

# Wrex + Holochat: The Ultimate Chat Paradigm

design manifesto

deicidus

In the burgeoning world of Holochain app design, a big question mark is the form chat will take. So far there is Clutter, a fully-decentralized (cat-themed) Twitter clone, and the barest beginnings of Holochat, a fully-decentralized Slack clone.

These apps will likely become major gathering points for programming and design efforts in the coming months, as the Holochain app developer community seeks to make its communications technology highly functional, beautiful, and usable, so that using Holochain for secure, peer-to-peer chat becomes attractive to not only Holochain developers and enthusiasts, but also a mainstream audience.

## Self-evolving dog food

Conway's law states that *"organizations which design systems . . . are constrained to produce designs which are copies of the communication structures of these organizations"*. Right now, Holo uses Mattermost as its primary communications medium, so there is at least some influence for Holo's organization structure to resemble the enclosed teams containing many side-by-side channels which Mattermost implements. To evolve the Holo organization, we must simultaneously evolve our communication tools, so Holochat emerges as a critical Holochain application which can be developed to provide Holo control over its own communications infrastructure, and to give the organization the ability to upgrade its capacities by augmenting its communications platform with new features or modes of relating and organizing people and content.

By ["eating our own dog food"](#) in a tighter feedback loop, the upgrade and self-feedback mechanisms of the organization become intensified, and Holo becomes more agile and quicker to respond to environmental information. At the same time, the core infrastructure that Holochat represents receives heightened focus and more frequent upgrade cycles, pushing forward this critical infrastructure and ["mother app"](#) and quickly evolving it into an advanced, decentralized tool for communication and increasingly decentralized organizational management.

For this reason, Holochat is important to do first and to do right. Building cybernetic loops into the design process of Holochat *early* in its evolution is critical for beginning the pattern of a self-evolving system and embedding this pattern into the design process of both Holochat and the continual (re)design of Holo's communications infrastructure.

## The spectra of chat applications

Ideally, all chat messages, no matter which app they originate from, could be conceived of as being conceptually the same kind of digital object, that of a “chat message”, “sent communication” or “shared document”. But, looking at the number of chat paradigms and formats that exist already, there are a number of variations and limitations which make the different kinds of “chat messages” seem, at first glance, to be somewhat incompatible with each other:

- Twitter messages are limited to 280 (formerly 140) characters.
- Emails are long-form and contain messy HTML formatting and embedded or attached images.
- Chat messages are enclosed within their platform and often contain platform- or app-specific content, metadata, or features.
- Chatrooms afford and reward synchronous (live, real-time) communication, while email, forums, and SMS afford asynchronous (unpredictably-delayed response) communication.
- Some chat applications (such as Telegram) are making overtures towards affording both synchronous and asynchronous messaging within a single interface (for example, through message-read notifications, online status, or the ability to post threaded replies to old chat messages). However, the code to supporting both asynchronous and synchronous communication within a single unified GUI has not yet been fully cracked or implemented.
- Some forms of chat, such as live chat-rooms, afford “rooms” of shared audibility; some, such as email, afford threaded conversations and need additional tools (mailing lists) to resemble rooms or “groups”; and some, such as forums, contain multiple modes of addressing, grouping, and notifying which allow threads or content to move between public and private (however, these moves are made using a grab-bag of unrelated features such as replies, private messages, boards, thread reply notifications, and groups).
- Answering chat messages across multiple platforms—often from the same person!—is a confusing exercise in detective work, often involving having to remember which of several chat programs is preferred in which context, having to check multiple places for messages, or having to reconstruct a probable timeline or sequence in which the same or similar messages were sent on multiple platforms.
- Clutter is being developed on Holochain as a Twitter clone; Holochat is being developed as a “Slack clone”; if we unify messaging conceptually and informationally, will the labor of building these two apps separately become redundant?
- Rich media chat apps, such as Instagram or Snapchat, use photos as the primary message type.

As you can see, there are several overt differences in format, message length, metadata, timing, and messaging rules that make messages from different applications and paradigms hard to classify together. These represent inconsistencies in the current historical concept of “chat messages” or “digital messages” as it is implemented in the current landscape of mainstream chat and messaging applications. Bringing coherence to the concept of a “message” will enable us to weave a unified fabric of interpersonal communication which can include all kinds of chat and long-form messaging in one paradigm. Cleaning this up will, in turn, bring clarity to how all forms of interpersonal messages can be presented in one or several unified GUIs, how communication between different kinds of apps and messages can be accomplished, and—most importantly—will allow focus to shift from chat platform to the higher-order, agent-centric questions of “Who am I talking with?”, “What conversation is this a part of?”, “What topic are we on?”, “Who can hear this?”, “What actions are needed of me next, and when?”. In other words, focus can shift from the information technology and the data-centric parameters (platform, format, headers, metadata) to the context-sensitive *meaning* and actionable, sequenceable prioritization of message-handling activity on the part of the human agents utilizing the system.

