Univeristy of California, Berkeley

Master's Final Project Report

# MemoryKloud: Preserving Cherished Memories Shared Online

*Author:*
Rami Taibah
Jie Wu

*Supervisor:*
Dr. Ray Larson

*A final report submitted in fulfilment of the requirements for the degree of Masters of Information Management & Systems*

*in the*

School of Information

May 2012


THE UNIVERSITY OF CALIFORNIA
LET THERE BE LIGHT
1868

*"A good creative team beats the lone genius every single time."*

Laurie MacPherson

UNIVERSITY OF CALIFORNIA, BERKELEY

# *Abstract*

School of Information

Masters of Information Management & Systems

**MemoryKloud: Preserving Cherished Memories Shared Online**

by Rami Taibah
Jie Wu

Most social media platforms such as Facebook, Twitter, Foursquare enable us to share content in a status-update fashion, or what we might call byte-size formats. This gives us up-to-date information about our friends and family informing us on what is going on with their lives. However, there are two major problems with this format. First, status-update streams come in from different sources, all of which compete for user attention. Users may find it challenging to stay on top of a particular event. Also, status updates usually come in as separate pieces of information, which users may find it difficult to differentiate their relevance and/or importance. Both of these problems cause fragmentation of the narrative, and ultimately the bigger picture of the story gets lost. This makes the Internet a one-way street. People push their precious memories to the Internet with the goal of sharing them with people, but there is no simple way to retrieve the memories back again.

These two problems are partly introduced by social media platforms because they are designed around individuals. While most memories are experienced with our family and friends, the mediums that these memories are shared on focus on the person not the event. The solution we envision, is to flip the paradigm on its head; a social network with events in its center and individuals feeding into it, instead of an individual feeding out their own memories.

An event-centric social network can be achieved with a suite of solutions called MemoryKloud. MemoryKloud consists of a mobile application, which would sit in between users and social media platforms to record interesting moments based on events, and a web application that could potentially pull users updates from various social platforms and auto-classify them into relevant events.

We believe MemoryKloud enhances users experience for memory preservation. On one hand, precious moments in our lives are kept in a digital format where they are easily accessible through different devices; on the other, social interactions for a particular event is enhanced by sharing, thus making memory preservation easier and more fun.

*Keywords: moment, memory, sharing, nostalgia*

# *Acknowledgements*

First and foremost we would like to thank our parents for always being there for all of our endeavour. Dr. Ray Larson who has been instrumental in many of the decisions we arrived to. We would also like to thank all the people who helped & supported us: May Abu-Ghazalah, Ali Al-Najjar, Adiyta Chand, Kari McGlynn, Sara Connor, Suhani N Mehta, Tapan Parikh...

# Contents

# List of Figures

# 1

## 1.1 The Problem with Online Memories

As a team, we know that our interests converge in terms of memory preservation. And given the fact that this is a rather wide topic, we agreed on a particular fictitious user story to specify our problem.Imagine you attended your best friend's wedding (let's call her Jane.) Roughly two hundred guests showed up at the wedding; beautiful flowers, lovely music, happy faces all around. You took many pictures, so did the all the other guests. At the end of the day, you went back home and posted these pictures on your favorite social media, say Facebook. You are certain that others uploaded their own set of "memories" somewhere else. If you are friends with them on that network, you might be able to see their "memories," but if you are not, tough luck.There are two main problems in this scenario. First, different people post different memories on different web platforms. How can Jane stay on top of all these memories? She might be interested in curating all these memories, but how can she find all these memories and "capture" them for her own keepsakes? We call this the "attention problem." Also, Jane's wedding memories might be posted next to news of an awful catastrophe happening somewhere in the world, or above a rather distasteful Internet meme. Jane's memories will also soon be buried by deluge of other posts from other people. This would further occult the larger picture of Jane's wedding. How can the relevant posts be put together? We call this the "relevance problem." Both of these problems can potentially result in losing precious moments captured in a digital form. When people are interested in looking back and attempt to reminisce over special memories with special ones, they might regret that there is nothing (or too much) to resonate.

## 1.2 Competitor Analysis

Looking at the current tech & Internet industry landscape, many companies are trying to solve both of the attention and relevance problems we defined. News, shopping, social media, entertainment industries and many others are in desperate need for solutions to these problems. There are many companies that are trying to solve the memory preservation problem. Here are some that we identified:

### 1.2.1 Facebook

As of April 2012, Facebook has 901 million active users, who share their most cherished moments on a daily basis. Even though Facebook seems to try to be everything and everyone at the same time, they are too big to ignore. Recently, Facebook launched "Facebook Timeline," which tries to preserve memories on an individual level. Timeline was inspired and designed by Nick Feltron, who is known for his personal annual reports, where he documents seemingly banal events of his life and compiles stunningly amazing graphical reports from them. While Feltron's work is impressive, Facebook's decision to build Timeline using the Feltron reports as a framework only entrenches the problem. Remember, memories are experienced with others and not alone.

### 1.2.2 Path

Path is a mobile social network that is aimed to share photos, video, and location with close friends and family. In a sense it is trying to address the problems introduced by the Facebook platform. Unlike Facebook, users are expected to share only their experiences and what they are currently doing. Path is a great tool for personal memory preservation, but again it puts the individual in the center of things, whereas events should be at the center.

### 1.2.3 Popset

Popset is an iPhone app that bills itself as a group photo solution. Users and their friends can contribute images and share them in photo albums. Popset is the closest to what MemoryKloud is trying to do. It puts the event at the core, then asks people experiencing the event to post images related to it. However, Popset is marketing itself as a "group photo" solution, not a memory preservation solution. As a result, the solution seems lacking because it fails to capture anything but images.

## 1.3 Problem Validation & Insight Generation

Inspired by Clark Kellogg's Problem Finding and Problem Solving (PFPS) Process (Figure 1.1), we moved on to observe potential users and attempt to validate the problem. We employed master-apprentice contextual interview methods to validate the problem we defined, then created an affinity diagram to generate insights from users.

