

The Live Web Series

The Personal Channel

The Extraordinary Benefits of Communicating Via Personal Clouds

Drummond Reed, Managing Director, *Respect Network Corp.*
Phillip J. Windley, Ph.D., Founder & Chief Technology Officer, *Kynetx*

June 2012

The Live Web Series sets forth a vision for the future of the Internet and our interactions on it. This paper is the second paper in that series. This paper builds on the previous paper, *From Personal Computers to Personal Clouds: The Advent of the Cloud OS* (<http://www.windley.com/cloudos>) . We **strongly recommended** that you read it first. Future papers in this series will build upon the ideas presented here to show how personal clouds form a foundation for richer and more satisfying online applications and experiences.

Executive Summary

The personal cloud—a cloud-based virtual computer controlled by an individual—is looking as inevitable as the personal computer was in the 1980's. But while new applications like spreadsheets and word processors drove the adoption of personal computers, it will be new communications capabilities that drive the adoption of personal clouds. Links between personal clouds—called a *personal channel*—can simplify, automate, filter, and protect communications in ways that have never been possible before. Personal channels combine the best features of email with the best features of social networks while eliminating key drawbacks of both. They will also introduce new levels of security, privacy, and control over personal data that will enable trusted relationship management between people and organizations everywhere.

Table of Contents

- Introduction: The Power of Connecting
- Personal Channels
- Benefit #1: Access Control and Volume Control
- Benefit #2: Lifetime Portability—Channels Never Break
- Benefit #3: Secure Delivery
- Benefit #4: Simple, Standardized, Contextual Authentication
- Benefit #5: Scalable Trust with Contextual Reputation
- Benefit #6: Automated Event Processing
- Benefit #7: Intelligent Filtering and Organization
- Benefit #8: Intelligent Notification
- Benefit #9: Automatic Data Exchange
- Benefit #10: Rich Sharing
- Conclusion
- Finding Out More
- Lexicon

Introduction: The Power of Connecting

The Social Web has shown us the power of connecting. Facebook has *friends*, LinkedIn has *connections*, and Twitter has *followers*. And between them they now connect well over a billion people—at least one-sixth of the planet.

These connections form a graph of relationships that are often called “the social graph”. Each connection also forms a communications channel, although the capabilities of these channels vary widely between the various networks.

But all of these networks have one thing in common: *they are proprietary*. Despite all the money and time invested in creating APIs to access them, interoperation and extension is difficult, brittle, and in some cases actually a violation of the network’s terms of service.

The natural progression of the Internet will be to develop a unified, interoperable relationship network based on open standards. This is what happened with the Internet itself in the 1970’s, with email in the 1980’s, and with the Web in the 1990’s. In each case, the network effects unleashed by the emergence of a general-purpose, open standard network completely disrupted the highly dominant players of the time. Just think what email did to proprietary email networks like MCI & CompuServe or the Web did to AOL.

In the case of the Social Web, the rise of the *personal cloud* will lead the transformation from proprietary social networks to a general-purpose, open standard relationship network. Personal clouds represent a new computing paradigm for individuals and are described in detail in our paper [From Personal Computers to Personal Clouds](http://www.windley.com/cloudos) (<http://www.windley.com/cloudos>) .

In this paper, we describe a key reason personal clouds will be so disruptive: they enable a breakthrough new form of communications connection—the *personal channel*. We will cover:

- What is a personal channel?
- How personal channels provide ten major benefits not available through other communications networks, in particular email and social networks like Facebook.
- How these benefits enable *rich sharing*, a marriage of the benefits of email and social networks.

Personal Channels

When they first appeared in the early 1980's, personal computers were powerful tools in their own right. That changed in the 90's with the emergence of widespread network connectivity. Nowadays, a PC that's not connected to the Internet is non-functional for many of the tasks that people perform every day (e.g., email, web browsing, social networking). To test that assumption, just try turning off the network on your computer for a day. And of course, the very newest personal computer—the smartphone—makes connectivity the very foundation of the platform.

Even more so than personal computers, personal clouds are only interesting when they are connected. The connection between two personal clouds—or between a personal cloud and anything else it is connected to—is called a **personal channel**. The network of people and organizations linked via personal channels is called a **relationship network**.

On an open standard relationship network, the attributes, permissions, and capabilities of a relationship are standardized and extensible. Every relationship is a **link**. A link may be a simple one-way (asymmetric) subscriber relationship that does not require involvement of the second party, or it may be a stronger two-way (symmetric) relationship in which both parties may act as publishers and subscribers.

In either case, when data and messages can flow in one or both directions across a link, it is a **channel**. The control each party has over the channel—the terms and conditions to which they agree about how it will work—is called a **link contract**. Figure 1 shows two personal clouds connected via a channel controlled with a link contract. Note that both parties store a copy of the same link contract just like they would a paper contract.

