“So-Mo-Clo”

PG Madhavan
15 June 2012

Social-Mobile-Cloud – an emergent framework to create and deliver services. Can this framework be applied to humans and devices? Yes! Can it be used in the enterprise and for personal use? Yes! Is this something new? Not really . . .

So-Mo-Clo is the result of progressive maturation of mobile devices, cloud computing and social networks. “Social” networks can encompass a “tree graph” in an enterprise (hierarchical org chart), a “mesh network” of information exchange among friends and a collection of machines operating as a “society” (Internet with its distributed routers).

Some 15 years ago, while I was developing solutions to monitor large machinery such as a paper mill or a steel plant, we connected up the machine monitoring devices into a local gateway, sometimes wirelessly. This gateway led to the Internet and a server far away. The analysis of data done at the server provided machine condition information to the plant floor operators, to their foreman and plant manager. Some of them had laptops with connection to the corporate Wi-Fi. As you see, a full-fledged So-Mo-Clo system! The society of plant staff consumed the data provided by the server on mobile laptops. No smartphones then for SMS notifications . . .

Today, I am involved in developing a social commerce system using mobile phones that leverage the connections among friends where integration, messaging and commerce are executed on a platform on Azure cloud (could be AWS also). A true full-blooded So-Mo-Clo solution! As you see things have not changed much but it has become a lot easier – my machinery solution took over a year to put together but the recent social commerce solution took 3 months! All the pieces are in place to apply the framework to a myriad of situations which has the potential to change the way we live.

Is there any problem that does not fall within the purview of So-Mo-Clo? I cannot think of any. In the mobile, cloud and social loop, all our work and play scenarios can be addressed.

- My mobile app for finding a taxi: my “mobile” connected to the taxi dispatch “cloud” that interacts with the “society” of taxi companies that bid for my business on my mobile.
- My enterprise sales team info-share network: I upload juicy tidbits (“CTO of “X” has recently moved to “Y”) into my company Sales Social network; my enterprise Active Directory disperses the information within the relevant portions of the org chart; my colleague in the relevant territory picks up the news and send me an SMS asking for tips about the prospect.
- My enterprise Wi-Fi network: SNMP and 802.11k information is periodically updated to the IT diagnostic backend where a “heat map” is created; a dark spot is communicated to field engineers’ mobiles and the nearest one goes to the location to find the AP unplugged.

When you are a hammer, every problem looks like a nail – perhaps; but to me, So-Mo-Clo framework seems to be pretty all-inclusive!

Business Case for So-Mo-Clo:

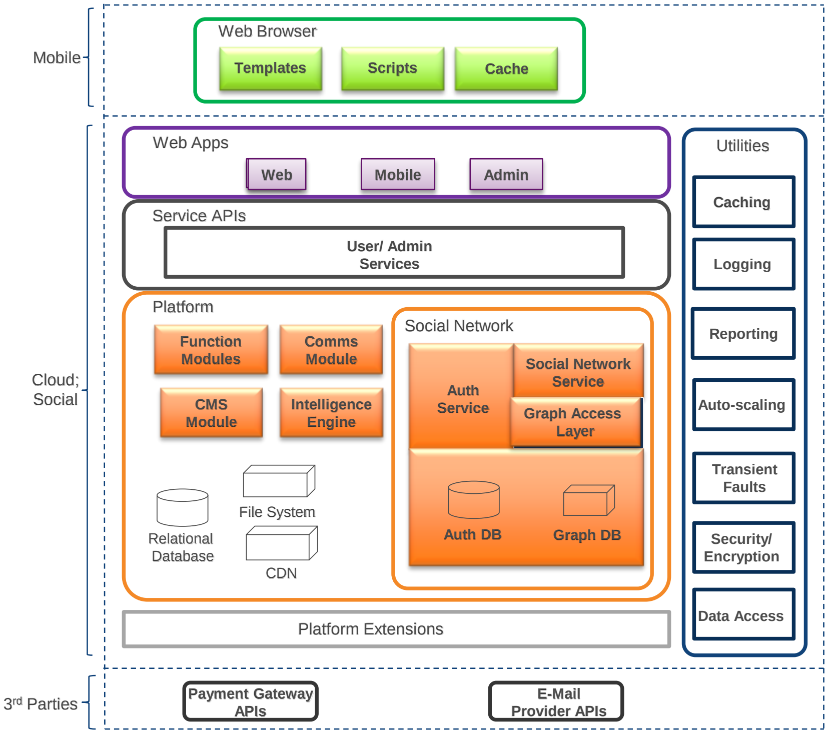
What is the monetization opportunity arising from the realization that So-Mo-Clo is a framework that can be applied widely?