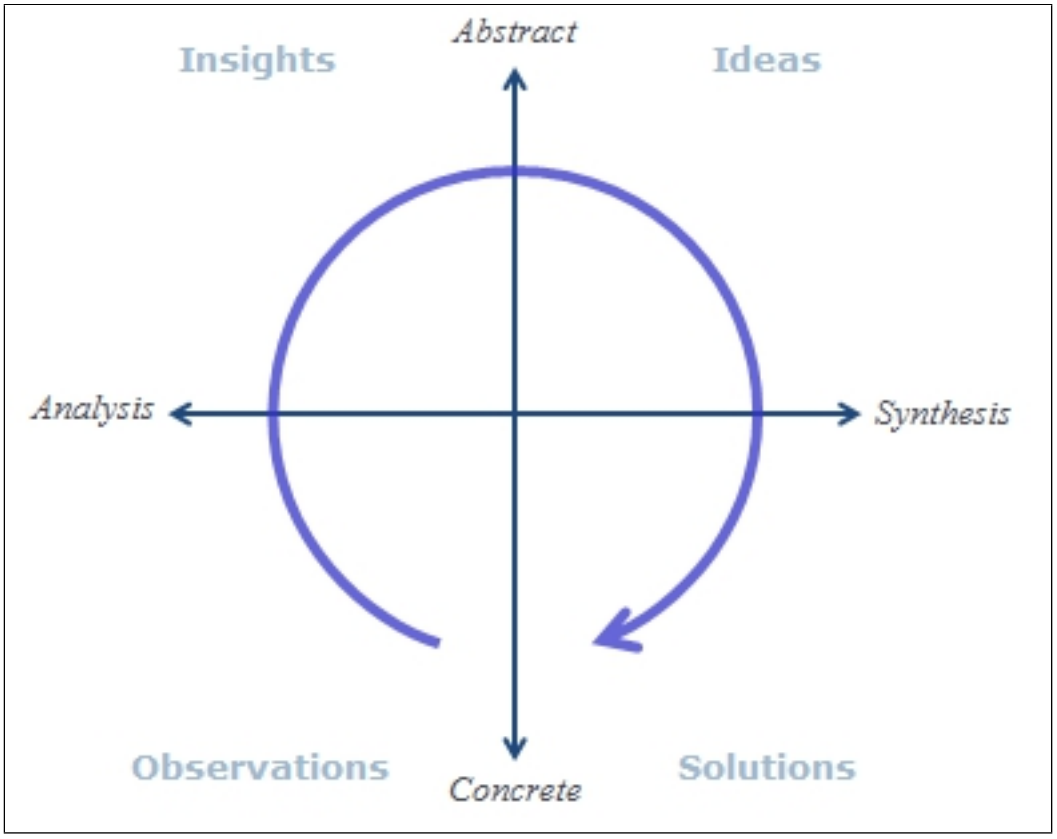


FIGURE 1.1: Problem Finding and Problem Solving (PFPS) Process

In total, we conducted five contextual interviews where we sat down with our interviewees, observed how they curated their memory artifacts: diaries, pictures, videos, etc. In the meantime, we asked them three types of questions: their understanding about memory preservation, their pain points in managing their digital memory artifacts, and their social sharing habits. We gathered their answers and our observations throughout the process and wrote them down on post-it notes. We then proceeded to create an affinity diagram


FIGURE 1.2: Affinity Diagram

This exercise allowed us to generalize our findings and helped us generate major insights

- First of all, when talking about their understanding about memories, most people mentioned the word "personal." Someone commented: "memories have a personal touch," while another lady said: "memory is about documenting your life, such as taking pictures, writing and talking."

- Secondly, people say they only take pictures when they are "special," "out-of-blue," or "fun," which resonated with the characteristics of memories. Also, photos seem like the main medium for memory preservation since few people mentioned taking videos or using videos to preserve precious moments. In that sense, we might be able to approximate picture preservation as memory preservation. That's why we ask people what digital tools they use for memory/picture preservation and sharing. The most frequent answers we got are Facebook, Instagram and Flickr. We picked Facebook to take the questions further

- When we asked interviewees about their Facebook usage for memory preservation, some commented: "Facebook is certainly not the best way to preserve memory; it does so many things." or "I only use Facebook to stay in touch with friends whom

I don't usually have chance to meet often." When we asked their general Facebook experience, we got surprising negative feedback: "Picture quality on Facebook is really bad, I would prefer Flickr"; "I found Facebook really creepy when it showed me a map with all the places I checked in." At least two people had commented that Facebook is not UI friendly and hated the timeline feature.

In addition, we confirmed our initial assumption that the "attention problem" was an issue, since more than one person said something like: "there is too much going on in Facebook, I use Facebook to stay touch with friends, and I don't usually pay attention to what happened one or two days ago." We didn't directly ask about the "relevance problem" since it is quite apparent that Facebook just lists updates based on time regardless of their relevance.

Through our affinity diagram analysis, we extracted three personas, each of which exhibits distinct characteristics and personalities. Their names: Jennie Marcus, Brett Sanderburg, and Victor Fuenes (Figure 1.3).