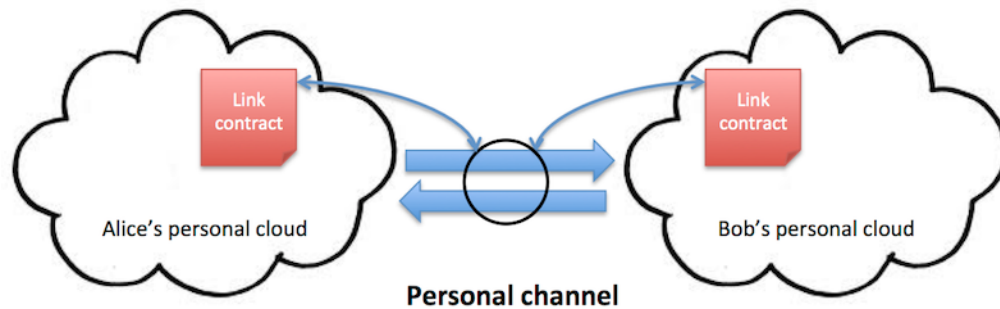


Figure 1: Every personal channel is controlled by a link contract.

Like email, personal channels all speak the same protocol, forming a point-to-point network between personal clouds. However, unlike an email server, whose sole function is usually email processing, a personal cloud is more like a general-purpose computer in the cloud, i.e., it has an operating system that runs applications, processes events, and manages data on behalf of and under the direct control of its owner. So personal channels can be much smarter communications links than ordinary email or text messaging. For example:

- A personal cloud may have any number of inbound and outbound channels. And two personal clouds may share multiple channels for different purposes.
- Personal channels ensure accountability for shared data because the authority to share can be granted, modified, and revoked on a channel-by-channel basis.
- Link contracts are a flexible means of declaring fine-grained access control to data and services. Link contracts specify the nature and behavior of a channel.
- Channels use a combination of the Event eXchange Protocol ([EXP](http://developer.kynetx.com/display/docs/Event+Exchange+Protocol) (<http://developer.kynetx.com/display/docs/Event+Exchange+Protocol>)) and XRI Data Interchange ([XDI](http://www.oasis-open.org/committees/xdi/) (<http://www.oasis-open.org/committees/xdi/>)) protocol that give them [metaprotocol capabilities](http://www.windley.com/archives/2012/03/protocols_and_metaprotocols_what_is_a_personal_event_network.shtml) (http://www.windley.com/archives/2012/03/protocols_and_metaprotocols_what_is_a_personal_event_network.shtml). Channels are a *way of doing something rather than a place for doing something*.
- Channels may pass any type of message between personal clouds, not just text, images, and attachments. Messages may include event notifications, data queries, and data transfers. Messages may also be orchestrated into workflows.
- A channel need not be restricted to just two parties. It may connect the members of a group (e.g., email distribution lists), or access may be fully public (e.g., blogs or Twitter feeds).

In sum, personal channels on an open-standard relationship web can be dramatically more useful to individuals and businesses than ordinary email or Web connections. The following sections explain the ten major benefits of personal channels over other forms of communication and data sharing, in particular email and social networks.

Benefit #1: Access and Volume Control

Internet email is the most widely used electronic messaging system on the planet and yet an [estimated 80% of all email traffic is spam](http://www.maawg.org/email_metrics_report) (http://www.maawg.org/email_metrics_report) . How did we end up in this position? How could the vast majority of traffic in our most heavily used messaging network be junk mail?

The answer is simple: trust was not in scope when the SMTP email protocol was developed. In other words, trust was not an issue when the Internet was much smaller. Most of the server administrators knew each other, or at least knew each other's institutions, and they never imagined a day when the Internet would be so large that spammers could hide in all its dark corners and constantly dodge enforcement in an endless game of whack-a-mole.

SMTP's most glaring weakness and the most important reason spam exists is because email is a **single-channel network**. You have one email address and anyone who has it can send you a message. The second reason is that email is *opt-out*—your address permits *uncontrolled access* by default. That means it's the recipient's job to filter out messages she doesn't want.

Personal channels reverse both assumptions. First, a relationship network is a **multi-channel network**. In this respect, it functions more like a social network. For example, on Facebook, each relationship in your social graph is like a channel. When you friend someone, you've essentially opened a symmetric channel between your Facebook account and theirs. By virtue of that link, you can post on their wall and they can post on yours. No link, no post. If someone keeps posting things you don't like, you can sever the link—close the channel—and they can no longer use it to contact you.

Personal channels work the same way except:

- The network is decentralized, so you may choose any service provider you like. This extends to hosting your own service if you so desire.
- Channels and messages are much richer in scope and type. Banks, for example, are unlikely to use a Facebook “friend” relationship to send you bills or account information, but a personal channel is perfect for that.
- You can have different channels with the same party to better organize and filter messages.

This latter capability gives personal channels *volume control*. Rather than only being “on” or “off” (subscribe or unsubscribe), you have the option to “tune” a channel to the type and volume of messages you want. Don't want every update coming over your son's lacrosse team channel? Tune it to get only game time notifications and parental updates.

Personal channels are also *opt-in*, i.e., *controlled access* by default. This means no

one can open a personal channel with you without your permission. Nor can they even send you an offer of a new channel unless it passes your trust filters (for more on this, see [Benefit #5—Scalable Trust with Contextual Reputation.](#))

So unlike email, where economic incentives drive the generation of spam no matter how great the cost to everyone else on the network, a well-designed relationship network will give every member of the network the controls they need to maximize signal and minimize noise.

