

# Bubble City Design Proposal

A Twitter Alternative  
Which is not a Social Medium

It is a  
Real Time Idea Router

Monica Anderson  
Syntience Inc  
October 31, 2022

# Background

We can today choose from a plethora of chat, messaging, email, video, and social media apps for our phones and web sites for our browsers. They have made different design choices, and each has their fans and users.

A new service can set itself apart by making a set of design choices nobody has made before them. The overall philosophy and the resulting coherence behind these choices will **define the philosophy and coherence of the service**.

With the arrival of modern Natural Language Understanding technology, as exemplified by GPT/3 from OpenAI, BERT and LaMDA from Google, and my own UM1 (Understanding Machine One), it has become possible to perform, in a robust fashion, a previously unavailable feat:

## **Routing messages by content.**

I wish to create an "Idea Routing Service" centered around this feature. With this as a starting point, a number of other rather uncommon design features become available and from among these, we have formed a new coherent whole which (we expect) will produce a stream of messages of unprecedented quality on any topic in real time. This will be wrapped in a very smooth and easily fine-tunable reading experience.

In essence, we are designing a real-time messaging system where an AI (an Understanding Machine that has learned to Understand some human languages) does the filtering and message routing (sorting) for all users, according to each user's wishes.

The routing, how it works, and how to fine-tune it will be described in what follows. We will list the key features and design decisions of the proposed system as it stands today. No code has been written yet, and all proposed features are malleable or removable based on stakeholder feedback.

I have named this system "Bubble City" (BC) because the name plainly and irreverently exposes the main feature of the service that makes it radically different from everything else.

I have acquired [bubble-city.com](http://bubble-city.com).

The design has been iterated for a few weeks in a Private Secret Facebook Group. This is version 2.0 of the design proposal. It is mostly a cleanup of the original whitepaper, because it mostly discusses the Minimum Viable Product. Many suggestions in the group will be implemented only in later versions. If you want to be included in the group to see what we are planning in the future, please PM pandemonica on Facebook or send an email to [monica@syntience.com](mailto:monica@syntience.com)

This work started as a 20% project at Google in 2006. Google requested that I patent the design done by me and Pablo Maygundter, which I did. These included patents for content based routing, many UX features suitable for this project, several monetization strategies, real time search, and others. Many features of the system (including content based routing) were invented by me before joining Google.

Google is not in the habit of enforcing their patents. Most likely, if BC becomes popular, they might want to buy into it, which I'm totally willing to discuss.

# Bubble City

## An Idea Router, not a Social Medium

The goal of the system is to provide very high quality **uncensored** (to the extent permitted by law) yet spam-free and hate-free real time message feeds, precisely personalized to each user, on any number of specific topics.

Each topic is confined into a "Bubble". These are so powerful that I sometimes conjure up the mental image of a Stainless Steel Bubble that is impervious to information that does not belong in the Bubble.

Information bubbles are only a problem if you only have one. In Bubble City, you might have dozens or even hundreds of active bubbles. Hence the name.

Imagine situations like watching the World Cup finals. By instantly creating or tuning in to a "World Cup" Bubble (which merely requires typing in "World Cup" as a search term) you will open a window (or tab) to a "Bubble Window". It's a message overview list where you can see titles of messages from people everywhere watching the World Cup. These can be watched in a passive mode, or in an active mode where you may contribute your own messages. To view the details of a message, to inspect the author, to see more of the thread that the message was posted in, or to reply to a message, click on the message title to open a "Thread View" in a new window or tab, as described in what follows.

Besides real time events, I designed this system to become a place for professionals of all kinds to share information they find useful and to learn about everything that may be relevant to them in their field by participating in real time conversations with peers. Leukemia Researchers, Touring Musicians, and Chefs would all be able to very quickly create their own radically different streams and engage in conversations with peers.

And there will be no hate or advertising in this Bubble. Because UM1 (The machine that Understands language in BC) knows what advertising and hateful speech look like just as well as any human does and it will not let such messages reach you – unless you want them to. If you want to see ads for Viagra, you will need to create a Bubble for it by asking for it.

One way to think of this mode of operation is that BC acts like a search engine, but instead of returning a list of documents, each search query returns a new window or tab containing **one single chat room** where (nearly) everything discussed is about the topic you searched for, and the information keeps coming, in real time, and appending to the bottom of the message list.