Therefore, a unified chat paradigm must have a way of comprehensively considering any type of Holochain content as a “message”, and relating these messages together within a single communicative fabric. Here are several polarized dimensions (or axes) which begin to describe a space which could conceptually and digitally contain *all* forms of messages:

- Short-form to long-form messages (Twitter has short messages, chat messages are longer, email messages usually the longest)
- Asynchronous to synchronous messaging (various expectations regarding “reply requested/required”, “expected reply latency”, “are we currently in a conversation or just asynchronous back-and-forth”, “urgency/priority set by sender” and so on)
- Private to small-group to public messages (1-to-1 conversation, groups/rooms or webs of conversation locality, up to fully public)
- Unactionable to actionable messages (“just read this and reply if you feel like it” to “take this specific action” or “skim or read this specific document”)
- Please-share to shareable to please-don’t-share messages (Ads, Twitter, and private gossip, respectively)

Many more axes could be imagined, but using the axes above, we can imagine almost any conceivable chat message on current platforms being located in the same space, located somewhere along each of these dimensions. This allows all chat messages to be conceived of as the same datatype or object—a type of object which, I will next attempt to convince you, is best conceived of as a hypercard.

## The origins of Wrex

Wrex was originally conceived of as “the Wrec”, a tool for receiving and prioritizing recommendations (hence the name) from others. I designed this tool for my personal website, where I hoped it could be a kind of public inbox or personal subreddit where people could leave one-line recommendations of books, articles, links, TV shows, movies, video, people, places, organizations, projects, tasks, music (albums, artists, or tracks), games, or any other content which they pleased. Each line entered by a visitor to my website would be entered into a publicly-visible table, containing (depending on the media type) fields for the name/title, author/artist, reason for recommending, name of recommender, link, date of recommendation, etc. I would also add the contents of my many (30+ topical) booklists, movie list, videogame list, list of favorite authors, etc. to this unified public catalog of all of the media and publicly-accountable tasks in my queue.

As a kind of personal subreddit, it would also be up to the public to help me prioritize these tasks and recommendations by upvoting or downvoting their favorite recommendations or the recommendations of others. This would allow me to sort the list and easily see which things others wanted me to read or do the most, first. I envisioned creating a 24-hour timer on voting, so that, if someone was especially excited about me looking at their link or performing their task as soon as possible, they could revisit my website once per day and upvote their pet recommendation every day, quickly pushing it to the top of the list. This would allow expression of the very real phenomenon of eagerness in recommendation or urgency in tasking, fairly balanced with the actual amount of attention someone was willing to spend in order to get my attention. Additionally, people could leave donations on certain tasks, an expression of importance and an environmental feedback I would surely take note of and include in the ultimately personal, overall, actual decision one always makes of which task to take on next (introducing donations or bounties on tasks makes me into an “open public servant”). As the Wrec on my website (named because of the immense vastness and relative messiness of the unorganized plain-text lists of recommendations and tasks I keep) began to become sorted by self-selected crowdsourced labor (hopefully, those who most personally cared what I spent my time on), it would more-and-more come to resemble a coherently-organized, somewhat hierarchically-sorted ideal sequence of meaningfully-sequenced personal tasks within my world. This, I call “the Stack”, and the Wrec helps one to reorder and maximize the coherence of one’s personal stack.

By adding trust metrics to the mix, it would be possible to modify the votes people made on the sequence of stack items by their distance from you in your network of friends. Or colleagues, or club-members—trust can be context-specific to different topics, conversations, projects, or other localities imaginable in the metadata. However, overall, we each (as attentive agents) have a global “current context” which holographically contains

other contexts as list-items which, when mechanically arrived-at and executed in the stack, unfold into a new/modified context.

The way this process can develop local, grassroots action (local comparative prioritization of list items or groups of list items in relation to each other) into global prioritization or order of deference allows the Wrec to potentially contain intelligent programs of investigation, directed action, and operations management. Additional narrative- or thematic-program-authoring tools could be developed to expand the capacity for context-sensitive intelligently ordered sequencing over time.