Benefit #2: Lifetime Portability—Channels Never Break

A second key limitation of both email and social networks is that many addresses are *not portable*—if you change service providers, all your relationships break.

The same used to be true of telephone numbers, but with the ascendance of mobile phones, the need for telephone number portability could no longer be repressed. In the United States, [LNP \(Local Number Portability\) was mandated by the FCC in 1996](#) (http://en.wikipedia.org/wiki/Local_number_portability#History) .

In fact Internet email has actually always had a solution for email portability: registering your own domain name and administering it to maintain continuity of service when you change email service providers. But not many users are savvy enough to take advantage of this option.

Unfortunately the same option is not available to users of the major social networks—the social graph and data you create with one provider is not portable to others.

On an open-standard relationship network, where every user maintains the data and relationships in his/her own personal cloud, *address portability must be built-in from the outset*. Users shouldn't be asked to accept the risk of having their personal cloud locked up with a single provider for life. Users must be able to move their personal cloud to a different provider (or to self-host it) without breaking any links in the same way that users can switch banks without losing their money.

Furthermore, portable addresses for personal channels have enormous benefits: for the first time ever, there is no reason for a relationship to break just because a party moves or changes service providers. The only time a relationship link ever needs to be severed is when one or both parties no longer want it.

Benefit #3: Secure Delivery

Another key limitation of Internet email is that it is insecure. Numerous attempts to establish high-security, trusted email have been made, and there are thousands of proprietary solutions. But there are simply too many email servers and accounts in

the world—[by one estimate over 3 billion](http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/) (<http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/>) —to do a “forklift upgrade” of the entire network.

Messaging via centralized social networks is similarly insecure, if only because the social network itself has access to all the data. This is why social networks are not used to share banking statements, finance data, or medical records.

By contrast, personal channels on a relationship network can offer secure cloud-to-cloud delivery of messages and data. The precise levels of security depend on the trust framework(s) under which the parties are operating. For example, a trust framework could mandate that a minimum of HTTPS must be used for authentication and confidentiality across all personal channels subject to that framework. As shown in Figure 2, this would set a minimum “hardness” for all channels on the network.

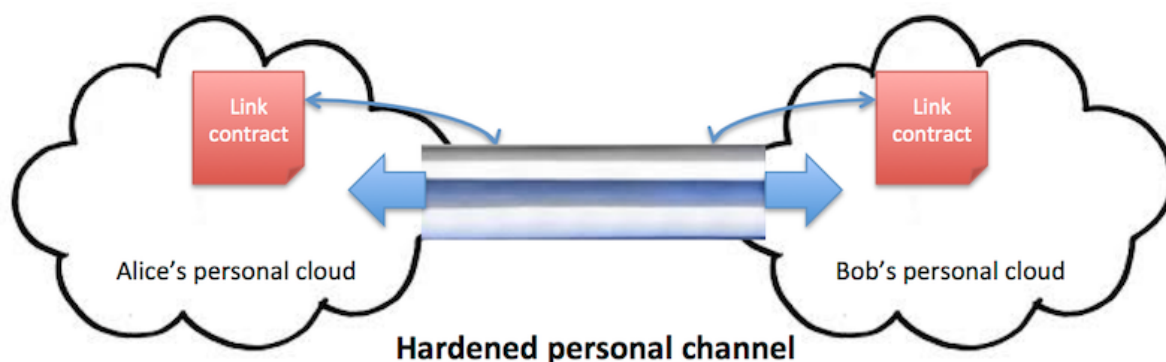


Figure 2: Link contracts can harden the security requirements as needed for any particular channel.

Even better, link contracts may be used to specify additional security requirements that may apply only to specific channels, specific messages, or specific data. For example, extended validation certificates and message-level 1024-bit encryption may be required for transmission of credit card data or confidential documents. Personal cloud providers who are not able to meet these requirements will not be able to service clients who need this level of security protection. This turns security (and privacy) into a market-driven race-to-the-top rather than a never-ending lowest-common-denominator battle over standards.

Benefit #4: Simple, Standardized, Contextual Authentication

The most common form of Internet authentication today is email address verification. The standard technique is called “closed loop verification” because you give your email address to a website, it sends you a verification email, and then you “close the loop” by clicking on the link to verify that you control the address.

However in the past few years, **social logins** from Facebook, Twitter, Google,

LinkedIn, etc. have rapidly gained in popularity because:

- You can “close the loop” faster—the website can send you to the social network and within seconds receive back verification of your account—and you never have to leave your browser.
- The website can request data you already have stored at the social network, such as your contact information, photos, or friends list.
- The website can also ask for permission to share back to your social network, creating a viral effect for new apps.
- The process can be more secure than email.

Initially, each social network had its own proprietary social login API. Now they are moving towards open standards such as [OAuth](http://oauth.net/) and [OpenID Connect](http://openid.net/connect/). However these standards only control the connection protocols. The social networking services you are connecting to still remain proprietary. This leaves users and Web sites with two choices:

- Continue to use slower, kludgier, less useful email address verification because it is a decentralized open standard where users have a choice of providers.
- Use faster, easier, more powerful social logins from proprietary social networks but run the risk of being locked in to those providers.

Personal channels represent a third way: using a **personal login** via the user’s personal cloud. These provide all the benefits of a social login without any of the compromises. And with a relationship network that supports personal cloud portability, users are never locked-in.

