In: Eichler, G., Wienhofen, L.W.M., Kofod-Petersen, A., Unger, H. (Eds.), 12th International Conference on Innovative Internet Community Systems (I2CS 2012). Bonn : Ges. für Informatik, 2012. pp. 155-164

Lazy Disclosure – Mixing Cloud and Local Storage

David Bouck-Standen, Jörg Cassens and Michael Herczeg

Institute for Multimedia and Interactive Systems
University of Luebeck
Ratzeburger Allee 160
D-23562 Lübeck, Germany
{bouck-standen,cassens,herczeg}@imis.uni-luebeck.de

Abstract: The use of multiple computer systems by a single user makes data transfer a necessity. Cloud storage solutions provide synchronisation features and file accessibility across systems but the exact location and type of storage varies between services. Many systems rely on a client-server architecture where the data is held at the data centers of the service provider. This leads to several difficulties for the end user, for example with regard to privacy or data security. This is countered by the convenience of highly available servers. If the user wants to minimise exposure to the challenges of Cloud-based services, then he is required to either host a server infrastructure himself or to rely on inconvenient on-demand synchronisation by means of file transport protocols or even physical storage devices. The approach presented here tries to harvest the positive aspects of using Cloud-based services while keeping the user in control of his data. To this end, we focus on a hybrid system of a client-server and a peer-to-peer architecture which expands the Cloud to the user's system giving him full control over which data he selects to be stored on a server in the Cloud and which data he prefers to keep private on personal computing devices while ensuring unified access to both types of storage over the network.

1 Introduction

Cloud computing is best described as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources like for instance networks, servers, storage, applications and services, that can be rapidly provisioned and released with minimal management effort or service provider interaction [MG09].

Cloud storage solutions allow users to store their data online making files accessible from any location with an Internet connection [AFG+09]. These solutions are also utilisable for synchronisation, collaboration or backup purposes.

Uploading data requires trust in the provider [CKS09] and his abilities to ensure the safety and security regarding both storage of and access to the uploaded data. The data's storage location is concealed within the Cloud and the servers storing the data may not be run in the same country the user resides in. This also hints at legal issues as laws governing data protection and data security vary between countries.

Most Cloud storage solution providers as i.e. Amazon S3[1], Microsoft SkyDrive[2] or Dropbox[3] offer a software tool which once installed on a client system manages the file uploads. When configuring the tool it is common practise to select a root folder on the client system. For Cloud synchronisation or upload, the user is required to move files and folders into this root folder.

We propose an integrated hybrid solution which allows the user to keep files on his personal computers and in the folder structure he is accustomed to. The user is free to choose whether to synchronise the selected folders with a Cloud storage, giving convenient access in a client-server architecture, or to make the files available to other computers without uploading them into the Cloud. In the latter case, only metadata is synchronised with a server, making it available to other computers hence providing a searchable file index. The actual file content is fetched directly from the client system that makes the file available. This minimises the user's exposure to the Cloud.

2 Cloud Storage

Internet usage continues to increase throughout Europe. Since 2006 the number of households with Internet access grew by 39% to a total of 70% in the EU27 countries, 87% served by a broadband connection [SL10]. Fixed broadband subscriptions also increased in the U.S. by 29% and Australia by 37% since 2005 [Ofc11] indicating a clear international trend toward increased global networking and Internet usage.

In the course of growing Internet usage, a demand for storage solutions is present. This finding is supported by two independent surveys carried out by our research group in 2011. The first survey was concerned with a mobile client for the Network Environment for Multimedia Objects (NEMO) [LCHS10], the second with an augmented reality system. Both systems use the NEMO system for Cloud storage [Sto10].

The first survey shows that 88.4% of the respondents are synchronising files and folders between systems of which 63.0% are using a direct connection to i.e. network storage, external hard drive or a USB-Flash drive [Kha11]. Only 37.0% of the respondents are actively using a Cloud storage service. In the second survey 60.4% of the respondents answered having heard of the term Cloud [Ebe11].