For this reason, I stumbled upon the name Wrex, because “rex” means “king” in Latin. Wrex is the necessarily-true abstract algorithm which already governs the actions anyone takes, and by capturing this algorithm visually and behaviorally in a cybernetic process between person and interface, we accelerate and make visible the process and its rules. This allows the rules we use to program our attention to be made explicit and put under the control of each agent via decentralized, GUIfied protocol, in conversation with their audience of patrons and the limits of formalizable metadata plugins.

### **What’s the action? Tasks vs. recommendations**

*“An abstract list of my next actions, which either changes depending on the context or always remains destinally, perfectly unchanging?”* It sounds very abstract, like the list could contain almost anything—and, in practice, the action attached to each stack item in Wrex could be almost anything.

This reveals the distinction between recommendations and tasks, and that recommendations can often be considered to contain an implicit task, or conversely, that recommendations always-already imply the task of consumption of the recommended media or experience. Often, no explicit task is given with a recommendation, so the amount of emphasis or attention one should give the recommended item is unclear. For example, I receive several links to long articles per day, so it is unclear whether people expect me to *skim*, *read*, or *peruse* these articles, or even simply *know this exists*. There are a number of different, nuanced levels of consumption or action we could attach to every file or recommendation we send to each other, and I think that including this tasking information on every recommendation would bring a lot of clarity to the metadata of Wrex and the ability of the system to intelligently sort its content.

Therefore, Wrex is an idealized image of the attentional economy, with *task* or experience-consumption chunks being the root data type.

## Hypercards as autonomous labor packets

This system gets very interesting when everyone has a Wrex and people start sharing and sending stack items between their stacks. Each stack item can be thought of as a *hypercard*, a holographic card which is itself a small Wrec/Wrex/Stack, thus making a Wrex a holonic, recursive structure of stacks upon (or within) stacks.

Hypercards can contain anything—it can be a file, it could be a chat message, it could be a person's contact information, or it could be an album of music. Each of these hypercards comes with contextual information, including implicit or explicit tasking information—for the previous sentence, the tasks might be, respectively, “open this file and become familiar with its contents”, “read this and mark read”; “contact this person to help you with project X”, and “listen to this”).

Hypercards can be distributed GUIs for Holochain apps. There are many different models of networked interactivity between people's individual copies of the same protocol-based hypercard which could be imagined. For example, many people could possess a hypercard containing the same shared document, or many people could each contain separate copies of the documents—or, in Fractal Wiki, many people might contain a hypercard containing somewhat-shared, somewhat-distinct or mixed versions of the same document. A hypercard containing an album might be cryptographically signed by the artist, and contain an onboard artistic GUI containing album art, metadata, music playback controls, and lyrics—as well as metadata through a standard music API which allowed the music to be played through or copied into other systems.

Each API (such as might be exposed by each Holochain mixin) can be implemented as a hypercard. If you flip a hypercard over, the “back” of the hypercard can be conceived of as a control panel for managing all the hypercards which combine to make up the GUI “front” of the hypercard. Each hypercard snap-in installed on the back of a typed hypercard augments its capacities or modifies its GUI. And every hypercard could have not only a front and a control panel on the back, but also an API defining how that hypercard type acts when plugged-into the back of a *different type* of hypercard.

This creates a holographic snap-in system where each hypercard coincides with a relatively standardized GUI object. This indicates that a unified, holographic GUI is emerging from various cybernetically self-constructing sub-components which are coalescing as converging standards in software architecture, GUI design, peer-to-peer storage and sharing of content, and interpersonal communication.

In this totally atomized, maximally agent-centric, distributed economy, each actor is a holon implementing a personal stack or Wrex, and chat messages are, like any other

activity performed in sequence on the computer screen, a micro-task which can be sequenced rationally by the computer, according to contextual clues such as timestamp of previous message, closeness within social network, or currently-avowed or scheduled task or activity. This allows the prioritization of tasks to become increasingly automated, according to the parameters, questions, and contexts set by the user, and the sorting mechanisms invented and developed as sorting snap-ins by users of the platform.

This creates a sense of place, space, and time in using the computer. For example, rather than being radically exposed to and always-online for all Telegram conversations at once, a contextual stack would allow one to “wander in” to one or a few chatrooms at a time, and then “wander out” again in one’s personal context—and visual or other interface cues would gently provide this presence information to the people in those chatrooms. This would prevent many annoyances, such as not being sure who is currently “listening”, or of having an active conversation go suddenly dark because other messages distracted one (or often, both) conversation partners. Because our computers do not track tasking or human-relevant prioritization information at the level of switching windows or sequencing incoming notifications, they cannot yet be our allies in maintaining or negotiating context, project, or local space. By including this information, Wrex allows for collective prioritization as well as smart individual prioritization to become automated—and this automation increases, rather than decreases, both personal and collective agency, because it allows excess information to be hidden or deferred according to our preferences, rather than overwhelming the context-management, multitasking, and information-prioritizing faculties of our brain with a glut of information and constant context-irrelevant interruptions. Wrex maximizes digital agency while at the same time helping to promote internal self-alignment in agents by creating an economy of content-sharing, protocol-synchronization, and collective tasking and prioritization which allows crowdsourced wisdom to combine with personal preference and feelings of priority and enjoyment non-coercively.

