

# Cooperation across Multiple Healthcare Clinics on the Cloud<sup>\*</sup>

Neil Donnelly<sup>1</sup>, Kate Irving<sup>2</sup>, and Mark Roantree<sup>3</sup>

<sup>1</sup> School of Computing, Dublin City University

<sup>2</sup> School of Nursing and Human Sciences, Dublin City University

<sup>3</sup> Insight: Centre for Data Analytics, Dublin City University,  
Glasnevin, Dublin 9, Ireland

**Abstract.** Many healthcare units are creating cloud strategies and migration plans in order to exploit the benefits of cloud based computing. This generally involves collaboration between healthcare specialists and data management researchers to create a new wave of healthcare technology and services. However, in many cases the technology pioneers are ahead of government policies as cloud based storage of healthcare data is not yet permissible in many jurisdictions. One approach is to store anonymised data on the cloud and maintain all identifying data locally. At login time, a simple protocol can be developed to allow clinicians to combine both sets of data for selected patients for the current session. However, the management of off-cloud identifying data requires a framework to ensure sharing and availability of data within clinics and the ability to share data between users in remote clinics. In this paper, we introduce the PACE healthcare architecture which uses a combination of Cloud and Peer-to-Peer technologies to model healthcare units or clinics where off-cloud data is accessible to all, and where exchange of data between remote healthcare units is also facilitated.

## 1 Introduction

Healthcare operations in many countries are examining cloud-migration plans in order to exploit the benefits of cloud based computing. While standards such as HL7 [9] sought to create a common healthcare record, the storage and management of these records using a cloud-based strategy is now in focus, including in the area of dementia. Dementia is a serious loss of cognitive ability beyond what might be expected from normal ageing. While dementia is a chronic and progressive illness with no known cure, there is now strong evidence that dementia can potentially be delayed by adopting midlife lifestyle changes aimed at improving areas such as cardiovascular health, poor diet and cognitive activity. Given the huge social and economic costs of dementia, even a delay of one year would make such interventions cost-effective [3]. The Dementia Elevator project represents a

---

<sup>\*</sup> This work is part of the Dementia Elevator Project and funded by HSE Ireland and Atlantic Philanthropies.

collaboration with experts in dementia, which is seeking to build a new cloud-based data management system for dementia patients. Elevator is funded by the Irish healthcare system (the HSE) and The Atlantic Philanthropies.

Healthcare systems and research projects are numerous. In recent years, the trend has been towards personal health based systems where sensor devices can capture and monitor heart rate [11] with systems that generate large volumes of data. These types of systems were also used in projects looking at sports and performance data in personal health systems [12]. Both of these types of projects produce large sets of data that can be easily managed in cloud-based architectures. However, there are problems associated with using purely cloud-based systems, as summarised in [13]. Trust was one of the biggest issues listed. Users and governments are reluctant trust that healthcare records are secure on the Cloud. Our approach is to use cloud-based technology to store anonymised healthcare data in while using a P2P approach to facilitate the storage and sharing of personal patient data between users. While this provides a level of security on top of the cloud, we do not focus specifically on security and assume that approaches such as [6] can be used in conjunction with the PACE system.

**Paper Structure.** In §2, we introduce the PACE system and provide a brief overview of components; in §3, we provide a detailed discussion of the peer-based setup and how communication is provided; in §4, our evaluation is presented; in §5, a discussion of the similar healthcare approaches is provided; and finally in §6, we provide some conclusions.

## 2 The PACE System and Clients

The PACE (Peer-to-peer Architecture for Cloud based EHealth) system uses modern web technologies and a decentralised hybrid P2P topology, as described in [1], to enable the sharing of confidential patient data between healthcare professionals. The cloud portion of the system maintains the anonymised patient data on a SQL database and manages the super-peers. Each PACE user acts as a peer, is connected to a super-peer and can transfer patient data with other peers over a P2P connection. Figure 1 demonstrates the design of the PACE system, which has four main components (P1 - P4) which will now be described.

**Client Interface (P1).** The Client Interface component is the primary interface between the PACE system and the user. There is a basic set of functionality at this level and these are mostly mapped to functions in the backend processes.