Another thing to note is that the system uses 100% pull (until you get to the Thread View). You will not see anything you didn't ask for in Bubble View.

But Threads are not tied together by topic; messages form a reply-to tree, so you may see more numerous off-topic messages in Thread View than you will in a Bubble View, but there are several reasons even this will be unlikely, even before you apply any swipe-tunings, and users can mitigate this with optional "Safe" filters.

So, to summarize, nobody can push any message into any Bubble where that message would be out of scope, and therefore nobody can push spam to large numbers of uninterested users.

The only way to reach a large audience on Bubble City is to post something interesting.

## Users are de-emphasized

Many of the problems and ills of modern social media can be traced to misdirected incentives. Rewarding authors with "likes" is useful in social media, where other users are used as proxies for message topic selection.

BC is trying to seriously de-emphasize users. If you find yourself following a lot of users (by creating special bubbles for each of them by nickname) then you are "using the system ineffectively" and should find a real social medium where users matter a lot more.

If you have a hard time imagining this kind of a system, consider a version of Twitter that only shows texts and threads, not the authors of the messages. How would you adjust your Twitter use if you could not follow other users? You can still search for tweets by text, and you can retweet and reply to any tweet, and the chained replies form a tree. These things are mostly true in BC also. BC will show author nicknames but they are not supposed to be important. What matters is ideas, and using Bubbles are how you follow ideas. This may take some getting used to.

This has impact on the user experience. If these are unacceptable surprises to you, then you are still thinking of BC as a Social Medium.

We haven't spent much thought on the issue whether we should support private messaging to other users... at all. But if we do, they may (possibly optionally) be subject to some Bubble like limitations. But this is getting into distracting details.

## No Like Button and No Dislike Button

It is a BC principle that author feedback in any form increases the risk of people finding ways to game the system. Especially bad would be any micropayment based monetary reward for creating messages that many people might want to read, whether true or not. If you post a message in a Thread View, you will see the reach counter (how many people have seen your actual message) for your post increment in real time. This is the only feedback you get as a poster, unless you get replies to the message.

## More Like This

A "More Like This" button or swipe operation informs the system that you would like to see more messages like this one. It does not provide any feedback to the author, only to the system, and to a first approximation, only affects the filtering in your own Bubble, and not anyone else. The exception is that we track "More Like This" events to identify popular messages for the Popular pseudo-bubble.

Right-swiping on something (a message, the nickname/handle of an author, or a phrase in the message -- there is special UX magic to allow this) will instruct UM1 that you would like to see more messages (or posts from the author, or more messages containing the concept/phrase you specially selected, respectively) like the message/author/fragment you are right-swiping.

Social media sometimes conflate a "More Like This" button into an author feedback "Like" button. This is a serious mistake made by some but not all competing message routing systems.

## Fewer Like This

A "Fewer Like This" informs the system that you would like to see fewer messages like this one, fewer messages from this nickname, or fewer messages containing this specially selected concept, respectively, depending on what you are swiping or selecting. This operation affects your message stream from the moment you click the "Refresh" button but does not provide any feedback to any author of any message in the stream, only to the system, in order to guide its future selection of messages to show you.

# Message Routing By Content

From the time of the Republic of Letters until today, following competent people has been the main way to improve the quality of your input stream. But today we have Machine Learning technologies like Deep Learning and Organic Learning that provide a useful level of Natural Language Understanding (NLU) in our machines.

These "AI" style Machine Learning technologies allow the service to **Understand the message content itself** and to **make a decent estimate**, based on individual user's Bubble definitions, **whether the contents of a new message would be welcome** in their Bubble or stream.

This is a totally novel concept in this space. I patented a messaging system based on these ideas in 2007 while at Google, and the patents are assigned to Google, but they are not in the habit of enforcing patents aggressively and the core functionality will be based on UM1, which is not in the patents.

Google has not built any service based on these patents as far as I know.

## Moat

"Organic Learning" and "UM1" are my own novel technologies created exactly for message content based classification and routing. These technologies are our moat against competitors.

They are discussed on my [Experimental Epistemology site](#) in [Chapter 8](#) and [Chapter 9](#) respectively.

