be designed to fit end user requirements as well as industry needs.

[1]FlexCloud (Flexible Service Architectures for cloud computing): http://flexcloud.eu/. This work has received from the European Social Fund and the Free State of Saxony, Germany, under project number 080949277

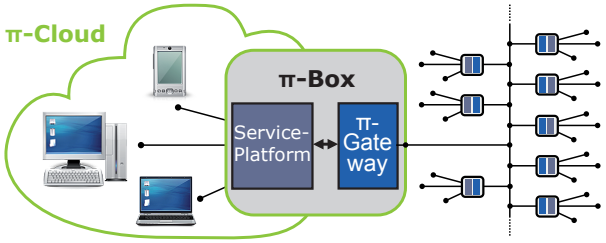

Figure 1. This is an initial architectural layout of the π-Cloud with a coarse grained subdivision of the π-Box into the Service Platform SPACE and the π-Gateway. The latter connects it with other π-Clouds or public clouds.

The unified database for the devices could be placed on a home server or an external hard drive connected to the Wireless-LAN-router or a micro- or a plug-server at home. It might actually be part of a virtual machine hosted by a cloud storage provider whom the user trusts. So might be parts of the π-Box. These parts can also be distributed over the devices within the π-Cloud. For example the Data Module—a data analysis module that defines which data might be sensitive and which might be not—could be integrated in every device to enable self-sustaining file exchange.

III. Scenario

The following "smart home" scenario gives a brief overview about what the π-Cloud is aimed at and which benefits are to be expected in daily life. Imagine a household in the near future. Devices are "intelligent" to ease user's way of living. Therefore they are connected with each other. They form a π-Cloud—mediated by the π-Box. Devices involved in the processes of generating, collecting, storing and editing data like computers, smart phones, printers and notebooks are part of the π-Cloud. So could be household infrastructure like heating, ventilation, washing machines, microwaves and fridges. The latter recognize their content for example by RFID tags. Let us assume there is only one bottle of milk left in a fridge. The number of bottles—for example determined by the number of associated RFID tags—reaches a predefined threshold. This fact triggers the fridge to alter the user's digital shopping list, which is part of a calendar file. The file is stored in the user's unified database—reachable via all types of authorized devices. Since there is only one bottle of milk left, the priority for the purchase is set to a high level and is therefore entered for the same day. The examination of the day's appointments reveals that there is no time left for shopping. It is time for a contingency plan: the status of the milk in the list is set to "delivery" and a query for a delivery service is placed against the service broker module of the π-Box. This module holds two lists—one for services provided within the user's domain and another for foreign services, provided for example by other π-Boxes. Assumed that a milking machine is not part of the π-Cloud, none of the user's devices is able to deliver milk as a service. In this case the query is placed solely against the external list. It is set according to the user's predefinitions entered once. The user likes to drink organic milk with a fat content of at least 3.5 %. And because local farmers should be supported, the milk has to be from a farm within a 50 km radius from the user's living place. According to the urgency the π-Box will only take offers into consideration that can be delivered at the same day. From the offers fulfilling these requirements the cheapest is chosen. Every night the calendar file is checked by management routines. Let us assume that an entry for the next day points out the beginning of business trip lasting several days. At a normal working day the return of the user in the evening might be plausible and therefore the cleaning robot vacuums at daytime. When the user leaves the house for several days, the robot will not start the cleaning immediately. It will wait and use the cheaper electricity at night. Trips several days in the future will influence the quantity and balefulness of food entered into the shopping list. But these are not the only adoptions. The target location of the business trip is checked for Internet network coverage. When the coverage is identified as poor a connection to the unified database from there is assessed unlikely and important files for the trip are migrated. Depending on files' age, metadata and content the importance of the files is evaluated. Supposed the user prepared material for a presentation the files might have been created a few days before the trip. Only files edited in the last few days should then be of importance. If the number of them is too huge for the presentation device more advanced filtering takes place. Slides might for example be checked as following. Is their file name related to the trips place or date of the presentation? Or are these information parts of the slides front page? Additionally information about access rights granted to the files before, are handed over to the presentation device. Even if being away from home—if connection of any kind can be established—the user gets access to all files in the unified database. As an example for the file exchange with other users imagine the user meets a friend who was awarded a few weeks before. During the awarding ceremony the user took some photographs with the smart phone. Now the friend asks for these photographs. The user already uploaded the files in the unified database where one of the powerful devices applied face recognition. According to a predefined rule access to the files for the depicted person was granted—in this case for the friend. The photographs stored in the smart phone were replaced by encrypted versions. And the key for the decryption was encrypted with the public key of the persons that were authorized for access—in this case the depicted friend. The encrypted photographs remain on the smart phone for some weeks. This safekeeping is done because the need for sharing of data seems to be more likely in a period shortly after the creation. If the user meets the friend in an area with no network coverage the safekeeping pays off. The user transmits the encrypted photographs and the encrypted decryption key to his friend's smart phone for example via Bluetooth.