1. Productize it and sell the So-Mo-Clo platform to device+service companies; make it easy for them to add their specific services.
2. Build your own device+service business and launch it.
3. Major cloud and telco service providers offer So-Mo-Clo platform to all comers; lots of startups can get off the ground very quickly (attractive in emerging markets) and grow in response to market demand (till cloud costs become so prohibitive that you have to bring it in-house to your own data center; this is not all bad – it means that you have successfully created a huge business!).

Business Model #3 can be a threat to the first two in that So-Mo-Clo as a “feature” in a major offering such as Azure can subsume them. On the other hand, Business Model #1 has the advantage that it can be tailored for many applications. Let us consider an enterprise example. A full implementation can serve the entire organization; a mini-platform can be used by a department and a micro-version can be used for a project. Add to it document sharing, audio and video calling/ conferencing and you have an excellent cooperative-work solution.

So-Mo-Clo Architecture:

While many variations and technologies are possible, let me introduce a canonical example.



A canonical So-Mo-Clo Architecture

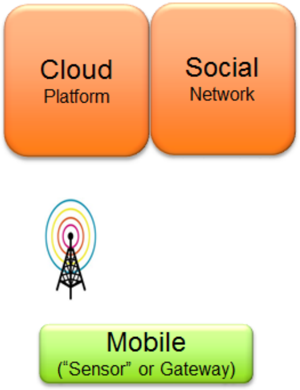
- Client at the top is the rendering engine for Mobile and Web destinations. The client will be responsible for interpreting/loading markup along with executing other payload parts such as scripts and styles. Technologies include HTML5, JqueryMobile, PhoneGap, Javascript and AJAX/JSON.

- Cloud portion hosted on Azure is the backbone of the system. The Platform itself and the Social Network is separated into 2 components making it dispersible (with Social Network running on AWS, for example) for redundancy and other purposes. Technologies used could be Azure, ASPNet MVC 3.0, WCF Services, REST/JSON, Neo4J Graph DB and so on.
    1. Presentation Layer: This layer comprises the server side presentation components. They work in tandem with the client side scripting frameworks to present a rich user experience.
    2. Service API Layer: This layer is an interface for the Platform. It presents the Platform features to the other layers or the external world. The Presentation Layer and the external applications integrate with the Platform through this interface. The layer also acts as an orchestration layer forking /consolidating the request/responses into/from the platform layer.
    3. Platform Layer: This layer comprises of the core business logic components in the Platform. The components are distributed to provide the required scalability, flexibility and performance and are accessible through the business logic APIs per component.
    4. Data Persistence Layer: This layer contains all the storage mediums like SQL, Blobs, CDN and is used by the Platform for all the storage needs. The relation data is stored in SQL Azure, the files and large non-relational data entities are stored in Blobs and CDN is used for the media content.
    5. Cross cutting Utilities Layer: This layer contains the components which can be utilized horizontally cross the functional components/layers. The layer uses the Enterprise Library from Microsoft Patterns and Practices group to wrap the components to handle the non-functional aspects.
    6. Social Network: Authorization Service can be a REST based WCF service. RDBMS database can be used to keep user credentials, authentication token and user permissions. Social Network service can be another REST based WCF service which uses JSON format to communicate through APIs. Social network API applies authorization through authentication token generated by Authorization service. This service communicates with two databases one (Graph Database) for keeping the person (node) and relations (edges) and other (RDBMS/NOSQL) for keeping user profile, private messages, statuses and notifications. GraphDB keeps the person (node) and relations (edges) information.
    7. As a rough guide to cloud infrastructure sizing, a 1-Million customer application we had developed uses 30 large Azure instances and 300 GBytes of memory. Of course, these numbers will vary according to the application specifics.

***Security** is of utmost concern in large consumer facing applications which typically call for So-Mo-Clo solutions. Web and mobile-web portals should be SSL based (https) for the transport layer security. The security can be augmented by the Unified Login and Form Authentication - store encrypted username and encrypted password in the data store and authenticate by comparison of encrypted (using the same key) inputs. Service API internal end-points and the Business Logic service internal end points can be secured through Windows Azure AppFabric Access Control Service. The strategy is to generate a Secure Web Token from the ACS and then pass this token to the services via an Authentication http header. Encrypting and protecting user data in the data stores are also of utmost importance. This can be achieved by encrypting (using public-private key pair from Windows Azure Certificate Store)the configuration sections and allowing the code running on Windows Azure be able to decrypt and read them. Because of the data sensitivity and the laws around the security of personal information, the* *access to data should be constrained by making sure that there are no endpoints for individual data access and only very specific endpoints are available for aggregated access of the data.*

So-Mo-Clo as a Superset & Some Trade-Offs - A Personal Perspective:

So-Mo-Clo architecture can be the framework to address any of the scenarios described in the introduction section (the issue of "hammer and nails" notwithstanding). Things we call IoT (Internet of Things), M2M (Machine-to-Machine), Connected Devices etc. map into the architecture framework of the last section.