OL can learn a useful amount (enough to pass a simple non-adversarial classification test at 100% correct) of any language on the planet in five minutes when learning from a 5MB unsupervised UNICODE corpus on an old x86 laptop without using a GPU. It dumps a language Model to be used by UM1.

UM1 can classify messages at about 1 million Cps on a reasonable cloud host to billions of categories. Category definition is done by client software at runtime, which is novel.

## No Censorship

Services have to abide by the laws in their jurisdiction which sometimes means that certain classes of messages should be pre-filtered. Hate speech, speech inappropriate for children, religious proselytizing, and political agitation can all be filtered away at the user end of the transaction. The user may indicate that they want to subscribe to the Safe stream as opposed to the Raw or Popular streams as their starting point for using the system. Basically, message selection is implemented as a cascade of NLU filters that are completely under user control. The "service level input stream" is called "Raw". The default when any user signs up for the service is the chain Raw → Safe → Popular → UserMoreOrFewer where the last step is initially empty; the users "more like this" and "fewer like this" clicks will affect the UserMoreOrFewer filter in a transparent way. A fresh chain like this appears whenever the end user wishes to create a new Message Viewer Window so each topic/window can be fine tuned separately.

Removing both Safe and Popular is not recommended for average users but is absolutely allowed, subject to applicable laws.

In the US, the Service can (likely? I am not a lawyer) be legally viewed as a Common Carrier like a telephone provider which means the Service will not be legally impacted by users using the system for illegal purposes.

But to the extent permitted by law, there should be no censorship of posted messages. You can use the system for illegal purposes by for instance posting about drugs for sale and people wanting to buy those drugs would be able to find those messages.

And so would the police. Without paying a penny.

## Only Pay To Post

Account creation is free. Reading is free. Tuning your stream is free. You pay a penny to post.

First time you want to post the system ask for a credit card etc. to charge \$10. For this you get a purse of "postage stamps" for 1000 posts which will expire in one year if not used up. Discounts and promotion games can be played with both the sizes of purses and the cost of a purse, and with expiration times. All purses are tied to their creating Nickname which the end user cannot change and users cannot move stamps around between differently nicknamed purses. These rules serve two purposes: They discourage various spam-like behaviors and if attempted, we simply make more money. The system is spam-resistant at other levels as part of the design.

This is one of the more powerful things we can do. Spammers cannot afford to post thousands of messages and cannot afford to create thousands of accounts to post from. And people will start being more careful with what they say in a post if it costs a penny, raising the quality of posts.

And this is **sufficient to monetize** the project. Twitter sees 500 million tweets in a day. It might drop to something like 10 million tweets if everyone had to pay for tweeting. That's still \$36M per year. Doing better than that is easy especially if you can target every language on the planet (which Organic Learning will indeed allow).

Importantly, there is **no cost and no effort to a user to start using the system** to read messages, which speeds up adoption rates a lot. Once you have something important to say, a penny looks like a good deal. We will warn users that this will happen: They click on a message to reply to it, they type in a reply, and when they click post, the system says that their purse is empty and they will have to enter a credit card number (or equivalent). The user will lose the post if they do not pay. This only happens when the purse on this nickname is empty.

**Marketing Slogan:** "If your post isn't worth a penny even to you, why would we want to read it?"

Many users will never post. We don't care, see the above slogan. This significantly raises the overall quality of posts and people will feel they get more out of this kind of service than from the competitors.

This is Novel (in this space, although it has been proposed for many services including email)

## Reader Anonymity

This is the default since anyone that creates or enters a nickname (not currently in active use on the system) and password can create a (possibly new) read-only account to read anything they want, create any bubble they want, and to tune those bubbles using swipe-tuning. The account (defined by the nickname) is required to persist the state of their reader windows and their Bubbles, including tuning information. Accounts dormant for 3 months are deleted and the nicknames released.

## Poster Anonymity Using Nicknames

This is desirable in some situations and there are many tradeoffs involved including following local laws. A two tiered system may work, where each posting user has to validate with at least a credit card to pay for posts (if they cost money) or a subscription to the service (otherwise). So the system could follow the money back to the posting user and would yield that information if so requested by a subpoena.