If there are no buffered files in the smart phone the user might also hand out links to the files in his unified database to the friend. After receiving the files the friend can decrypt the decryption key with his private key and then decrypt and access the photographs. To enhance the ease of use, this process might be kept transparent by automation.

IV. Concept

The dissertation is derived from the work-package of the FlexCloud project which is dedicated to the transparent service deployment. Within the dissertation the basic architecture of the π-Gateway should be defined. Furthermore a concept shall be created to distribute it and its components among several instances in a redundant and secure way.

A. Problems

The scenario shows applications for a system that features a unified protected database and helps the user sharing data with other persons in a secure manner. In this scenario several problems arise.

The way data are published today levers the shielding effect of common firewalls. Data is stored on distant servers where the user is not able to control for example who gains (physical) access. The outsourcing of the data results in a highly dynamic environment that involves the need for a replacement of common firewalls. This is where the π-Box—especially the π-Gateway—should be applied as a cloud management or cloud control mechanism. The π-Cloud resulting from the encapsulation of the user's devices is a personal domain, a kind of safe harbour for sensitive data in hostile cloud environments. To be competitive with common cloud solutions data in the π-Cloud have to be available everywhere independent from the device. In order to protect user's privacy it must be possible to distinguish sensitive from non-sensitive data on the go and share these decisions over several devices to avoid overload by duplication of efforts.

Another problem arises from the fact that in such a highly dynamic environment 100%-connectivity cannot be guaranteed. Imagine the depicted friend from the scenario asked for the photographs before they could be synchronized with the unified database. Then—if the smart phone is not performant enough—no face recognition and no encryption could be done. In this case the smart phone could not hand out encrypted files. This shows the need to provide links in advance in case the synchronization mechanism is delayed and manual user intervention to grant access is undesired.

So the main requirements the π-Gateway has to meet are the encapsulation of one's devices, the management of a unified database, the ability to distinguish between sensitive and non-sensitive data and the possibility to handle offline situations of single devices.

B. Research Questions

The following research questions arise from the presented scenario and my work at the FlexCloud research group. They result from the need to create a π-Gateway that serves as a kind of firewall in a highly dynamic environment where common firewalls are useless.

1) How does the basic architecture of the π-Gateway have to look like? Should it be a central instance that manages all the other devices in a star topology or should it be a kind of virtual machine or even an application that can be migrated from one device to another?
2) Which criteria are important for the election of the device that serves as π-Gateway/π-Box?
3) Which are the basic functions of the π-Gateway that every device has to provide to host it?
4) How can the encryption and access decisions taken be shared among the devices in order to avoid overload by duplication of efforts?

C. Solution Approaches

The following ordered solution approaches address the research questions and are labeled accordingly.

1) The design of the π-Gateway should be based on a requirement catalogue that has to be developed first. This design has to address the distribution of the π-Box or parts of it. The Data Module—that part of the π-Gateway that distinguishes sensitive from non-sensitive data—might for example either be part of a central π-Box or existent in every device permanently or temporarily by migration. An initial layout of this module can be seen in Fig. 2. If it is distributed over the devices it has to be ensured that the decision base is shared among the devices. So if this information is not solely stored in the π-Box a suitable distribution and synchronisation mechanism is needed to share the information, how to treat which data in a consistent and efficient way.
2) The strategies for the election of the device that serves as π-Box have to be developed with the dynamic needs of a mobile environment in mind. If the user leaves for example the house and takes only a smart phone along to a location without network coverage, the π-Box/π-Gateway has to migrate to the phone.
3) A criteria catalogue that has to be created should help defining the basic functions of the π-Box/π-Gateway that every device has to provide to host the π-Gateway. Depending on the device's performance it might be necessary to outsource some functions to a more powerful device that is always available—like for example a broadband Internet gateway or a trusted cloud service.
4) Overload by duplication of efforts might be prevented if the decisions taken are shared among the devices. This means every device has to synchronise its decisions with the other devices if a connection is established. In times without a connection this approach ensures that buffered files can be shared with other persons based on access rights already granted before. Thus a potential time-consuming