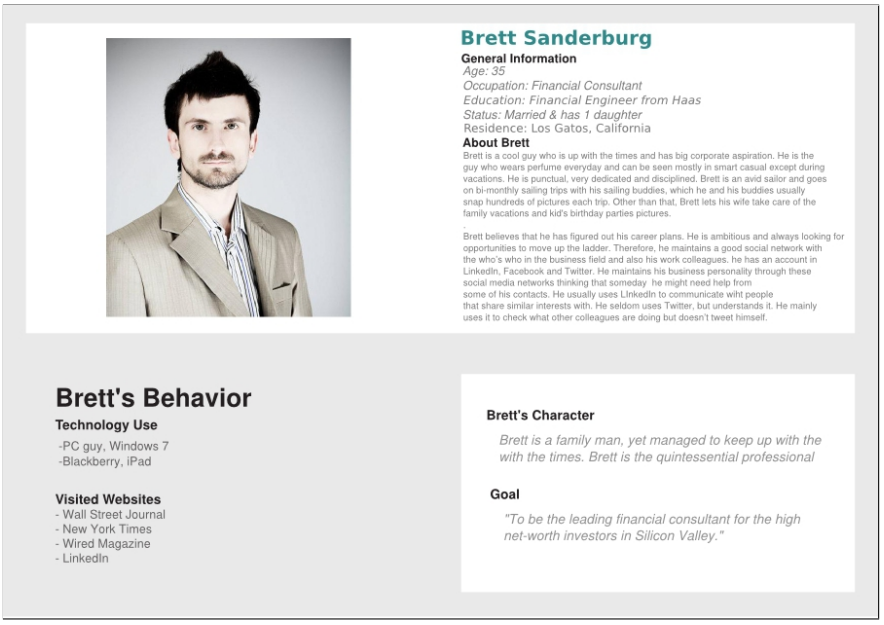


FIGURE 1.3: Brett Sanderburg

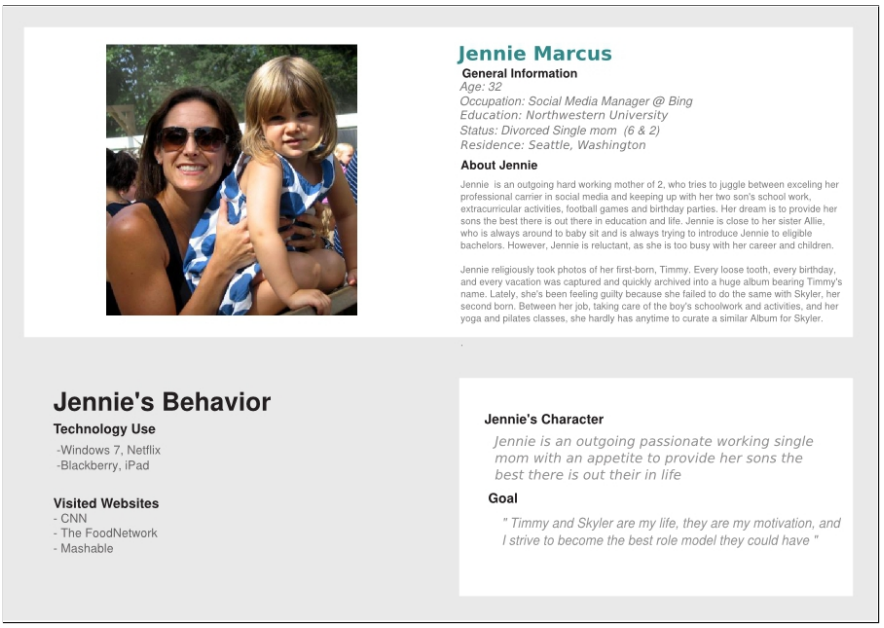


FIGURE 1.4: Jennie Marcus

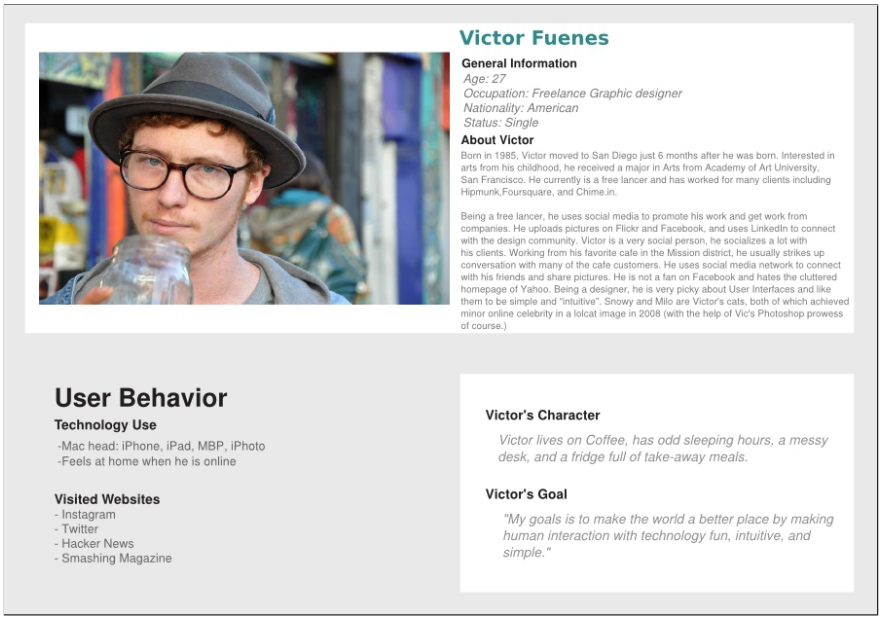


FIGURE 1.5: Victor Fuenes Persona

After a long and intensive discussion, we agreed that Jennie and Brett are our main personas. This decision was based on the fact that Jennie and Brett tended to have more special occasions to capture (Jennie has two kids, and Brett travels a lot.) Also, because both had very busy schedules, they would be looking for solutions that made the task of memory preservation less of a chore.

### 1.3.1 Ideation & Solutions Generation

The next stage in PFPS process is to generate ideas from insights. The contextual interviews suggest that memory preservation and sharing is mostly conducted through Facebook. However, the interviews also revealed that Facebook is not the ideal method for memory preservation and that there is a real pain point with the memories shared on the platform.

With these insights and the original problems in mind, we came up with two ideas: we could either place an information gateway between users and social media to capture precious moments before they are sent out to the ether (Figure 1.6.) Or attempt to group moments already posted on various social media platform to reconstruct events (Figure 1.7.) The final result for both ideas is to create an event based digital scrapbook where users can preserve and share.