## **Work without jobs**

In this environment, *labor becomes autonomous* as hypercards that move through the network of stacks and seek their best executor. Autonomous labor, as deterritorialized intention, “escapes” its initial origins in a person’s idea when that person shares the hypercard with a friend or colleague. With increasingly-optimized smart recommendations/prioritizations, automatic recommenders, and the enormous labor of crowdsourced curation, it is *as if* labor becomes self-aware and holoptically self-optimizes itself within the collective Stack of all human labor.

In other words, once Wrex becomes implemented at a 1.0 level as the cybernetic feedback loops of “eating our own dog food” in a distributed chat-task hypercard ecosystem, labor leaps gleefully into line in the most efficient, effortless, *powerful* (in the

sense of expressive capacity / expressive brevity) sequence possible, based upon currently-collectively-available data and protocols/algorithms.

In this model, someone could have a great idea for a movie they would like to see, and then they could meow this to their friends as a task to “make this movie: <synopsis>”, which could then move around the network (perhaps garnering a cult-like following of bated-breath subscribers or accumulating image macros, comments, or crowdfunded bounties) until it found a film crew with the resources and eagerness to make the film. Then, holders of the original film-idea hypercard could be notified with a high-priority message when the movie was released (hopefully, this hypercard would include a copy of the video file!).

In Wrex, people receive credit for the cards they complete, and so cards and meta-cards increasingly evolve into abstract, Ceptr-like symbols, and increasingly come to resemble qualitative currencies and metacurrencies. Exploring this development of hypercards-as-distributed-currency-interfaces is beyond the scope of this paper.

## **Accelerating the movement**

The cybernetic feedback loop represented by Wrex is the ideal form of the fastest way to accelerate the development of the Holochain app platform and the decentralization and distributed development of the Holo ecosystem of organizations and initiatives. This movement, coalescing though it may be around values such as fairness, anarchism, anti-corporatism, and consent, is nonetheless first of all the evolution of a new economic form of cooperation, including the infrastructure for such coordinated activity.

This is why starting at the (*non*-)center and, through iterative development, building outwards from the chat application (Holochat + Clutter) to other areas of application development will serve best to liberate new capacities and organize these capacities into collective flows of accelerating activity: upgrading the core of our communications infrastructure—our messaging application(s)—allows new expressive capacities to be developed and rolled-out almost immediately to the edges of the network, progressively eliminating the conceptual disjuncts which act as unspeakable catches in interpersonal relations and deep-process. In other words, by developing the core capacity to collectively imagine, communicate, distribute, and prioritize our tasks, we make the entire economic engine more efficient—and this efficiency becomes codified as intuitive expressive capacity in the user interface.

Ever since iPhone implemented the pseudo-holonic navigation of its apps and the home screen, a slouching, half-mobile version of this ouroborous-like hypercard stack has been accelerating into existence. However, Apple’s definition of an “app” and their app-centric unitary control mechanisms on the App Store, as well as an ultimately

traditional data model, have prevented a full acceleration and launching of the cybernetic feedback loop (which would allow the apps economy to transcend out of Apple's stranglehold control and into self-management less amenable to capitalization). By operating this same kind of loop on open-source software and a distributed GUI, the users of the system can begin to collectively upgrade its capacities and coalesce a grammar (or shared meta-grammar).

## **Specific software recommendations**

What does all of this imply? There are several key takeaways:

- There should be communication between the core paradigms of the Clutter and Holochat apps—both should communicate within a shared construct of “messaging”. This messaging construct could be a Holochain mixin establishing basic metadata parameters and compatibilities amongst different modes of chat/messaging/sending.
- Various applications, application intentions, and GUIs are evolving towards a unified holographic GUI with pluggable snap-in hypercards which can occur in multiple appearances and which can modify each other. The sooner we take into account and begin to express the holographic grammar of this unified interface, the less redesign work we will have to do later in the process, and the sooner our entire system will accelerate into glory.
- Wrex, conceived of as a Holochain mixin supporting individual, interpersonal, and collective prioritization and signalling of prioritization and tasking information, is a critical app to upgrade collective intelligence and formalize the ability to design plugins which invent new and smarter ways to sort, conceptualize, and collectively (and fairly) distribute labor.