Reduced to its bare minimum, Cloud provides the computing platform for UX, data processing and intelligence. Social Network represents the "mesh connectivity" among the users and the intelligence embedded in the Cloud can use this infrastructure to understand and communicate insightful information (more about how to do this in the section on Social Networks) and not just raw data to interested parties within the "social" network.

Mobiles connect to the Cloud platform via wireless communication channels provided by MNOs or wireless+wired channels provided by ISPs. Mobiles may themselves be ends-points generating data (from the apps or audio/ video inputs) or they could be gateways aggregating data from other sensors (such as exercise monitors or glucose meters) connected to them via Bluetooth or USB cables.

When So-Mo-Clo is described in this generic fashion, it is clear that this is a classic M2M network or IoT for home automation! Add middleware for pub/sub protocols (AMQP, DDS or recently popular MQTT) to reduce network activity and implement Java OSGi Bundle technology to dynamically utilize on-premises or in-cloud computing resources; you have a highly elastic and multi-purpose So-Mo-Clo solution wide ranging application domains! Connected Transportation, Meters & Sensors, Machines & Screens and of course, social commerce and other human endeavors fall within the umbrella.

Again, there may be a bit of over-reach on my part but the unifying vision is compelling I believe.

There are many potential trade-offs in the solution outlined in the previous section. Let me make it clear that I am technology agnostic. My personal preferences are as follows:

- Mobile: Most applications require populating more than one form factor (mobile, tablet, laptop display); even within one such as mobile there are multiple sizes and resolutions.
    - HTML5 with associated native access libraries (such as PhoneGap) is mature enough today to provide best in class UI/UX. Developing for each native mobile app may be overkill.
    - In UI design, responsive design so that UI is pixel-perfect for any screen metrics is a must to provide outstanding UX.
- Cloud: Many of the recent benchmarking studies put Azure and AWS neck and neck. Pricing works out to be similar.
    - Hence the choice is purely driven by familiarity. .Net programmers will prefer Azure for the continuity it provides in the development environment.

- No cloud is a perfect utility (even our electrical utility fails once in a while); both Azure and AWS have gone down (not frequently). So the safe approach is to create redundant deployments on both Azure and AWS (or some other clouds) if your financial situation permits.

- Social: Structure and storage affect performance as the network scale (number of users) grows large.
    - While many organizations such as business firms are structured as a "tree" in a hierarchical fashion, communication among the nodes (users) are in a "mesh" or many-to-many fashion.
    - Traversing a tree from a leaf-node via root to another leaf-node may become very inefficient in terms of latency if the tree is large. While it is seen that a relational database with Memcashed is a way to manage latency, a true graph DB is the right storage for a multi-million node social network.

Social Networks:

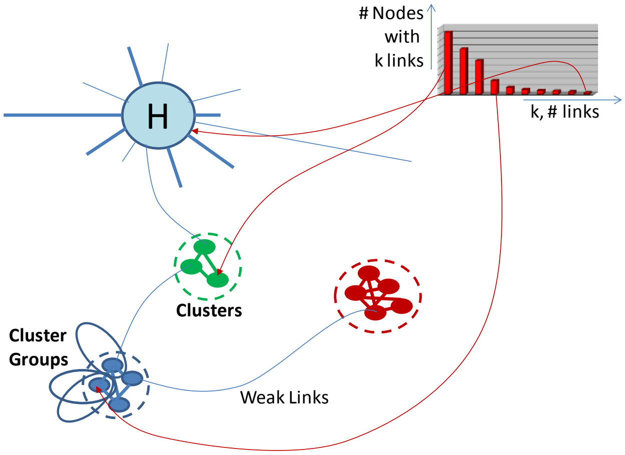
To the initiated, "mobile" is well-understood and "cloud" is becoming better known. However, "social networking" and how one can exploit it profitably is still a black art. So here are some highlights.

In the last few decades, interest in Social Networks has seen explosive growth with Facebook being the poster child. Indeed work had gone on before that time in Physics especially in the area of "random network" (Erdos and Renyi are famous names). In such networks, each new node is randomly connected to any previous node and its evolution was studied. The work of Barabasi and others (Scale-Free Networks: A Decade and Beyond, Science, Vol 325, 24 July 2009; pp 412-413)showed that while the physics of Random Networks was interesting, it had little to do with "human made" networks such as Web, Internet, friendship networks, etc. They all fall into the category of "Scale-Free Networks" or SFN.