These findings are supported by the latest survey of user habits of Spanish Internet users where 10.6% of 34.096 respondents said they had used FTP for file transfer on the day before the survey [AIM12]. In addition, the latest OfCom report states the increasing interest of Cloud computing to public bodies [Ofc11] even towards a European strategy for Cloud computing [Kro12].

[1] https://aws.amazon.com/s3/

[2] https://skydrive.live.com/

[3] http://www.dropbox.com/

2.1 User-Centered Design

In the following we take a user centred design process as a basis. By presenting a paper prototype to a potential user in an interview, the interviewee has the option to make remarks, changes and additions to the prototype [STG03], [TBBS06b], [TBBS06a].

The concept focuses on users who are accessing their data from multiple systems, are often on the move and cannot take all their data along with them at all times, or need to synchronise their data between systems. In the process of our research, a number of interviews were completed with users representing possible target groups of users [BS11]. The interviews were conducted with a university professor, members of mid- and high-level management as well as university students. The following vision scenario makes use of a persona [CRC07] derived from key characteristics of the interviewee's.

2.2 Scenario

Phil McNeil is a project manager with a company in the oil-business. Being a commuter, Phil has to travel an hour each direction every day. On his way to work he listens to the radio and notices a particular song. Having arrived at his office, he uses a computer system accessing his data stored inside the Cloud. The recent changes he made to a presentation at home yesterday evening were downloaded automatically at system start-up and Phil was able to continue work right where he left it, being also able to see the latest drafts his team finished this morning.

During a break, Phil remembers the song he listened to earlier on. Searching his personal audio library within the Cloud, he notices he had already purchased that album. He then downloads the entire album to his MP3-Player.

Being in a management position bearing responsibility for various projects, he often has to travel, even on short notice. In the afternoon Phil gets summoned to a conference he has to attend. In this case, the data stored inside the Cloud enables him to access all his files regardless whether they have been synchronised and uploaded before. This way Phil can keep track of his files without having to download everything to his laptop, he usually takes to conferences, which has only a limited storage capacity.

Sitting at the gate prior to boarding, Phil chooses a movie from his library stored on his computer at home he might want to watch this evening at the hotel. He notices that his computer at home is switched off and cannot transfer the file immediately but the system saves his request. Having arrived at the hotel, Phil switches on his notebook, which automatically downloads the movie through the Cloud completing his request. The computer at home has in the meantime been switched on by his wife and has, according to his earlier request, automatically uploaded the data into a cache storage location within the Cloud.

3 Concept and Implementation

The outlined scenario describes a hybrid system consisting of a Cloud storage solution enabling a client to up- or download data in a client-server context as well as a peer-to-peer context. In client-server context the data is highly available from the Cloud as all data is uploaded and stored inside the Cloud. Data which is available in a peer-to-peer context may not be obtained at all times, but is mainly stored on the client's system. As the trend of the number of broadband connections is showing an increase, connection speeds are negligible.

3.1 Concept

In our concept the user selects one or more folders on the client system which are enabled for synchronisation or made available. These locations will be referred to as repositories.

The content of each repository is represented by a catalogue containing information about files and folders of the repository, their size, meta-data, file type and preview. Settings regarding repository access and sharing policies are also set within the catalogue. A catalogue of a repository enables the user to browse and search the repository without having to download all its content.