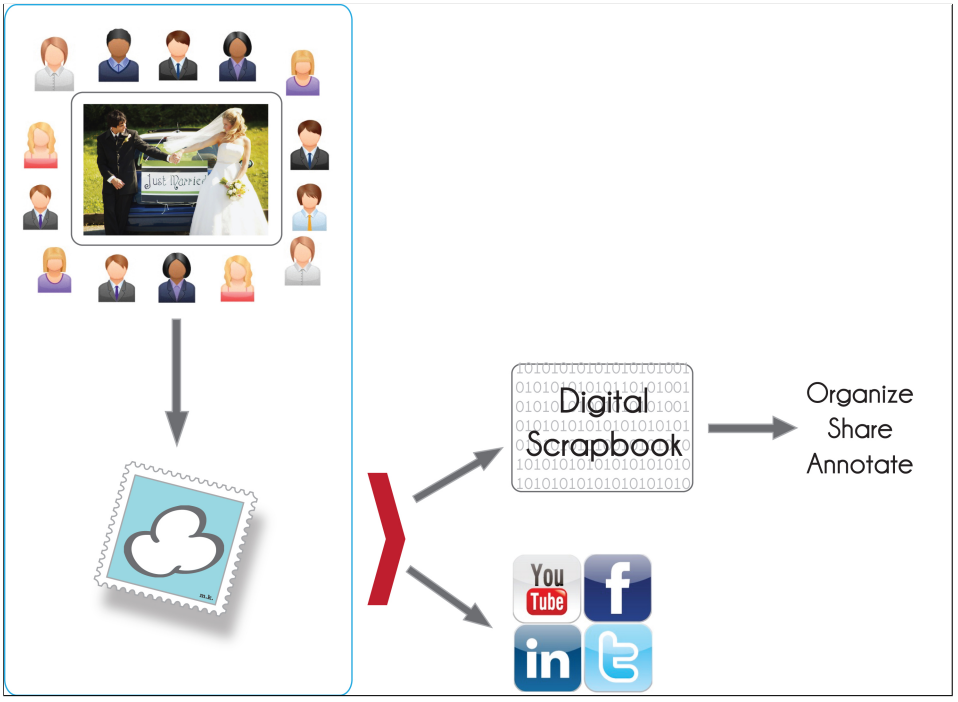


FIGURE 1.6: Information gateway between users and social sharing


FIGURE 1.7: Information grouping and filtering to reconstruct memories about an event

This ultimately manifested into two corresponding solutions: a mobile application which allows users to capture memories on their smartphones and post them to specific buckets (or kloudlets) in MemoryKloud. If they choose to, users can ask MemoryKloud to push these memories to the social media platform of their choice. The other solution would be a web application that will pull all of users' own posts on various social media sites, and via rule-based mechanisms or data mining techniques, it will be able to automatically

assign posts to existing events. Of course, such automatic classification methods cannot guarantee a hundred percent accuracy, that's why the web application is also needs to provide capabilities to allow users to manually curate photos in case of automatic classification errors.

### 1.3.2 Solution Delivery

The solution we arrived to is a web service application powered by Python, which provides RESTful API's to both an Android mobile application and a web application (See Figure 1.8.) The backend storage database engine was powered by MySql.

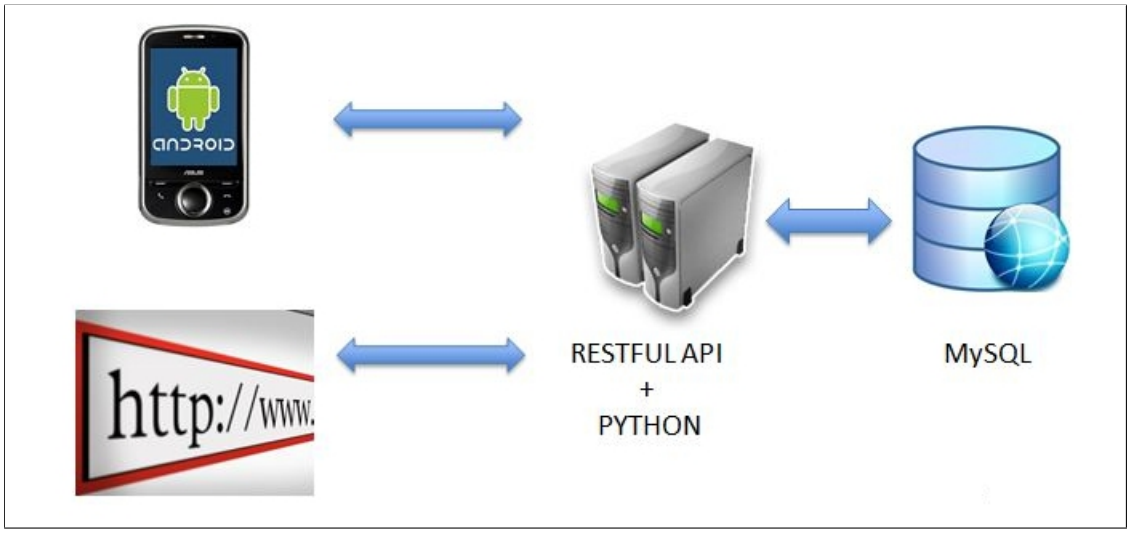


FIGURE 1.8: MemoryKloud Architecture

#### 1.3.2.1 MySQL

MySQL was our main database to store user information and the content they upload. We designed the database schema by examining the relationship between different entities and drafted out ER diagrams to model the user, the kloudlets, the moments, and their underlying associations (Figure 1.8). We intended to follow the recommendation of third normal form where we usually have two entities tables with a mapping table in between. We took an iterative approach to design the schema, where we started with necessary entities that are needed for the given tasks, and then elaborated the schema design by drilling down the details of the tasks.