Personal logins have several additional advantages:

- **Authentication is inherently contextual.** Each time a user connects to a new site, the user can choose whether the personal channel uses an anonymous, pseudonymous, or verified identity—whatever the user feels is appropriate for the context of the relationship.
- **Any type of data may be shared between the user and the site.** With proprietary social networks, only the data types they support can be shared. With an open standard relationship network, any party may introduce new data types as needed.
- **Trust can be managed under a mutually acceptable trust framework.** Rather than the one-way dictates of a site’s privacy policy or a social network’s terms of service, a personal channel link contract can reference a mutually acceptable trust framework designed to represent the best interests of both parties to a relationship. For a specific example, see the Respect Trust Framework that establishes [five universal principles of “fair trade” for personal data](http://openidentityexchange.org/trust-frameworks/respect-trust-framework/) (http://openidentityexchange.org/trust-frameworks/respect-trust-framework/). See also Chapter 20 of Doc Searls’ book, *The Intention Economy* (http://www.amazon.com/Intention-Economy-When-Customers-Charge/dp/1422158527/).

Benefit #5: Scalable Trust with Contextual Reputation

Trust at scale is a challenge for any network. For example, as discussed in [Benefit #1](#), Internet email is beset by spam today because trust was not in scope when the protocol was developed.

By contrast, the major social networks have done a relatively good job of enforcing trust because they maintain a central point of control and can use arbitrary authority to suspend accounts or remove users from the network. Facebook uses the graph of real relationships to provide valuable feedback about account authenticity.

This is not an option for an open standard relationship network where there is no single point of control. Rather, trust mechanisms must be baked into the DNA of the network using one or more trust frameworks to which the members of the network have agreed.

One such framework is already in use: the Respect Trust Framework, which [won the Privacy Award at the 2011 European Identity Conference](#) (<http://blog.connect.me/connectme-wins-2011-eic-privacy-award>). The key innovation of the Respect Trust Framework is that it treats the entire relationship network, called the Respect Network, as a **peer-to-peer (p2p) reputation network**. Each personal channel represents a connection between two members of the network. For any connection, either party may give the other a positive reputation vote—called a **vouch**—on one or more tags representing contexts where that trust has been earned.

Figure 3 is an example of a reputation card for one of the authors of this paper, showing the vouch count for each of six contextual tags.

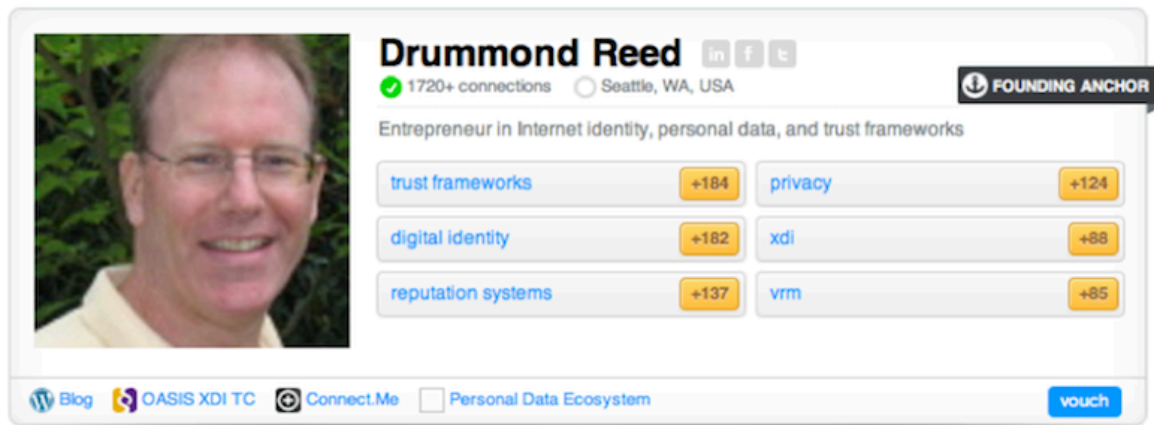


Figure 3: An example of a contextual reputation card using the Respect Trust Framework.

To prevent gaming attacks, especially the **Sybil attack** (http://en.wikipedia.org/wiki/Sybil_attack) where fake accounts are used to manipulate reputations, members must progress through four escalating levels of trust (Unverified, Verified, Trusted, Trust Anchor). The highest level, Trust Anchor, requires personal verification by three other Trust Anchors. Trust Anchors serve as the administrators of any problems or complaints, much as the most experienced Wikipedia users serve as administrators for the Wikipedia community.

Not only can this trust model scale globally, as Wikipedia has, but it can also provide valuable contextual reputation metadata about every user, company, service provider, and application in the network.

Benefit #6: Automated Event Processing

Most sophisticated email users create rules that are triggered each time their inbox receives a message. These rules automate routine tasks, e.g., storing a message in a particular folder, forwarding it to another address, sending out a vacation notice, etc. The more sophisticated email user agents also know how to process specific types of email attachments. For example, they can automatically add a calendar request to your calendar, or a vCard attachment to your address book.