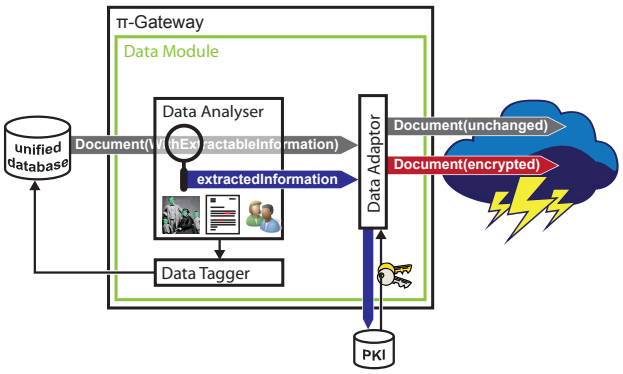

Figure 2. This is an initial architectural layout of the Data Module, that part of the π-Gateway, where the outgoing data are analysed and encrypted (using a Public-Key-Infrastructure (PKI)) if necessary, before it is stored on the servers of a potentially insecure cloud storage provider. Previously untagged data are enriched with the gathered metainformation to avoid duplication of efforts.

analysis of the files does not have to be repeated on a low performance device like a smart phone. If the synchronisation is delayed and the current device is not able to encrypt the requested files it may nevertheless hand over links that are valid in the future. The files could be identified by metadata like timestamps. The prediction of future link-paths could be achieved by a database which associates these timestamps with link-addresses.

D. Related Work

The π-Cloud idea is focused on cloud storage but also intends to support the trading of services which might go beyond storage. In this context it is important to distinguish two types of data handling. If data are only stored they are called Data-at-Rest. If the data should be processed they are called Data-in-Use. Cloud services intended to process data have to work on plaintext rather than on ciphered one. There are approaches addressing this problem [4]. Unfortunately, at the current technical state the processing of algebraic operations on ciphered text—as it is possible with homomorphic encryption—is far too slow to be used in practice. So at the moment, if data are outsourced user's privacy can only be ensured for Data-at-Rest. Given these circumstances, the π-Gateway has to deny access to sensitive data for public cloud services that intend to process the data. A deeper look at available cloud storage solutions reveals limitations in the protection of client data. Dropbox [5] and many others encrypt solely on server side which requires trust in the provider since the encryption key is deposited at the provider's server and the user has no control over it. Only a few providers like Wuala intend encryption on client side. But the software they provide for encryption and upload are not open source. So the weak point is only shifted and the usage of such solutions is a matter of trust in the provider again. Trust provided, Wuala offers an interesting range of features. In addition to the "Cryptree" protocol [6]—a sophisticated key exchange scheme which enables users to share encrypted files with other users—local disk space can be traded for additional storage space in the cloud. Similar approaches are to be found in [7] and [8]. The former emphasize that most internet users have hardware which is much more powerful than needed for standard tasks like web browsing, blogging and chatting. Proceeding from that insight they suggest users might short-lease their resources. Based on similar assumptions the latter introduce the Cloud@Home middleware and a belonging software architecture implementation which should build a cloud based on heterogeneous nodes. In addition to backups in Public Clouds mutual backups between several π-Cloud users are thinkable and might be handled likewise. Based on the work presented in [9] information dispersal algorithms will be used to ensure n-redundant backups without the need to allocate n-times the space of the original data.

In addition to the limitations of public cloud storage solutions regarding the protection of client data management issues arise. Today's cloud storage solutions lack automated management interfaces so that access control involves manual intervention by the user. This complicates the migration and integration of existing infrastructure and lowers the ease of use. The π-Gateway is intended to automate access control as far as possible.

Eben Moglen—professor of law at the Columbia University—tries to make people aware of the threats that are accompanied with technological changes like social networks and cloud computing. According to him, companies like Facebook—whose database reached 21 petabyte in May 2010 [10]—gather data about users to such an extent, that it might be easily used for suppression and the creation of a police state. The FreedomBox foundation [11] is fund-raising for the realisation of the vision based on Moglen's idea of a decentralized architecture that enables users to share information hidden from commercial or governmental surveillance. This architecture includes that every participant contributes a low power, low cost plug server where everyone's data are stored encrypted and distributed. The π-Gateway approach is similar to some extent. Started independently from the FreedomBox approach and Moglen's vision the π-Gateway stands out with its concept of services and their distribution. The approach to encapsulate the user's devices is also out of FreedomBox's scope. Another difference can be found in the performance optimization that should be reached by differentiating between sensitive and non-sensitive data, so that not all the data have to be encrypted. The file examination taking place in the Data Module might be based on existing open source data analysis software like Apache Lucene [12] which therefore have to be compared and checked for suitability.