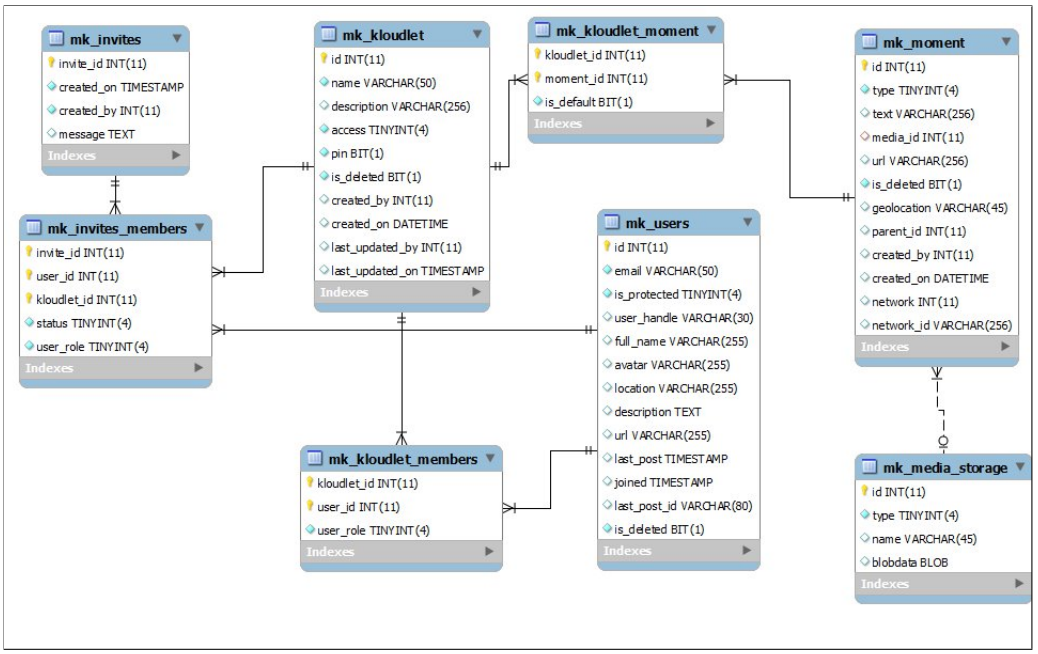


FIGURE 1.9: MemoryKloud Database ER Diagram

#### 1.3.2.2 Python

A Python-based server sat as an intermediary between the MySQL database and the web front-end & Android application. The server provided RESTful API's so the applications can communicate with the database. Generally, there are three components:

- *Database logic:* this piece specifically deals with the interactions with databases, for example, issuing SQL statement queries or stored procedure calls.
- *Rendering logic:* The database objects are being converted into proper HTML or JSON formats
- *Utility logic:* perform some utility functionality such as file IOs, data type conversions, etc.

#### 1.3.2.3 Android SDK & Eclipse

Eclipse served as our main development IDE for our mobile application. Its debugging functionality is extraordinary and proved to be very useful. Moreover, the ability to test the code in virtual phone as well as in a real phone helped us in executing the code and seeing its results faster. The disadvantage with this is that the interface shown on real phone isn't the same as shown on virtual phone. Hence, we always needed to test the code in a real phone to ensure correct presentation.

#### 1.3.2.4 SVN

SVN was our version control system of choice. SVN helped us a lot in collaborating with each other when developing while staying at home. The program is quick to download and install, and it helped us to integrate our code smoothly. Also, it is highly reliable in updating the codes correctly every time.

#### 1.3.2.5 HTML/CSS/Javascript

These basic web development tools were used to build the web front end of MemoryKloud.

#### 1.3.2.6 Amazon EC2

We employed an Ubuntu instance to run the Python server and the web application. EC2 proved to be a very flexible and robust platform. Since things had a hight propensity to break at such a an early stage, the ablity to copy instances and make several images of it proved to be indispensable.

### 1.3.3 Android Application

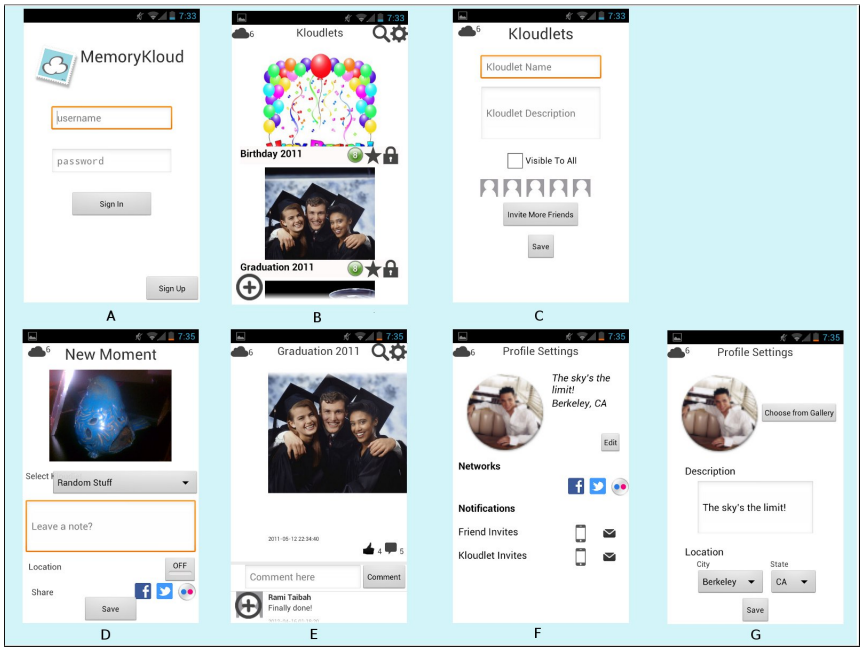


FIGURE 1.10: Android Application Screenshots

#### 1.3.3.1 Application Common Vocabulary

- *Moment:* a particular piece of memory a user would like to preserve; it could be a video, picture, a comment, etc.
- *Kloudlet:* a container where relevant moments are being grouped together i.e event

#### 1.3.3.2 Application Screens Explanation

- **Screen A:** This is the login screen where a user provides a username and a password. When they click on the "Sign In" button, the system will authenticate their credentials. If they are authenticated, **B** will show up, otherwise, there should be a pop-up informing the user of the wrong username or password. We provide functionalities to the user to sign up for the application and retrieve their credentials if they fail to remember them