This brings us to specific design considerations for a Holochain-based chat application. Is Slack really the best model? Slack arose very recently as a historically-specific, corporate IRC clone which capitalizes on a candy-colored interface and labor-controlling features in order to create enclosed and isolated servers on Slack Co's computers. It has many subtle design characteristics which undermine good communication, such as traditional binary-membership channels, global online/offline status, and interface design which is not user-modifiable. Because it is closed-source and for-profit, Slack's GUI contains many affordances designed to enhance corporate culture and team enclosure, and its limited history (for unpaid users) and limited export options for message histories cripples Slack in fundamental ways that obscure potential amazing new ways in which we could conceptualize distributed history and distributed conversation.

In contrast, let us consider a few key advantages Telegram has over Slack. First, Telegram does not have enclosed team-servers—all conversations (including groups, individuals, and conversations with bots) are in a single list. Second, Telegram messages

are increasingly flexible and interactable, resembling hypercards, and there are multiple message types, such as stickers which allow anyone to copy the sticker set by clicking it. Third, Telegram supports rich threading, forwarding, and sourcing features. These features support open communities which cross-pollinate more easily than the more enclosed, text-oriented, linear conversations more afforded by Slack.

The first advantage of Telegram also reveals a weakness: All messages in Telegram are in a single list, making it a somewhat overwhelming experience to scroll through the list of conversations to reply or find a particular conversation. A mix of personal, work, and group conversations, my Telegram contains so many conversations that I can't keep up with them all, and often miss notifications. This flatness and lack-of-context is the opposite of Slack's rigidly-isolated multiple "servers". What we need is something in-between: a common interface which opens up a vertical, abstract, or meta space of organizing and prioritizing above the level of messages, but below the level of groups. This *infra*- level of temporality or meta-organization can be largely automated, but it is the fabric that allows the computer to begin to know our sequencing and our contexts.

So, my overall recommendation is to create a Holochain chat application which is holistic, which can synthesize chat paradigms from multiple modalities of messaging, and which is based around a GUI which includes some kind of new pluggable modalities of meta-object temporality (prioritizing, sequencing, recommending, arranging, organizing, sorting, stacking, destacking) which are collectively upgradeable. Finally, I recommend abstracting the idea of "message" to include digital objects sent as task items, with a chat message being a subset of object-task packet which include primarily a message text and the task "read and mark read" or "read and reply". This synthesizes chat-space with file-space, and moves towards fusing this combination with GUI-space/window-space/app-space. This further evolution of a synthetic holographic GUI emerges as a collaboratively-constructed grammar of standard practices codified as increasingly-coherent and inter-embeddable holographically-typed hypercards, which themselves evolve together as increasingly higher-order glyphs of interactability, analogous to Ceptr-level rich semantic protocols or linguistic tropes. These pattern languages evolve as powerful glyphs which are also increasingly expressive, holonic protocols. These glyphs, in turn, evolve a diagrammatic logic or *diagrammar* which results in the eventual emergence of moving symbols whose visual parameters meaningfully and intuitively correspond to their numeric, content, or affective parameters.

## **The symbol of Holo**

The overall symbol for this modality of decentralized self-organizing of content by many agents is *wheels upon wheels*, or a fractally-spoked wheel of increasingly-small circles or bubbles (spokes made of lines of adjacent circles), each of which could be zoomed-into

to expose a new version of the entire holonic symbol: another wheel of spoked wheels of bubbles.

This structure, when it “turns” like a wheel, turns not only the entire structure but also every level, including spinning the small circles which act as the “atoms” of the whole system, producing a “gears within gears” mandala of transcalar vortical action (a strange loop). This drives the acceleration of acceleration and cross-pollinates the entire system across scales and levels, resulting in the transversalization of ideological content as well as the distribution of memetico-political power throughout groups and levels of the system. This creates balance between the needs of individuals and the needs of groups, through the mediating objects of the autonomous labor hypercards, which abstractly churn in data from the Outside and process this data into a collective Stack. This assemblage made of bodies, screens, computers, networks, and autonomous labor resembles a large dance, making the “pattern of patterns” the symbol of Holo, and a decentralized/decentralizing Leviathan (or Cthulhu) the symbol of this embodied, transpersonal dance which results from this hypercorrelation.

How do we build a better dog food bowl? Let’s start by making it out of dog food, so you can eat the bowl, too!