Approaches from different research fields have to be analysed for a proper design of the π-Gateway. Because of the large scale of this task the current status of the thesis allows only a brief overview over the involved research fields. The architecture could for example be based on hybrid peer-to-peer techniques—as used in [13] for the

purpose of resource sharing between organisations—as well as on backup strategies. In the first case one of the devices would be chosen as π-Gateway/π-Box—like a kind of super-peer. Grid computing solutions even though not optimised for distributed requests but for the fulfillment of a common goal—might nevertheless also deliver appropriate approaches. Knowledge from mobile agent technology might be of use as well, even though mobile agents are of little importance today. Especially knowledge about transmission of states is of interest for the creation of an architecture for the decision exchange. Today's insignificance of mobile Agents might be substantiated in the inability to protect carried data, code and the runtime environment reliable [14]. This shortcoming is irrelevant for the π-Box architecture because it can be assumed that only trusted devices exchange states. Backup strategies might provide Token-techniques that could help assigning responsibility to devices. Depending on the importance of migrated parts of the π-Box transaction control might also play an important role.

V. Summary and Outlook

I have discussed the need for a secure access to cloud computing services which leads to the π-Cloud idea that was introduced afterwards. Then a scenario which showed the expected benefits of the π-Cloud idea was presented. In the concept chapter the arising problems where discussed, followed by the resulting research questions, a first solution approach and related work.

The next steps are dedicated to the creation of a requirement catalogue. As mentioned in the related work subchapter many different fields of research have to be observed in order to design the architecture of the gateway properly.

References

[1] Z. N. Peterson, M. Gondree, and R. Beverly, "A position paper on data sovereignty: The importance of geolocating data in the cloud," in *Proceedings of HotCloud, 2011*, 2011.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology (NIST)*, 2009.

[3] "SPACE service platform - free, user-centric, modular, powerful, integrated, driving the internet of services," http://serviceplatform.org/, last accessed on 2011-05-25.

[4] C. Gentry, "Computing arbitrary functions of encrypted data," *Communications of the ACM*, vol. 53, no. 3, p. 97–105, 2010.

[5] R. Spoor and A. Peddemors, "Cloud storage and Peer-to-Peer storage," 2010. [Online]. Available: http://www.surfnet.nl/nl/Innovatieprogramma's/gigaport3/Documents/EDS-3R Cloud and p2p storage-v1.1.pdf

[6] D. Grolimund, L. Meisser, S. Schmid, and R. Wattenhofer, "Cryptree: A folder tree structure for cryptographic file systems," in *Reliable Distributed Systems, 2006. SRDS'06. 25th IEEE Symposium on*, 2006, p. 189–198.

[7] C. Teixeira, R. Azevedo, J. S. Pinto, and T. Batista, "User provided cloud computing," in *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, Melbourne, Australia, 2010, pp. 727–732. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5493398

[8] V. D. Cunsolo, S. Distefano, A. Puliafito, and M. Scarpa, "Applying software engineering principles for designing Cloud@Home," in *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, Melbourne, Australia, 2010, pp. 618–624. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5493416

[9] R. Seiger, S. Groß, and A. Schill, "SecCSIE: A Secure Cloud Storage Integrator for Enterprises," in *International Workshop on Clouds for Enterprises (C4E)*, Luxemburg, Sep. 2011, accepted for publication.

[10] "HDFS: facebook has the world's largest hadoop cluster!" http://hadoopblog.blogspot.com/2010/05/facebook-has-worlds-largest-hadoop.html, last accessed on 2011-04-23.

[11] "FreedomBox foundation," http://freedomboxfoundation.org/, last accessed on 2011-05-11.

[12] "Apache lucene website," http://lucene.apache.org/, last accessed on 2011-05-29.

[13] A. Gupta and L. K. Awasthi, "Peer enterprises: A viable alternative to cloud computing?" in *Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference on*, 2009, p. 1–6.

[14] H. Peine, *Run-Time Support for Mobile Code*. Universität Kaiserslautern, 2002, ISBN-3-925178-93-7.