- **Screen B:** This is the *kloudlets* page for MemoryKloud application. There is an "action bar" on top with three main components:
  - A "kloud" logo icon along with a number that illustrates the number of notification
  - A title named "Kloudlets" to denote the main page
  - Action buttons including:
    - Search: allow users to do keyword based search for kloudlets and moments
    - Settings: When clicked, **F** will show up
  - Below the "action bar", we have kloudlets being presented to the user in a grid format
  - For each kloudlet, several elements are available:
    - A selected picture in the kloudlet used as the cover page of the kloudlet. If a kloudlet doesn't contain pictures, a default "blank" picture will be shown
    - The name of the kloudlet at the bottom
    - A lock showing whether that particular kloudlet is public or private. Public kloudlet are visible by the general public while private kloudlets are only visible to selected participants
    - A number showing how many updates this particular kloudlet has
  - By default, the latest updated kloudlet will float on top. In addition, users will have the capability to "pin" a kloudlet to the top

  - Users will be able to scroll up and down to see their kloudlets, once clicked the moment page **C** will show up

- **Screen C:** This is the page where a user can create and share new *kloudlets* . Below the action bar, you will need to type in a name of a new kloudlet along with a description. You can define this kloudlet being a public or a private one. In the meantime, you can invite friends in your social network to join and contribute to this kloudlet. When the user clicks on create, the new kloudlet is created on the server and it will be shown in the kloudlets page.

- **Screen D:** This is the page where a user can create and share new *moments* . Below the action bar, you will need to select an existing kloudlet or create a new one where this moment belongs to. You can provide additional description of the moment, turn the geo-location meta-data on or off, choose what social media to publish on and by clicking on save, the moment should be saved onto the server, and it will be shown in the moment page.

- **Screen E:** This is the *moment* page for showing contents (video, pictures, comments, etc.) of a particular kloudlet. Similar to kloudlets page **B** , it does have the exact same "action bar" with the exact feature explained above. When user clicks on the "Add" button, **D** would show up.When user clicks on adding moment, **E** would show up; when user clicks on adding kloudlet, **G** would show up.

  - Below the "action bar", we have moments being presented to the user in a list format.
  - Each moment shall have an avatar and an associated name to denote who posted the moment. When the avatar is being clicked, user can see its user profile as well as its moment "wall", where they can also add user as friend.
  - The moment could be a video, picture, comment, etc. And, associated meta-data should be displayed as well including post location, post time, with who, etc.
  - User will be able to slide up and down to see the moments being listed. They can also swipe left and right to see other different kloudlets. And when they click on a particular moment, a pop-up would show up asking if they would like to like, comment, or delete

- **Screens F & G:** These two screens are basically the setting page has the following components:

- *Profile:* here lists user's avatar, name, description, and location. And when user presses on them, they can change their profile.
- *Network:* Network: here lists user's network configurations so that when configured properly, the moments shall be posted to different social media.
- *Notifications:* There are two ways of getting notified: in-application notification and email notification. So far we would be able to send the notifications when friends sent an invitation request or kloudlet request. User has the option to turn it off or on.
- *Bio Editing:* A simple text field to provide a short bio of yourself

### 1.3.4 Web Application

While MemoryKloud is essentially mobile driven in a sense that moments are expected to be captured with a mobile phone and uploaded from there, a web application plays an essential role in the whole MemoryKloud experience. MemoryKloud Web provides a fuller and richer experience when reminiscing around an event. One does not expect a family huddled over a mobile phone to look at last year's Christmas photos. Instead a phone will be passed around to show a picture or two. This makes the experience somewhat disjointed and very incomplete. A web application is essential to simulate the original paper photo album experience while adding features that were impossible to achieve using the old fashion album such as multimedia, crowdsourced content, commentary, and theming. The web application also plays a very important role with editing and curating content for any given event. MemoryKloud Web's main purpose can be summarized in three points:

- **Richer Experience:** As mentioned above, the scenario of reminiscing over Christmas photos is incomplete with a mobile only solution. A full-fledge website will provide MemoryKloud users a myriad of features not available for mobile technologies.
- **Curation & Editing:** Having a web application makes this abundantly easier than a mobile-only-solution. Dragging & dropping, batch editing, batch uploading, captioning, and other editing methods can only be done through a modern desktop operating system. The importance of such techniques becomes obviously clear when employing data analysis and machine learning techniques to identify memories worth preserving (see next section.) No machine will be able to identify important memories, after all it is up to the person experiencing a memory to say what is worth preserving and what is not. However providing the right tools to

make quick decisions on machine output can increase the efficiency of the system dramatically.

- **Business Strategy:** Having a linkable website is sound business strategy. In order to cut costs, many tech startups jump directly to the mobile application part and discard the web. This in our opinion is a mistake. Many underestimate the sheer power of the simple concept of linkability. This simple concept is what made the Internet what it is today. Being able to link a documents to another, or share a document with others using a simple link, are deceptively simple concepts that many tend to easily dismiss them. Path for example, the immensely popular social network, skipped the web and went directly to mobile. While there might be a business reason behind this, it ultimately frustrates users. Not being able to have quick and direct access to their own content can put many users off.