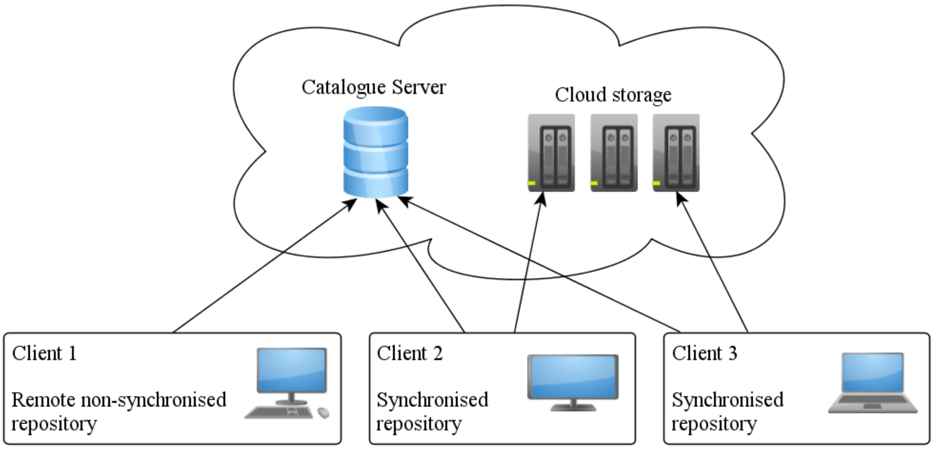

Figure 1: Client 1 holds a remote non-synchronised repository connecting to a catalogue server in the Cloud. Both Client 2 and Client 3 hold a synchronised repository. Catalogue information on the repositories as well as the data is stored inside the Cloud.

As shown in Figure 1, for this concept the Cloud consists of two components: a Catalogue Server and Cloud storage. The latter is primarily serving as storage space for repository contents. This storage is accessible through an Internet connection with high availability. The Catalogue Server represents a central storage for all catalogues. Each catalogue is stored under a unique identification reference identifying the repository.

Two types of repositories are distinguished:

1. **A synchronised repository** is automatically synchronised with the Cloud storage and is therefore available in a client-server infrastructure. Features like for instance version control or multi-user collaboration[4] when working inside a repository are available. Synchronisation between multiple systems accessing the same repository, as shown in Figure 1 is also supported. A synchronised repository is accessible as long as the storage location is available inside the Cloud.

2. **A remote non-synchronised repository** contains data which is available in a peer-to-peer infrastructure. A Catalogue Server stores information on repositories and their location, as they are hosted by other clients which generate, maintain and upload catalogue information. Hence data of remote non-synchronised repositories is not stored in Cloud storage it is available on demand only.

A client's system works with both types of repositories at the same time. As outlined below, the concept is implemented in a client application which serves two purposes.

When working with a synchronised repository, the user's interaction with the application is minimised following initial setup. The user can continue working with his system and accessing files and folders as before. This therefore reduces initial learning effort [MP97]. In-depth interviews with users representing potential groups of users showed that a Cloud storage system should have minimal influence on the daily routine of user interaction. Such systems should therefore not require the user to i.e. change his custom workflow or how he structures his files and folders. These findings are also supported by [LED+ 99].

When working with a remote non-synchronised repository, the client application is used in order to browse, search or download files and folders. In this case the user is required to use the client application. During the in-depth interviews, the interviewees understood the application as a tool for browsing contents of remote systems and expressed the need for an interaction with this storage distinguishable from their files and folders stored locally. For this reason remote non-synchronised repositories are not i.e. mapped into the user's local file system.

3.2 Interaction Design

During the initial setup of either a synchronised or a remote non-synchronised repository, the user selects a directory of his choosing from his local system which serves as repository root folder. In-depth interview findings support the need for flexibility. Most of the systems we have referenced before work with one dedicated root

[4] This concept focuses on the technical sub-system whereas collaboration features can be augmented to extend functionality in the social sub-system [Her09] to serve i.e. as group communication tool reflecting social relationships of the user.

folder forcing the user to move his data into this particular directory whereas our concept allows the selection of any folder[5]. The selection of a directory as repository is indicated by a different folder icon. In-depth interviews prove this to be the most lucid way of displaying the folder's status.

After the initial setup the client application generates and submits a catalogue to the Catalogue Server. Depending on the type of the repository, data is transferred and stored in the Cloud storage, as outlined above.

3.3 Architecture

In order to keep track of the contents of and changes to a repository, the catalogue of a repository needs to be updated in the event of the creation of, changes to, or the deletion of a file. The client application monitors file events, updates the local catalogue and submits catalogue changes to the Catalogue Server which distributes change notifications to other client systems also connected to the respective repository.

With regard to remote non-synchronised repositories, data availability depends on the connection state of the client hosting the remote non-synchronised repository. As the catalogue of a remote non-synchronised repository is stored on a Catalogue Server within the Cloud, an authorised client may browse and search the repository independently of the connection state of the client hosting the data without having to download all its content.