But the system could otherwise prevent users from accessing any other users true identity. Account nicknames would be unchangeable but any individual could start multiple accounts to post from under different nicknames. And pay for each one using the same credit card, if they wish. This is to discourage nickname proliferation but is not explicitly preventing it. This would become very expensive for account-creating spammers, and the system would intelligently remove duplicate posts resulting in every end user seeing the post at most once, even if spammed from a thousand accounts with different nicknames paying \$10 each.

# Swipe Tuning

Tinder uses a swiping motion on a user picture as a like/dislike signal.

In a message app running on a swipe compatible device like a phone or iPad we could use that for really rapid feedback aimed at tuning our input stream filters.

If I swipe left on a title of a message, I want fewer messages like this one.

If I swipe right on a title of a message, I want more messages like this one.

If I put my finger on a word in the text, the word will highlight and I can swipe the word left or right to indicate whether I want more or fewer messages with that word in it.

If I keep the finger down for a moment, the selection will expand from a single word to a semantically useful phrase around the word and I can swipe left or right on that phrase. If I point at "Biden" it would expand to highlight "US President Joe Biden" or whatever the surrounding phrase is, allowing me to tune the feed up or down depending on my preferences. The growing highlights are supported by the UMI Service's Understanding of the language.

If we do not have a swipe-compatible device we can use the explicit selection method: We use the mouse to highlight a phrase, in full words, from right to left<sup>1</sup> and then we can click on separate "more" or "fewer" buttons.

We want to allow you to fine tune your Bubbles in these simple ways absolutely everywhere.

You may have a well tuned stream working for months and then some new kinds of garbage starts appearing in your stream. Just swipe left on a key phrase. Hit Refresh button to instantly update your stream by removing all things related to what you just disfavored.

At any point you can ask the Service to temporarily show you the messages it has filtered out and if there is any message you would have liked to see in your regular stream, swipe right on the message or the phrase you would like to see more of.

# Pacer

Message streams may be very voluminous. No human can read all posted messages, and cannot even read all Popular messages or all messages in their stream if they have many Bubbles. Social Media will have to prioritize messages and discard those that have poor engagement.

It would be an awful user experience if messages just scroll away before we can read them. The Pacer is essentially a speed control that indicates how many messages the user wishes to see per minute. It starts at something like ten messages per minute which means a new message every six seconds and can be tuned up and down.

Which messages are shown when 99.99% are discarded? We can attempt to show the most popular messages or the messages that best match our more-or-fewer filters. But the simplest implementation is to pick messages from the stream at random to fulfill the view rate set by the user.

This means that a convenient way to enjoy an app like this is to open the front page and just read whatever is scrolling by at your set speed – whatever you can keep up with. **It will be like watching television**, if you do not intend to interact.

But if you see an interesting post slowly scrolling up the screen then you can just click on it to see that message, in its thread, in a new "Thread View" window or tab

This feels to me like a very comfortable and coherent user interface design.

---

<sup>1</sup> The two selection directions can optionally be used to indicate different purposes for the selected text

## Sample User Experience

Here is an example of what the user experience might look like.

Suppose you want to follow an ongoing sports event, like some World Cup.

Use the Search Feature: Enter [world cup] in the search box and hit return. A new window (or tab, depending on your preferences) opens. You have created a "Bubble", and it is displayed in a "Bubble Window". In this mode, Bubble City is used a bit like a web search engine like Google or Bing, but it is designed to be used for real time messages. The result of a search on Bubble City is a topic specific chat room (just one, not a list of results as in web search) where people only discuss the topic you searched for. No matter what it is.

All bubbles open in a new window (or tab). This one will have messages about the World Cup, going back as far in time as you have screen height for, and can be scrolled.

You can read these messages to catch up. In low-traffic bubbles there may not be many at all, and some messages may be quite old. Messages you have already seen will be displayed in gray. You can scroll back in higher-volume bubbles. Or just mark everything as read.

## Sample User Experience : Swipe Tuning

You start by browsing message titles in the Bubble View. You swipe left on those that are irrelevant to you, or possibly spam, and they turn a reddish gray, but stay around for now. There may be "collateral rejection" of other messages on the same topic also turning reddish gray. You swipe right on the best stuff and such messages (and associated messages on same topic) may get a green highlight. There is a button for refresh, that redraws the screen, discarding everything that was labeled as irrelevant, and accepting the green highlights, which may mean other messages similar to the ones you have right-swiped may appear.