However, short of very customized programming, that's the extent of the automation email client can provide. Again, this is by design: with SMTP email, messages were intended primarily for human, not machine, processing.

By contrast, with the EXP & XDI protocols, messages are designed for *both human and machine processing*. In technical terms, this means every message is an **event**, and with KRL (Kinetic Rules Language) used by EXP, developers can use **event-based programming** to write rich, highly customized apps for processing the messages sent and received over personal channels.

For example, rather than simply post to your calendar, a calendar request delivered over a personal channel could:

- Automatically notify you when the appointment is particularly important or urgent (see [Benefit #8—Intelligent Notification](#)).
- Automatically deal with conflicts depending on the nature of the appointment and your relationship with the sender.
- Understand the particular type of calendar event requested (a lunch, plane flight, conference, birthday party) and automatically carry out related tasks (make a restaurant reservation, set a flight alert, file a conference registration, schedule a birthday gift reminder).
- Automatically link other information relevant to the appointment (client files, menus, itineraries, profiles).
- Automatically invoke a relevant application to help you process the related tasks. For example, a personal channel message about a cancelled flight could automatically invoke a personal trip management app that in turn determines in

real time the rebooking options—reducing the task to as little as a single click to approve the rebooking.

A programmable personal channel network, operating against messages designed to contain semantic data and events, can save people and companies countless hours in routine message processing and workflow tasks.

Benefit #7: Intelligent Filtering and Organization

Email programs are judged by their ability to sort signal from noise so users can concentrate on the most important messages and ignore the rest. One of the reasons this is so difficult with email is because the signals are so weak. The mere fact that someone is in your address book—or that you have replied to her messages—is not a strong indication of how important that person is, let alone of the importance of any particular message from her. What’s more, the only explicit mechanism for indicating message priority in Internet email—the priority flag—was so abused when email went mainstream that it is [not supported by most email providers](http://www.quora.com/iOS/Why-doesnt-iPhone-support-email-priority-flags) (<http://www.quora.com/iOS/Why-doesnt-iPhone-support-email-priority-flags>) .

Despite their emphasis on relationships, social networks provide only marginally better signals. The fact that you are connected to someone on Facebook or LinkedIn does not say anything about their relative importance to you. And following someone on Twitter is only a slightly stronger signal—it still does not indicate *why you are following* someone.

Many “influence metrics” such as Klout, Kred, and PeerIndex focus on retweets or reshares. Reshares are an explicit indication of the relevance of a specific message, and an implicit endorsement that the sender’s content is worth sharing with your own followers. In addition, with tweets, any associated hashtags may also indicate the relative importance of topics to which the message relates.

However, even if all of these signals are wrapped together, they would only be a walkie-talkie in comparison to a personal channel’s 100,000 watt transmitter. Here’s why:

- **Every channel contains metadata describing its messages.** A hashtag in Twitter is a channel in a relationship network. But unlike Twitter, channels may have subchannels to whatever depth is required to meet publishers and subscribers needs for filtering the channel.
- **Every sender has a reputation on every channel.** This is the power of the contextual reputation system described in [Benefit #5—Scalable trust and contextual reputation](#). Reputation can be directly calculated relative to the subscriber, to the subscriber’s own network, or to the entire network. So your personal cloud can automatically and dynamically understand which senders are most important to you in any given context. For example, messages from the coach on your son’s lacrosse team channel will carry more weight than a message from a marginally involved parent.
- **Messages can carry their own explicit indicators of urgency and importance.**

The email priority flag died from a [tragedy of the commons](http://en.wikipedia.org/wiki/Tragedy_of_the_commons) (http://en.wikipedia.org/wiki/Tragedy_of_the_commons) — there was no reputation system to keep senders honest. In a relationship network where every sender has reputation in every channel, senders will be careful to only flag truly urgent or important messages as such or risk having their reputation immediately devalued.

- **Link contracts can define additional rules for channel filtering and organization.** Any message may trigger rules associated with its semantic content. So individuals, groups, and apps can all define and execute rulesets operating against the semantics defined for any particular channel. For example, a sports team app could automatically send requests to every parent on your son's lacrosse team to sign up to bring after-game snacks—and automatically collate their responses.

The result will be a radically new notion of an inbox, where you can instantly sort messages by their urgency and importance relative to any subject, many messages are answered automatically, and all the decisions and actions you are being asked to take are queued up ready for execution.

Benefit #8: Intelligent Notification

Today, if you need to reach someone regarding a matter that is particularly urgent or important—for example being late for a scheduled lunch, you generally have no choice but to use *channel escalation*, i.e.:

- If email is too slow, you try instant messaging.
- If the recipient is offline, you try texting.
- If the urgency is high enough, you escalate to a voice call.
- If it is a full emergency, you use all channels at once.

This scenario illustrates three shortcomings of the existing options:

- All these forms of messaging are **transport-specific**.
- None of them are **context-aware**.
- **Humans do all the work** of figuring out when to notify you about what and where.

Personal channels can solve all three of these problems:

- Messages are **transport-independent**.
- Messages are **context-aware**.
- **Apps do all the work** of figuring out when to notify you about what and where.

In short, your personal cloud acts like a personal secretary managing your notifications. For example, in [Benefit #6—Automated event processing](#), we described receiving a calendar request. Let's assume it is an I'm-going-to-be-late-to-our-lunch message from a friend.