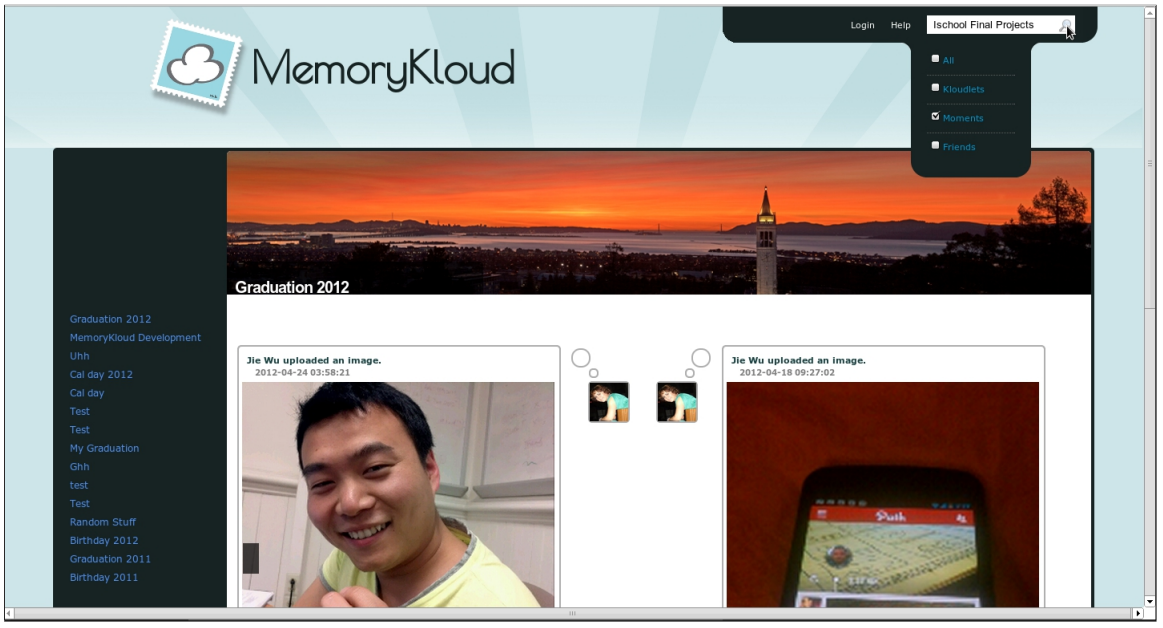


FIGURE 1.11: MemoryKloud Web Application

Due to many logistical reasons and time-constraints, MemoryKloud Web for this project focused on the first purpose; providing a richer experience. While we believe that providing a solid platform for curation & editing is necessary for a solid data analytics experience, however the true capabilities of data analytics in MemoryKloud are still unclear to design for. Luckily, designing for a richer experience bleeds a lot into business strategy.

#### 1.3.4.1 Design Process

Using our contextual interviews as a guideline, and our experience in the research that went into designing the Android application, we started off by doing rapid paper prototyping and testing them on different people in the I School information and the larger Berkeley community. Many of these tests helped us wade through the many features a website can offer, and kept us focused on what is important by keeping thing simple. One feature that stood out is that many participants valued an efficient and robust searching system. Some cited Facebook's own search is very convoluted and rarely accurate, and that they would value a site that would make searching for memories (i.e pictures, video, thoughts..etc) less of an arduous task.
Once we got a paper prototype that we felt happy with, we jumped directly into the HTML, CSS, and Javascript coding. We decided to skip creating any hi-fidelity prototypes on Balsamiq or any similar tool believing that the ultimate web prototyping tool is actually programming it. This approach gave us the flexibility of testing the prototype anywhere, helped us debug and test our API's and backend, and ultimately gave us code that would help us in future iterations.

### 1.3.5 Data Analytics

The question remains "how to group similar moments already posted on various social media platform based on their similarity?" In other words, for someone's posts on different social media, can we automatically classify them into events? Generally there are two approaches for such classification problem. For one, we can define a certain set of rules. For instance, moments posted on a single day with the same group of people might be associated with a particular event; or posts with the same hash-tags are likely to be related as well. On the other hand, we can use statistical methods to do classification. In this project, we are more interested in exploring the feasibility of the statistical approach rather than delivering an optimal solution for moment classification. Also, our focus was on evaluating the classification methodology. By no means, a statistical approach can provide a 100% accurate depiction of an event. A statistical method would certainly need to have the web application mentioned above as an buttress any classification mistakes that are bound to arise.
We collected sample Tweets on a single day (2011-01-23) from the NIST corpus http://trec.nist.gov/data/tweets/, and proceeded to cluster the tweets by their relevance (a.k.a, topic clustering). Below are the steps we took and we also reported the result from our research.

*Step 1:* Corpus preprocessing. NIST provided a tool where we need to run and crawl tweets on twitter.com based on the given Tweet IDs. The problem was that there are a lot of tweet entries where they are no longer available either because they are being deleted or protected. The other problem of using the linguistic corpora is that it consists of texts in different languages. We removed all the tweets entries that don't return any tweet text, and we also removed any tweet that contains non-ASCII code. In addition, we would like to exclude the tweets that contain lots of short acronyms. We did so by removing words with length less than 4 together with stop words in the English language in the tweet, and delete it if its remaining word count is less than 15. (Note: the average words of tweet is 14.98 http://web.resourceshelf.com/go/resourceblog/54128)

*Step 2:* Convert corpus to document-term matrix and dimension reduction We wrote a python program to convert the corpus into a document/Tweet-to-term matrix with the cell being tf-idf values. The tf*idf weight (term frequency-inverse document frequency) is a numerical statistic which reflects how important a word is to a document in a collection or corpus. Thus we ended up with a 6432 * 2527 matrix. Such matrix, however is too sparse since it contains lots of zero entries. Thus we used SVD (Singular Value Decomposition) to reduce the dimensionalities of the matrix. We decided to only take the first 10 dimensions for our following computation. These 10 dimensions essentially reflect the most representative features of the terms.