"Scale-free" in SFN refers to the nature of some networks where its diameter (average # of links to reach a node from the present node; remember "six degrees of separation" between people) is mostly unchanged as the network grows in size. SFN's degree (# of links per node) is distributed as a power law (and not a binomial distribution which is the case for a purely random network). A well-known example of power law is Pareto distribution ("80/20 rule"). SFNs arise in networks that grow and have preferential attachment property while growing (exhibits "rich get richer" and "first mover advantage" phenomena).

Real social network has 3 major features:

1. Clusters (of friends who "hang out").
2. Weak links (to your long forgotten high school classmates).
3. Hubs denoted by "H" (politicians and others with massive number of contacts).



Comparing the Pareto (power law) distribution (top-right) that governs SFNs, we can see that Hubs with many links (large “k”) are nodes that occur rarely whereas Clusters with few links are nodes that occur very often. Here we elaborate a mid-scale situation where certain nodes have a fair number of links and occur with some frequency – we call them “Cluster Groups”. They arise in cases where a certain Member belongs to multiple Clusters, their family cluster, work cluster, golf club cluster, etc. The same Member belongs to many Clusters that are tightly coupled leading to a Custer Group.

Failure Modes: (1) SFN is robust to Failure and (2) SFN is vulnerable to attacks.
Failure means random loss of network elements such as some computers or local routers going down in the Internet. Attack refers to concerted take-down of major “hub” routers. It has been shown that removal of a small percent (15 to 20%) of “hub” routers will pull apart the network and “cascading failure” will spread (due to routers NOT under attack being unable to carry the extra load from “hub” routers) leading to total network failure.

The knowledge of failure modes has applications in the real world. Since Hubs (or “Influencers”) have the potential to be very helpful in spreading messages such as in a viral marketing campaign, it may pay to reward them with extra incentives. By the same token, we saw that a few Hubs leaving the network can cause “cascading failure” – if your business has a customer support feature (who doesn’t!), pay special attention when Hubs call in with a complaint – coddle them!

SFN Terms	In Real Life	What can they do?	How to Find?	Action
1. Cluster	Close friends	Influence economic activity	Found as most used in the Contact List	Make messaging easy
2. Weak Links	Distant acquaintances	Increase the network size	Found as seldom used in the Contact List	Message infrequently and mention context
3. Hubs	Powerhouse “connectors”	Keeps network robust	Algorithms	Value exchange messages

These insights beg the question – how do you find Hubs and other key influencers? Rooting through the graph and counting the number of attached nodes may be one way. While useful, this approach could be misleading in that some node created a lot of connections to them early on but there has been not exchange of information lately. Here is a more elegant way to analyze the social graph and derive meaningful analytics from it.

Much of the work in social networks can be modeled and understood as the spread of infections in a population. In an IT context, we will talk in analogous terms of “mention of information or word” in a posting as indication of an “infection” (as in “that posting is infected by that particular word”).

Each Node has a different “influence function” – a Node talks about the “word” (Infection) to varying number of other Nodes (out of total of ‘N’ Nodes). Volume of Contagion for each infection (or word) is the total number mentions of the Word in the network (for each of total ‘K’ words of interest). In linear algebraic terms,

$\mathbf{v} = \mathbf{Mi}$

M - Influence Matrix (KxN);
v – Volume of Infection vector (Kx1);
i – Node Infection vector (Nx1)

To the initiated, this equation is a joy to behold in its simplicity and familiarity! Linear algebra has so many powerful tools, fast algorithms and practical insights that a lot of useful information can be extracted by the expert. In general, singular values and vectors (and related eigenvalues and vectors) of matrix, M, can tell you which nodes are the major "influencers" or Hubs, how big is their influence and what forms the bases of their influence even!

In applying this technique in practice, you will start with simulations of your social commerce network and predict Hubs, influence and other behavior. *For those interested, I can provide the 'm' file of the MATLAB code that Scale Free Network of 'N' nodes and 'd' degrees (2 is typical for social networks) and some plotting routines.* This is also the time to optimize your algorithms for large networks (large 'N' nodes) for matrix computations above as well as develop iterative or adaptive solutions so that you can study and predict network behavior over time with minimal computations.

Once your business is up and running, you can start replacing simulated data with historic data collected from your real network and apply the results and insights to your business to improve or control the viral spread of your marketing message or identify the Hubs to provide them with targeted incentives and so on.

From the foregoing, it must be clear that So-Mo-Clo framework is a powerful one that brings the power of mobility, cloud as a utility and social networking as an analytics tool. Applications are many and will impact many business segments. Whether you are a So-Mo-Clo platform vendor or a user, the solution framework is already past the feasibility threshold today and your insightful use will have a positive impact your quarterly revenues!