If you are on some device that cannot use swipe then you can use buttons labeled - (Minus) and + (Plus) on each message to deprecate or encourage that message, and any similar messages.

You can now click on any message that looks extra interesting. Another window/tab opens and in there, this message will be in the center. Above it, in reverse chronological order, you see older messages being added in reverse time sequence going upward (towards older messages) and if there is traffic in the bubble, you will see the new messages in the thread for the bubble of the clicked message appear below the initial message.

If you do not swipe-tune, then this bubble will stay true to the spirit of the original message you clicked on, but you can swipe-tune any message in any display at any time, and your swipe-tunes will become permanently part of your Bubble definition when you close and save the Bubble.

Also, when you have opened a Thread view on some single message and its thread, you can immediately reply to the message you clicked on. This is the most common way to post something. Click on something interesting, see its thread, type a reply, and then stay around watching the thread continue after your own message.

To get a topic started, create (enter) a Bubble. If there is no traffic in the Bubble, just post something and others may reply. They may have entered the same Bubble weeks ago.

## Sample User Experience : Playlists

When you close a Bubble View you will be given the option to save the Bubble to a directory under a name you choose. And option (defaults to yes) also adds the Bubble to your current playlist. You can also discard ("pop") the bubble. You can edit your playlist, including sorting your Bubbles in your preferred order.

See also US Pat. [US8015246B1](#)



## Follow Popular Messages

More like this impacts my own stream but it may also be used by the system as a hint about message importance. If many users click on More Like This or Follow a Thread of some message, then that message can be showed to users that want to see popular messages. We will not use popularity for filtering messages unless necessary, because popularity can be exploited by malicious users to improve reach.

## Follow New Messages

Follow the raw incoming message stream. This is more than any human can read so we still need to apply some filtering to the stream. The default filtering is to randomly discard enough messages so that a user can read what remains by using the Pacer.

## Follow Trusted Streams

Messages may come not only from other users but from well known named message streams such as Reuters. Historically, these have not been subject to additional filtering but if we use Message Routing By Content then there is no reason to not filter those using the same filters we apply to messages from other users.

## Follow Friends And Family

Social media are different from chat applications because of the emphasis on people and the use of some kind of friendship graph that is used mainly for message routing.

This has proven itself historically as very popular but I would like to focus on a Service that supports group problem solving rather than social interactions. Implementing a social graph brings along a lot of complexity and will be much more expensive to create and maintain than routing messages, and it may become a problem if we wish to (at least ideologically, if not legally) attain Common Carrier status.

The minimum level, where you can create Bubbles containing all messages from specific individuals and we do not maintain a social graph, is cheap and acceptable as an add-on for marketing reasons.

## ClickBait Headlines Completely Eliminated

We can totally avoid clickbait headlines. I reserve disclosure of this mechanism to NDA level discussions. This solution also simplifies or solves several other problems in Service design and implementation.

## Traffic Logging for Machine Learning

All posts are public. All posts need to be kept for at least some time in a repository so that they can be displayed to others as required, but this is designed to be a real-time system so we may impose limits on history length.

Because everything is public, users have no expectation of privacy for this content. We will use messages in our message repository as a corpus for training future versions of the system. We will not sell these corpora to outsiders but may share them with (corporate) collaborators for language research and product enhancement purposes.

## No Voting On Messages

SlashDot and Reddit use voting. This should be unnecessary and is mostly a nerd-favored solution to noise problems.

## No Need for #HashTags

#HashTags were invented by both IRC and Twitter users in response to the service lacking a decent semantic search capability.

We cannot prevent users from using these features but we can decide whether we want to explicitly support tagging by (for instance) providing a tag canonicalizing service.

IRC and Twitter users use this

## Edit Button

This seems important to some people, but editing would change the Bubble definition and this causes trouble in many places. This is definitively not going to be in the first release.

## No Subscription fees

This substantially lowers the threshold for joining.

## No Advertising

This would substantially lower the enjoyment of the system. We want to monetize this service exclusively by selling postage.

## Links

[Monica's Patents assigned to Google.](#)

[Introduction to AI Epistemology](#)

[Videos of Talks](#)