- **Authenticate.** This function authorises users signing in and also informs the super-peer that a peer has logged on and is available to share data.
- **Search.** The search function is the first step in the P2P sharing protocol described in Section 3. The function uses the Clinic Connector (P2) to locate a patient’s private data using an ID and initiate a P2P connection.

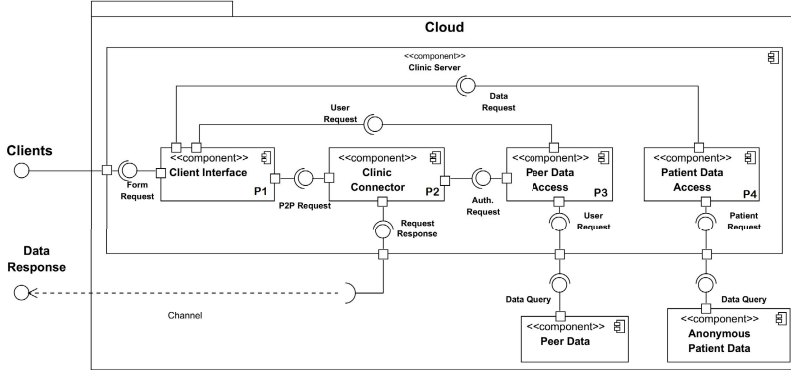


Fig. 1. PACE Components and Interaction

- **Retrieve.** This function uses the Patient Data Access (P3) component to find and retrieve patient data stored on the cloud using a patient ID.
- **Update.** The Update function uses the Patient Data Access (P4) to either update or add a patient’s record on the cloud. When a new patient is added the Client Interface uses P2 and P3 to synchronise data between peers.

**Peer Data Access (P3).** The Peer Data Access component is responsible for the overlay network of the peers and super-peers. It manages the data on the users (peers) and clinics (super-peers). This is described in detail in Section 3. Each super-peer manages a list of peers; the users of the system. In typical super-peer systems [2], peers may join or leave the network at any time while super-peers are responsible for managing their own grouping and for communicating with other super-peers.

**Patient Data Access (P4).** The goal for the Patient Data Access component is to manage all the anonymised patient data stored on the cloud. All functions are called by related functions in P1.

- **Patient\_Search.** This function locates specific patients based on a unique identifier and optionally a clinic identifier to reduce the dataset to search.
- **Query\_Search.** This function provides the ability to use SQL-like queries to retrieve multiple patient records matching criteria specified in the query.
- **Update.** This function updates a patient in the database, or if the patient does not currently exist, it adds the the new patient’s records.

**Clinic Connector (P2).** The Clinic Connector service locates and connects peers. It acts as a handshake mechanism that introduces clients in order to initiate a transient P2P connection. This has two scenarios: when a peer has been found to be missing patient information; and when a user requests data from another clinic. If a peer is missing information from the clinic, the Clinic

Connector simply introduces the peer to the clinic's other peers. If the clinician makes a request for a patient from another clinic, the Clinic Connector instead communicates with other super-peers to locate the data. It does with the help of the Peer Data Access (P3) service.

### 3 Cooperation across Super-Peers

As users and clinics are modelled as peers and super-peers respectively, this section begins with a description of our model constructs, followed by a description of how data is found and shared. As private patient data cannot be stored on the cloud and must remain in local storage, sharing data across clinics becomes an important goal. This section explains in detail how this is achieved.

#### 3.1 PACE Model Constructs

With peers randomly connecting to the system and individually creating data, all peers (end-users) are organised into peer groups (the clinics). As each user of the system belongs to a single clinic, peers were organised within a single group, managed by the super-peer. The following are the constructs of the PACE system in terms of their attributes and functionality.

- **Peer**. The attributes of the Peer are its ID, Peer Context, and the super-peer group in which it belongs. Its functions include: `getPatient()` which sends a request to the super-peer for a particular patient's data; `Connect(Peer)` creates a P2P connection with another peer; `sendPatient()` which sends requested patient data across an established P2P connection.
- **Peer Context**. This construct models the user of the system. It contains the name of the user and the list of private patient data it has stored.
- **Super-peer**. The construct attributes are a Super-peer Context and a list of clinic's peers. The functions of the super-peer are: `introduce(Peer, Peer)`, which begins the protocol needed to introduce two peers to set up a P2P connection; `peerConnect()` requests information from a peer; and `spConnect()` creates a connection to another super peer.
- **Super-peer Context**. This construct represents a clinic with attributes such as the clinic's name.