- The calendar request would come in over the personal channel you have with your

lunch partner. Consequently, your personal cloud would know the relative importance of the relationship you have with that person.

- Your personal cloud would know that it is a lunch meeting, involving a particular time window. The closer you are to that time window, the more urgent the message.
- Your personal cloud knows your current location and state (at your desk, driving, in another meeting).
- Your personal cloud knows your communications preference, i.e., that if you are at your desk, you prefer an IM; if you are in a meeting, you prefer a text; but if you're driving, you prefer a voice call.

So, all your lunch partner needs to know is to send an "I'm going to be 15 minutes late" message via your personal channel, and your personal cloud would take it from there, determining whether to send you an IM, text, or voice call.

Benefit #9: Automatic Data Exchange

With the exception of custom-programmed email workflow systems, the only form of data exchange supported by email is attachments. The fact that we are often drowning in them is a testimony to how badly we need a universal form of data sharing.

The need for better ways to share has driven adoption of social networks, as well as specialized social sharing services for photos, files, resumes, calendars, trips, parties, and so on. Each provides an easier, faster, more structured way of sharing specific types of data with specific people and groups.

But whereas email is a ubiquitous open standard, so far all of the social networks and social sharing services are proprietary systems with their own special terms of service, privacy policies, and data schemas. They are not open standards for data exchange that can be used by anyone, anywhere, for any purpose. And while most have some kind of API, APIs don't mitigate these fundamental limitations.

Better data exchange is a primary driver for adoption of an open standard relationship network. Personal channels are as adept at sending structured data between machines as they are at sending smart messages between people. In particular:

- Link contracts, used in conjunction with shared semantic data dictionaries, can specify the precise types and instances of data that a particular channel may read or write.
- With bi-directional channels, data may flow in either direction, in or out of a personal cloud.
- Link contracts can also specify the precise semantics of synchronization, i.e., whether updates should be pushed or pulled, and at what level of security. This enables a relationship network to automate very wide-area synchronization, such as a global change-of-address for an individual or a company.

- Personal channels can redefine the very idea of “forms”, i.e., a form becomes a semantic data query to which a personal cloud can prepare a fully automatic or semi-automatic (user-reviewed) response.
- Every user of the network may extend the data types that may be shared, and these new data types may be discovered and subscribed to via **dictionary channels**—shared connections to semantic data dictionaries.

Benefit #10: Rich Sharing

There is much more to data sharing than the mechanical function of transferring data between machines. Sharing is a social and business function that is inherently more complex than centralized, single-purpose applications can handle. No wonder so many different social sharing solutions have emerged:

- Facebook, Twitter, Google+, etc. for sharing snapshots, status updates, links, and discussions.
- LinkedIn, Xing, BranchOut, etc. for sharing resumes.
- Dropbox, Google Drive, Skydrive, etc. for sharing files.
- Flickr, Photobucket, Picasa, Shutterfly, etc. for sharing photos.
- TripIt, Dopplr, Yahoo Travel Planner, etc. for sharing trips.
- EventBrite, Evite, Punchbowl, etc. for sharing parties and events.
- Quora, StackOverflow, WikiAnswers, Yahoo Answers, etc. for sharing Q&A.

And countless more. The sheer number of these—and the extraordinary market value some have achieved—suggest the power of solution that combines the ubiquity of email with the power of social sharing. This combination is called **rich sharing**, and it deserves a more detailed examination.

The Requirements of Rich Sharing

Marc Stiegler of HP Labs has [written](http://www.hpl.hp.com/techreports/2009/HPL-2009-169.pdf) (http://www.hpl.hp.com/techreports/2009/HPL-2009-169.pdf) (PDF) and [spoken](http://www.youtube.com/watch?v=T92ZboMsH1w) (http://www.youtube.com/watch?v=T92ZboMsH1w) about rich sharing, and Alan Karp has [written about PubShare](http://www.hpl.hp.com/personal/Alan_Karp/pubshare.html) (http://www.hpl.hp.com/personal/Alan_Karp/pubshare.html) , a system Marc built that demonstrates rich sharing. Alan relates two stories that contrast our expectations about sharing in the physical and online worlds. The first takes place in the physical world:

In an emergency, Marc asked me to park his car in my garage. I couldn't do it, so I asked my neighbor to do it for me and said to get the garage key from my son.

The second involves an online file sharing scenario:

In an emergency, Marc asked me to copy a file from his computer to mine. I couldn't do it, so I asked my neighbor to do it for me and said to

get access to my computer account from my son.

The second story seems ludicrous because we can't see a reasonable way for it to work even though it closely resembles the scenario from the physical world.

Rich sharing characterizes the set of features that make human communication work in the physical world. Using this model, we can determine how to create better online communication systems. Communication systems, like email, that embody rich sharing feel natural to users and thus succeed. Systems that lack rich sharing's features feel stilted or unwieldy and thus don't scale the way their designers intended.

Sharing is easy and technically uninteresting in situations where the shared item is public and there's no need to authorize access to it. Similarly **workgroup-style sharing** is relatively straightforward and the tools for protecting resources in workgroups such as role-based authorization control (RBAC) and access control lists (ACLs) are well understood. For purposes of contrast, let's call unprotected and workgroup-style sharing **simple sharing**.