Figure 2 outlines the retrieval protocol for a repository request. Depending on the type of the repository, the data resides either in Cloud storage or on a client's system. Upon a request, the location of the requested data is resolved. Based on the assumption that Cloud storage serves high availability data stored in the Cloud is available directly on request. Cloud storage locations are managed by the Catalogue Server which holds an index and tracks usage quotas of possible storage locations as i.e. local server storage, FTP locations or commercially available Cloud storage locations.

If the request was directed to a file of a remote non-synchronised repository, through catalogue information one or more clients hosting the repository are located. If no system hosting the repository is available through an active Internet connection the request is cached until a host is available, as Figure 2 illustrates. Once one or more clients are available, data is directly transferred to the requesting client system via a peer-to-peer connection. If the requesting client is unavailable in the meantime, the file is cached inside the Cloud and will be downloaded when the client is back online.

The actual concept relies on Cloud storage providing fast and highly available storage space. In the future a peer-to-peer based infrastructure serving distributed storage will supersede today's Cloud storage which not only faces legal regulations but governmental encroachments being the reason for companies as well as individuals not to rely on Cloud storage solutions [JLGS09]. With the Catalogue Server, the outlined concept only

[5] Excluding system folders and folders which are already part of a repository.

stores information on files but not file contents which will remain on the client system only.

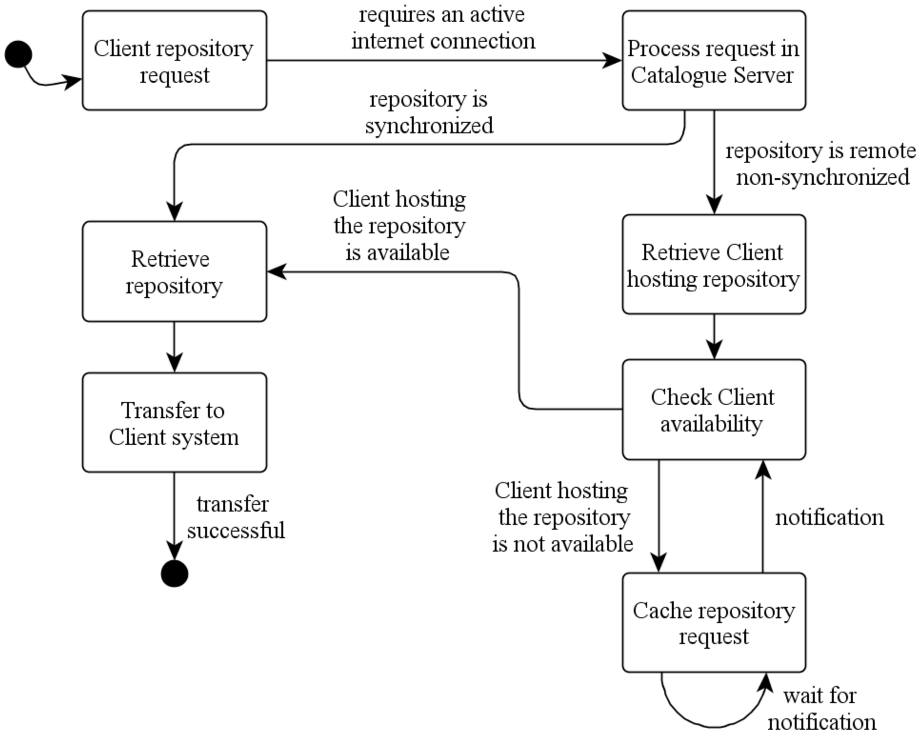
Client repository request
requires an active internet connection
Process request in Catalogue Server
repository is synchronized
repository is remote non-synchronized
Client hosting the repository is available
Retrieve repository
Retrieve Client hosting repository
Transfer to Client system
Check Client availability
transfer successful
Client hosting the repository is not available
notification
Cache repository request
wait for notification

Figure 2: The chart illustrates the retrieval protocol for a repository request. A client repository request is submitted to a Catalogue Server inside the Cloud. The Catalogue Server resolves the repository type and checks the repository's availability. A remote non-synchronised repository is not available from Cloud storage and transferred as long as the Client hosting the respective repository is connected to the Internet.