#### 3.2 Communication Method

We now describe the five steps involved in peers sharing data.

1. Patient Data Request. The PACE client sends a patient request using the peer's `getPatient` function.
2. Patient Request. The client interface forwards this to the Clinic Connector.
3. Data Location Request. The Clinic Connector requests a peer holding this information from the Peer Data Access using `spConnect`.

4. Return Data Location. The super-peer returns the a peer holding the data.
5. Introduce Peers. The Clinic Connector can now attempt to introduce the requestor to the prospective host:
  - (a) The Clinic connector sends a request to the host peer using `peerConnect`.
  - (b) Hosting peer confirms it has the requested patient data.
  - (c) Clinic Connector sends hosts addressing information to requestor.
6. P2P Data Request. The requesting peer establishes a P2P connection with the host using the `Connect` function with the received address.
7. Private Patient Data Response. The host of the data uses `sendPatient` to transfer data to the requesting peer.

## 4 Experiments and Evaluation

The PACE server was run using Google’s App Engine [7] platform. The machines ranged from a 3.4GHz Intel i7 with 8GB of RAM to a 3.2GHz Intel i5 with 8GB of RAM. All machines had the Windows 7 operating system. Each machine ran the client on Google’s Chrome browser and acted as a peer for the experiment.

For the experiment, we encapsulated the super-peers within the cloud. This design ensured the availability of super-peers for the evaluation. The super-peers operated as autonomous peers in the overlay network but were provided on a cloud platform. For this experiment, two clinics and six peers were used.

The experiment aimed to demonstrate that the PACE system could successfully manage multiple healthcare workers across different clinics and facilitate data sharing between peers. In order to do this, we needed to prove: patient data could be shared over a P2P connection; a peer’s patient records is updated on sign in; and every peer is updated when a new patient is added. The patient data to be kept by the peers included four Strings representing a name, an address, a phone number and a nine character ID. While the data stored online was a series of random boolean values representing medical records.

Once a peer entered the system, the supper-peer would update it’s patient records by introducing it to other peers. A user could also search for a patient using a patient ID. The super-peer can then locate a peer that holds this data and introduce the peers. For our evaluation, we sought to test adding patients; updating patients; accessing peer data; and synchronisation. As timing for functions such as synchronisation was important, we set timers on all operations.

Time to Upload 10 Patients:	7490ms
Time to Upload 50 Patients:	36123ms
Time to Synchronise 10 Patients:	2876ms
Time to Synchronise 50 Patients:	12407ms
Time to Find and Receive a Patient:	478ms

**Fig. 2.** PACE Experiment Results

During the evaluation, each peer could successfully add multiple patients to PACE's cloud storage and retrieve private patient data from other peers. When a user logged in, their local records were automatically updated. The average time was calculated for adding sets of 10 and 50 patients, which can be seen in figure 2. Figure 2 also shows that the time taken to synchronise data is far less than the insertion time. This difference is due to the slow insertion time with cloud's datastore and the peers being on the same network during the test.

## 5 Related Research

The authors in [10] present a method of protecting data from certain parties while making it available to others. In their paper, the authors suggest dividing patient health records into domains, with certain information available to some domains but not others. In one of the paper's examples, health insurance companies may see important details of the patient's personal information, but other data is kept private to patient. The PACE system uses a similar domain structure but between clinics. However, PACE builds upon this idea by introducing a physical distance between the information and leaving the personal info in the hands of only those who need it.

In [4], the authors used a hybrid cloud system to share data between hospitals and 3rd party auditors. Data would be held on both a private and public cloud. The private cloud keeps information secure and within the hospital while the public cloud made certain data available to others when needed. The PACE system also uses a public cloud to make certain data available to users. However instead of trying to maintain a private cloud, the PACE system uses P2P networks which distributes storage and sharing duties which can help efficiency.