Sharing becomes much more nuanced when access to the shared item must be restricted and the players in the sharing scenario operate in independent security domains. Many real-world scenarios require rich sharing. Stiegler and Karp demonstrate why workgroup-style sharing can't accommodate rich sharing scenarios in the papers mentioned above. We refer interested readers directly to those paper for more detailed descriptions.

Rich sharing is characterized by six key attributes:

- **Dynamic**—Sharing can be done without reconfiguring the system or having other work done by the sharer's IT department.
- **Attenuated**—Sharing happens with the right permissions on the right items.
- **Chained**—A shared item can be re-shared in appropriate ways. Authority can be re-delegated. Building attenuated chains of delegated authority is difficult in simple sharing architecture
- **Cross domain**—Sharing can occur across security domains without the user linking the domains in an ad hoc manner or the IT department having to setup special purpose federated identity systems.
- **Recomposable**—The shared item or service can be used in conjunction with other resources and services—even if those documents and services exist in a separate security domain.
- **Accountable**—Even though sharing can be re-delegated along a chain, the original owner must maintain the ability to audit and track the use of the shared item and hold the appropriate parties accountable for misuse.

Stiegler and Karp make a case that email succeeds because email demonstrates these six attributes. In contrast, we can easily find examples in other sharing architectures that fail to incorporate one or more of these features and thus become difficult to use as the sharing scenarios get more complicated. Today's

popular social networks and sharing services all fail to support one or more of the above attributes.

How Personal Channels Enable Rich Sharing

Earlier we explained that channels provide a metaprotocol for interaction. Thus they represent *a way of doing things rather than a place*. Rich sharing is more easily supported by ways—protocols—rather than by places. In fact, certain properties of rich sharing, such as being cross domain and recomposable, are nearly impossible to achieve using a place-based approach such as a website.

By contrast, let's see how personal channels as we have explained them in this paper stack up on the attributes of rich sharing:

- **Dynamic**—Personal clouds can form new channels and send or receive messages over them at any time, just like email.
- **Attenuated**—Link contracts provide a means of fine-grained access control that enables attenuation.
- **Chained**—Upon receiving a message on a channel, a personal cloud can forward or delegate that message to other personal clouds over other personal channels. This delegation may be algorithmic, but is always under the ultimate control of the channel's owner.
- **Cross domain**—Each personal cloud functions as its own independent domain in the same sense that each email inbox represents its own independent domain. Thus a personal channel can carry messages and data from one domain to another as easily as email, but with all the advantages of structured data and event processing.
- **Recomposable**—Messages sent along a channel, be they events, queries, or data can be composed with other information from other sources (e.g. APIs, other channels, etc.) as part of the processing done by a personal cloud.
- **Accountable**—Personal channels are uniquely identified and individually revocable. This combined with signed link contracts and reputation systems such as the Respect Trust Framework provides a flexible framework for accountability and trust that can be tuned to the requirements of any sharing community.

By meeting all six requirements, personal channels on an open standard relationship network can not only provide the full benefits of rich sharing, but do so at the same scale as the Internet email network or the largest proprietary social networks.

Conclusion

Computer systems often begin as centralized systems because such architectures are easier to build and understand. After a time, they evolve to decentralized systems as the limitations of the centralized solution become apparent. Figure 4 shows how architecture (centralized vs decentralized) and channel type (single channel vs. multi-channel) combine to create different systems. As we've

discussed, email, a single-channel communications model, moved from a centralized to a decentralized architecture via a protocol (SMTP), creating a much richer communications platform. Similarly, we believe that multi-channel relationship systems will undergo a similar transformation as we determine how best to support protocols that provide rich sharing environments.

	single channel	multiple channel
decentralized	Email (2000)	Relationship Network (2016?)
centralized	AOL (1990)	Facebook (2012)

Figure 4: The evolution of communications.

The benefits of a decentralized relationship network built from personal clouds connected via personal channels are clear. Table 1 summarizes how email, social networks, and personal channels compare across the ten types of benefits described in this paper.

Benefit	Email	Social Networks	Personal Channels
#1 - Access and Volume Control	No	Access control only	Yes
#2 - Lifetime Portability	With own domain	No	Yes
#3 - Secure Delivery	Ad hoc	No	Yes
#4 - Simple, Standardized, Contextual Authentication	No	Not standard Not contextual	Yes
#5 - Scalable Trust with Contextual Reputation	No	Not contextual	Yes
#6- Automated Event Processing	No	No	Yes

#7 - Intelligent Filtering and Organization	Ad hoc	No	Yes
#8 - Intelligent Notification	No	No	Yes
#9 - Automatic Data Exchange	No	Specific data sets only	Yes
#10 - Rich Sharing	Yes, but labor intensive	No	Yes

Table 1: Comparison of email, social networks, and personal channels

In this paper we've argued that personal channels are the proper architecture for creating distributed relationship networks built from personal clouds. Because personal channels provide a flexible, open way of sharing they can support the richness of human interaction in ways that a centralized Web site, regardless of its sophistication, never will. By combining the best features of both email and social networks, addressing key deficiencies in Internet security and privacy, and adding a powerful new layer of communications convenience and automation, personal channels are poised to be the next major step forward in both digital messaging and online relationship management.