3.4 Implementation

Our first prototype consists of a client application and a standalone server system written in .NET 3.5. Using MONO[6] allows cross platform deployment of the client application.

The client application was formatively evaluated during in-depth interviews throughout the process of implementation by using a paper prototype as outlined above. In addition, a final summative evaluation was carried out as well. The first component of the client application is a service which runs in the background monitoring the repositories handling catalogue generation and actualisation as well as catalogue and file transfers.

[6] http://www.mono-project.com /

The second component consists of a graphical user interface. When working with synchronised repositories there is no need to use a client application. The user can continue working with his files and folders as he is accustomed to using his preferred file browser. However data of remote non-synchronized repositories is not stored on every client system. To browse the catalogue and thereby the repository, the user is required to use the client application which allows for files to be found or requested for download if the respective host is unavailable and downloaded directly if the host is available. This allows for a lazy and incomplete synchronisation with several locations without having the need for explicit synchronisation points.

The server has been implemented serving a RESTful database-assisted API [FT02], [Rod08] not only offering a stable and flexible service-based infrastructure but an interface for future client development for i.e. mobile devices.

4 Summary and Conclusions

The concept outlines a catalogue-assisted multi-user file storage and collaboration system which is designed as a hybrid between a highly available but from the user's perspective in-transparent Cloud storage solution and a considerable more transparent but possibly less available access to local storage in a peer-to-peer infrastructure.

The surveys conducted reveal a general interest in Cloud storage and furthermore an active demand for storage solutions. The in-depth interviews show the innovative concept presented within this contribution to be understandable, usable and acceptable. A world-wide trend towards broadband connections supports the findings of storage space distributed throughout client systems to supplement highly available Cloud storage.

In context with the scenario presented in this article, various use-cases exist supporting possible applications for this hybrid storage system. Based on our prototypical implementation and our first empirical findings, a long term evaluation is under preparation. Future ideas reflect upon both architectural as well as client-side improvements. For instance improving file search by using the meta-data of files in a faceted context [YSLH03] may prove time-saving and more convenient for the user.

Protecting the user's privacy and data against access by any third party is essential. While the term privacy is ambiguous and can be used in different contexts, we consider the user's personal data to be sensitive. Nissenbaum identifies three concerns towards privacy: (1) monitoring and tracking, (2) dissemination and publication, and (3) aggregation and analysis [Nis09]. These challenges have to be taken into account when designing Cloud-based storage solutions.

Our approach allows the user to decide which data to upload to Cloud-based servers, making the data more vulnerable to be exposed to third parties, and which data to keep on computing devices under his control, better protecting it. The user is made aware of this difference by the need to use the specific client for remote non-synchronised repositories.

For the time being, data of remote non-synchronised repositories is still cached in the Cloud when transferred to another client. This will not be necessary once we move to a full peer-to-peer solution for data exchange. The catalogue data would be the only data still vulnerable to attacks. One could distribute the catalogue data between clients as well, but there is no guarantee that the information is up to date at all times. This is a trade-off worth investigating further.

Acknowledgments

The authors express their thanks to both Syavoosh Khabbazzadeh and Björn Eberhardt for providing data surveying Cloud storage in different contexts.

References

[AFG+09] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[AIM12] AIMC. Navegantes en la Red – 14a encuesta AIMC a usarios de Internet. Technical report, Association of Communication Media Research, February 2012.

[BS11] David C.E. Bouck-Standen. The Davebox – A Cloud-Based File Repository System. Bachelor thesis, Institute for Multimedia and Interactive Systems, University of Luebeck, November 2011.

[CKS09] Christian Cachin, Idit Keidar, and Alexander Shraer. Trusting the cloud. SIGACT News, 40:81–86, June 2009.

[CRC07] Alan Cooper, Robert Reimann, and David Cronin. About Face 3: The Essentials of Interaction Design. Wiley Publishing, Indianapolis, IN, USA, 3rd edition, 2007.