The authors of [5] present a P2P based healthcare system using JXTA. JXTA is a Peer-to-Peer development infrastructure allowing Java applications to connect as peers. Much like our PACE system, in the paper, they suggest using peer groups to store patient data that can then be shared by other peers, with peer groups organised based on the wards. However, for the PACE system we implemented a hybrid decentralised P2P topology with the cloud acting as an organising server. This allows for greater efficiency when locating information stored by the peers. Using the cloud as a central server also allows peers to share data between different peer groups.

## 6 Conclusions

In this paper, we introduced the PACE system, a healthcare architecture using both Cloud and P2P technologies to facilitate autonomy within clinics and users, while at the same time, creating an environment for sharing healthcare records. Healthcare workers are modelled as peers and their department or clinic is modelled as the super-peer, with the P2P overlay managed by the Cloud. Thus, we exploit cloud technology for the benefits of storage, elasticity and access to anonymised data, while the identifying data can reside on individual devices.

We collaborated with a number of dementia researchers to evaluate PACE, determine what types of queries were possible and how they were expressed, and used the PACE prototype (described in Section 4), for testing purposes.

In Ireland, it is not permissible to store healthcare data on the Cloud and thus, a hybrid application was necessary. The novelty in this system was the combined usage of cloud and P2P technologies to enable the types of sharing necessary across healthcare departments where user's devices are connected and disconnected in arbitrary fashion. Our evaluation was on a proof-of-concept basis, where we tested availability of peers, updating on-cloud data and the synchronisation process for peers. Insert and Update queries used an SQL-type language which was converted into low level PACE functions for execution.

## References

1. Androutsellis-Theotokis, S., Spinellis, D.: A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys* 36(4) (2005)
2. Bellahsene, Z., Roantree, M.: Querying Distributed Data in a Super-Peer Based Architecture. In: Galindo, F., Takizawa, M., Traummüller, R. (eds.) *DEXA 2004. LNCS*, vol. 3180, pp. 296–305. Springer, Heidelberg (2004)
3. Comas-Herrera, A., Northey, S., Wittenburg, R., Knapp, M., Bhattacharyya, S., Burns, A.: Future costs of dementia-related long-term care: exploring future scenarios. *International Psychogeriatrics* 23, 20–30 (2011)
4. Chen, Y.-Y., Jan, J.-K., Lu, J.-C.: A Secure EHR System Based on Hybrid Clouds. *Journal of Medical Systems* 36(5) (2012)
5. Lim, B., Choi, K., Shin, D.: A JXTA-based Architecture for Efficient and Adaptive Healthcare Services. In: Kim, C. (ed.) *ICOIN 2005. LNCS*, vol. 3391, pp. 776–785. Springer, Heidelberg (2005)
6. Clarke, A., Steele, R.: Summarized Data to Achieve Population-Wide Anonymized Wellness Measures. In: *Engineering in Medicine and Biology Society (EMBC)*, pp. 2158–2161. IEEE Press (2012)
7. Google App Engine, <https://developers.google.com/appengine/>
8. Halevy, A., Ives, Z.G., Madhavan, J., Mork, P., Suci, D., Tatarinov, I.: The Piazza Peer Data Management System. *IEEE Trans. Knowl. Data Eng.* 16(7), 787–798 (2004)
9. Health Level Seven International, HL7 (1987), <http://www.hl7.org/>
10. Löhr, H., Sadeghi, A.-R., Winandy, M.: Securing the E-Health Cloud. In: *IHI ACM International Health Informatics Symposium* (2010)
11. Roantree, M., McCann, D., Moyna, N.: Integrating Sensor Streams in pHealth Networks. In: *Proc. of 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 320–327. IEEE Computer Society Press (2008)
12. Roantree, M., Shi, J., Cappellari, P., O'Connor, M., Whelan, M., Moyna, M.: Data transformation and query management in personal health sensor networks. *Journal of Network and Computer Applications* 35(4), 1191–1202 (2012)
13. Spil, T., Klein, R.: Personal Health Records Success; Why Google Health failed and what does that mean for Microsoft HealthVault? In: *Proceedings of HICSS-47 (Health Informatics Track)*, pp. 2818–2827. IEEE Computer Society Press (2014)
14. Yang, B., Garcia-Molina, H.: Designing a Super-Peer Network. In: *Proceedings of the 19th International Conference on Data Engineering*, pp. 49–60. IEEE Computer Society Press (2003)