*Step 3:* Clustering. Now we have a 6432*10 matrix, where we are ready for clustering analysis. We fed the matrix into R and run k-means clustering algorithm. We defined the k to be 10, and initial random seeding for twenty times. kmeans(part,10, nstart=20). We plotted the data points along with the clusters, and we get Figure 1.12:

We can see from the figure quite clearly there are three main clusters: blue, green and red. The numbers of data points they contain are respectively 2459, 2610 and 1104. We did a further text analysis trying to get a rough understanding of what these clusters mean. We get the words from each cluster whose occurrence frequency is greater than 1, and for each word, we calculate the similarity based on the following metric:

The reason for this exercise is trying to figure out the words that would best distinguish the clusters. Following are some sample words for each cluster:

*Blue Cluster: indonesia, ujian, ngiler, makanan, maen*

*Green Cluster: packers, match, season, Chicago, # packers*

*Red Cluster: bitch, fight, hospital, hurt, jail, army*

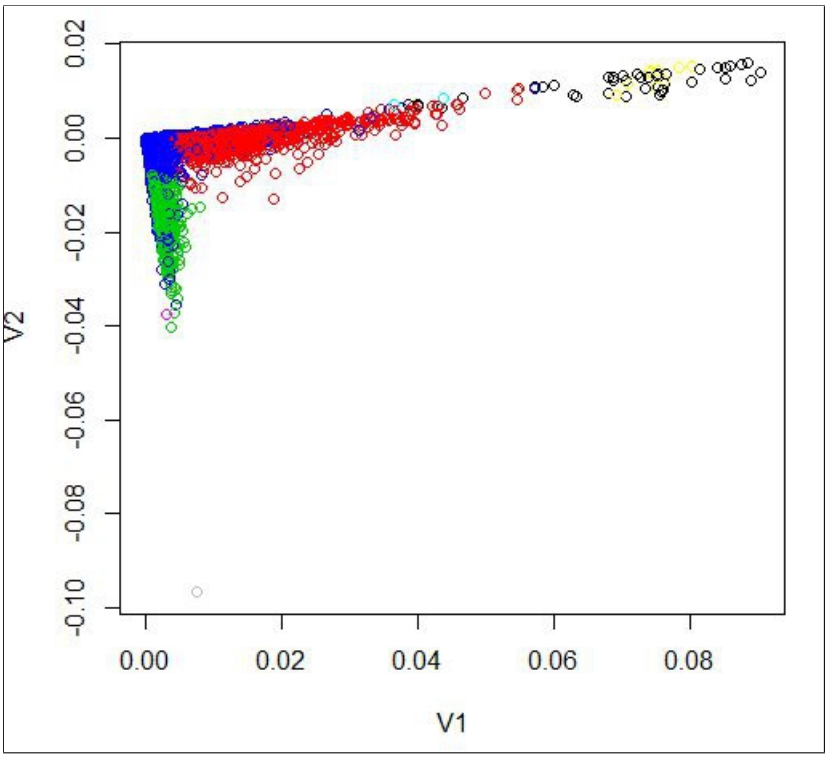


FIGURE 1.12: Clustering Results

$$\text{Similarity} = \frac{|\text{WordCount in Cluster A} - \text{WordCount in Cluster B}|}{|\text{WordCount in Cluster A} + \text{WordCount in Cluster B}|}$$

FIGURE 1.13: Similarity Metric

**Blue Cluster:** It seems like that the blue cluster is a Malay cluster. Malay is a language used in Malaysia, Indonesia and Singapore. It contains the exact alphabet as English, and we were not able to filter the language out using our non-ASCII method above. Also, Malay consists 6% of entire tweets space.
http://techcrunch.com/2010/02/24/twitter-languages/

**Green Cluster:** The key words shown in the text analysis suggest that this cluster is about a sport event. We looked up the event that day, and we found that there was indeed a NFL playoff event between Green Bay Packers and Chicago Bears on Jan 23, 2011.

**Red Cluster:** this cluster was not yet easy to identify since there are no apparent linguistic clues to associate the given words together. We are guessing it might be some negative tweets being grouped together.

*Step 4:* Classification. We went one step further interested to see how good it will be for our purpose of topic classification. We assigned each tweet with a label being their cluster ID, and then we divided the 6432*10 matrix into two parts, 90% for training and 10%

for testing. SVM (Support Vector Machine) algorithm is being applied to classification and following chart (Figure 1.14) shows the confusion matrix being rendered by the algorithm:

| | true | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| pred | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 2 | 255 | 12 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 10 | 209 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 2 | 12 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 8 | 1 | 1 | 0 | 0 | 0 | 3 |
| 7 | 0 | 1 | 0 | 7 | 0 | 0 | 5 | 0 | 0 | 4 |
| 8 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 4 | 2 | 17 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 20 |
| 10 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 43 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

FIGURE 1.14: Clustering Results

As we can see, the blue cluster (Cluster 3) and the green cluster (Cluster 4) yield a relatively consistent result where most of the data points are classified into the correct class. The red cluster (Cluster 10) can hardly get it right, implying that this cluster might not capture the representative characteristics of the data points within it.

### 1.3.6 Future Work

Overall, the effort and learning we gained from working on MemoryKloud provided us a very solid groundwork for future endeavors in the fields of UI design, user experience, and data analytics. And we are hoping that we could take it to the next step.

As far as MemoryKloud is concerned, the RESTful API's are the heart and soul of MemoryKloud. Getting a robust and full set of API's is of utmost importance and this stage. To get the most performance, porting the API code from the current Python codebase to a Java one might be a very viable option. Also, the API is currently missing a lot of functionality, such as user-specific attributes (avatar, location...etc) and commenting. API design will prove to be the next immediate challenge for MemoryKloud.

Other functionalities that must be implemented before having a minimal viable product would be getting authentication in place and deciding on the different options we have (native authentication vs. OAuth vs. both,) implementing an invitation and notification system, and implement navigational elements of the website server-side instead of client-side. The current Android app, can post images and create new kloudlets, however there is still a lot of work to be done on this front. Uploading content other than images, authentication, UI & theming enhancements, caching, location identification,

and pushing to other social media websites are the main deliverables before launching.

The website is heavily dependent on what the API can do. This is why we emphasized on the importance of improving the API as quickly as possible. Currently, we have a very basic website that can query the API, receive information from it, and display it. The next step would get authentication, notifications, invitations and user-profile working, all of which are API-dependant. As a further step, curation and editing tools need to be developed along with searching methods, both of which will prove to be challenging since they are heavily dependent on the quality of the data we have.

Once the technology is set in place, and the product is launched, MemoryKloud can go many places. For example business partnerships with entertainers, musicians, and sport franchises might be a viable option. Also, printing events in albums, posters, and other materials might also be very lucrative. In any case, we are very excited about the different possibilities that MemoryKloud has, and looking forward to work/see where it goes.