[Ebe11] Björn Eberhardt. Augmented Reality an gedruckten Dokumenten. Bachelor thesis, Institute for Multimedia and Interactive Systems, University of Luebeck, 2011.

[FT02] Roy T. Fielding and Richard N. Taylor. Principled design of the modern Web architecture. ACM Trans. Internet Technol., 2:115–150, May 2002.

[Her09] Michael Herczeg. Software-Ergonomie. Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme. Lehrbuchreihe interaktive Medien. Oldenbourg, München, 3., vollst. überarb. und erw. Aufl., 2009.

[JLGS09] Paul T. Jaeger, Jimmy Lin, Justin M. Grimes, and Shannon N. Simmons. Where is the cloud? Geography, economics, environment, and jurisdiction in cloud computing. First Monday, 14(5), 2009.

[Kha11] Syavoosh Khabbazzadeh. Entwicklung eines NEMO-Clients für mobile Geräte mit Android-Betriebssystem. Bachelor thesis, Institute for Multimedia and Interactive Systems, University of Luebeck, 2011.

[Kro12] Neelie Kroes. Towards a European Cloud Computing Strategy. Speech, World Economic Forum, Davos, February 2012. Neelie Kroes is the Vice-President of the European Commission responsible for the Digital Agenda.

[LCHS10] Sebastian Lob, Jörg Cassens, Michael Herczeg, and Jan Stoddart. NEMO – The Network Environment for Multimedia Objects. In Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia – IITM 2010, pages 245–249, New York, NY, USA, December 2010. ACM.

[LED+99] Anthony Lamarca, W. Keith Edwards, Paul Dourish, John Lamping, Ian Smith, and Jim Thornton. Taking the Work out of Workflow: Mechanisms for Document-Centered Collaboration. In Proceedings of the European Conf. Computer-Supported Cooperative Work ECSCW '99, pages 1–20. Kluwer, 1999.

[MG09] Peter Mell and Tim Grance. The NIST Definition of Cloud Computing. National Institute of Standards and Technology, 53(6):50, 2009.

[MP97] Gloria Mark and Wolfgang Prinz. What Happened to our Document in the Shared Workspace? The Need for Groupware Conventions. In Human-Computer Interaction INTERACT '97, pages 412–420. Hall, 1997.

[Nis09] Helen Nissenbaum. Privacy in Context: Technology, Policy, and the Integrity of Social Life. Stanford University Press, 2009.

[Ofc11] Ofcom. International Communications Market Report 2011. Technical report, Ofcom, December 2011.

[Rod08] Alex Rodriguez. RESTful Web services: The basics. Technical report, IBM developerWorks, New Orchard Road, Armonk, New York 10504, USA, 2008.

[SL10] Heidi Seybert and Anna Lööf. Internet Usage Statistics 2010, eurostat, 2010.

[STG03] Reinhard Sefelin, Manfred Tscheligi, and Verena Giller. Paper prototyping - what is it good for?: a comparison of paper- and computer-based low-fidelity prototyping. In CHI '03 extended abstracts on Human factors in computing systems, CHI EA '03, pages 778–779, New York, NY, USA, 2003. ACM.

[Sto10] Jan Stoddart. Entwicklung einer Serverkomponente für das ambiente System NEMO. Master‘s thesis, Institute for Multimedia and Interactive Systems, University of Luebeck, 2010.

[TBBS06a] Maryam Tohidi, William Buxton, Ronald Baecker, and Abigail Sellen. Getting the right design and the design right. In Proceedings of the SIGCHI conference on Human Factors in computing systems, CHI '06, pages 1243–1252, New York, NY, USA, 2006. ACM.

[TBBS06b] Maryam Tohidi, William Buxton, Ronald Baecker, and Abigail Sellen. User sketches: a quick, inexpensive, and effective way to elicit more reflective user feedback. In Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles, NordiCHI '06, pages 105–114, New York, NY, USA, 2006. ACM.

[YSLH03] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '03, pages 401–408, New York, NY, USA, 2003. ACM.