Finding Out More

You can discover more information about the concepts and technologies in this series from a variety of sources including [Project VRM](http://blogs.law.harvard.edu/vrm/) (<http://blogs.law.harvard.edu/vrm/>) , [Respect Network](http://respectnetwork.com/) (<http://respectnetwork.com/>) , [Kynetx](http://www.kynetx.com) (<http://www.kynetx.com>) , and the blogs of the series authors (listed in the biographies below). We also point you at Doc Searls' book [The Intention Economy](http://www.amazon.com/The-Intention-Economy-Customers-Charge/dp/1422158527) (<http://www.amazon.com/The-Intention-Economy-Customers-Charge/dp/1422158527>) and Phil Windley's book [The Live Web](http://www.amazon.com/exec/obidos/ASIN/1133686680/windleyofente-20) (<http://www.amazon.com/exec/obidos/ASIN/1133686680/windleyofente-20>) .

If you're interested in creating personal clouds, the [Kinetic Rules Engine](https://github.com/kre/Kinetic-Rules-Engine/) is open source (<https://github.com/kre/Kinetic-Rules-Engine/>) . However, the easiest way to get started is using the online service provided by Kynetx. You can try out personal clouds and the KRL programming model for free by [creating an account at Kynetx](http://www.kynetx.com) (<http://www.kynetx.com>) . Kynetx accounts are free and you can develop multiple applications and run them without charge for non-commercial use. [Examples and documentation](http://developers.kynetx.com) (<http://developers.kynetx.com>) are available online.

Lexicon

The following terms are used in this paper and in our vision of the Live Web:

- **event**—a notification or message containing data about a specific state change. Events indicate that *something happened*.
- **event-based programming**—a style of programming where routines respond to event notifications. In contrast to request-response style programming, event-based programming leads to looser coupling and less semantic entanglement.
- **link**—a connection, represented by a personal channel, between two personal clouds, whether one-way (asymmetric) or two-way (symmetric). A link is a subscriber relationship.
- **link contract**—a data structure that specifies the terms and conditions to which two or more parties agree for a specific personal channel. Link contracts specify the nature and behavior of a channel.
- **personal channel**—a communication link between a personal cloud and other personal clouds or network services.
- **personal cloud**—a cloud-based virtual computer that combines event processing and personal data. Personal clouds serve as the online representative of an entity, often a person. Personal clouds have an operating system that runs applications, processes events, and manages data on behalf of and under the direct control of their owners.
- **relationship network**—the network of people, organizations, and things linked through personal channels.
- **rich sharing**—the set of features that characterizes the flexible nature of human communication.

Rights

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](http://creativecommons.org/licenses/by-nc-sa/3.0/) (<http://creativecommons.org/licenses/by-nc-sa/3.0/>) .

Live Web Series Authors

Drummond Reed is co-founder and Chairman of Respect Network Corporation (RNC) and Managing Director of the Respect Network. He is co-author with Scott David, Joe Johnston, and Marc Coluccio of the Respect Trust Framework upon which RNC's [Connect.Me](http://connect.me) (<http://connect.me>) reputation network is based. Drummond has also served as co-chair of the OASIS XDI Technical Committee since 2004.

Prior to RNC, Drummond was Executive Director of two industry foundations: the Information Card Foundation and the Open Identity Exchange. He has also served as a founding board member of the OpenID Foundation, ISTPA, XDI.org, and

Identity Commons. In 2002 he was a recipient of the Digital Identity Pioneer Award from Digital ID World. Drummond blogs on digital identity, personal data, personal clouds, and trust frameworks at <http://equalsdrummond.name> (<http://equalsdrummond.name>).

Phillip J. Windley is the Founder and Chief Technology Officer of Kynetx. Kynetx is a personal cloud vendor, providing the underlying technology for creating, programming, and using personal event networks using KRL and semantic data interchange via XDI. He is also an Adjunct Professor of Computer Science at Brigham Young University where he teaches courses on reputation, digital identity, large-scale system design, and programming languages. Phil writes the popular [Technometria blog](http://www.windley.com) (<http://www.windley.com>) and is a frequent contributor to various technical publications. He is also the author of the books [The Live Web](http://www.amazon.com/exec/obidos/ASIN/1133686680/windleyofente-20) (<http://www.amazon.com/exec/obidos/ASIN/1133686680/windleyofente-20>) published by Course Technology in 2011 and [Digital Identity](http://www.amazon.com/dp/0596008783/windleyofente-20) (<http://www.amazon.com/dp/0596008783/windleyofente-20>) published by O'Reilly Media in 2005.

Prior to joining BYU, Phil spent two years as the Chief Information Officer (CIO) for the State of Utah, serving on Governor Mike Leavitt's Cabinet and as a member of his Senior Staff. Before entering public service, Phil was Vice President for Product Development and Operations at Excite@Home. He was the Founder and Chief Technology Officer (CTO) of iMALL, Inc. an early creator of electronic commerce tools. Phil serves on the Boards of Directors and Advisory Boards for several high-tech companies. Phil received his Ph.D. in Computer Science from Univ. of California, Davis in 1990.