

The Live Web Series

SquareTag: Making Everything “Smart”



Phillip J. Windley, Ph.D.,
Founder & Chief Technology Officer, *Kynetx, Inc.*

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 third paper in that series. We recommend reading other papers in the [Live Web series](http://www.windley.com/liveweb/) (<http://www.windley.com/liveweb/>) as background for this paper.

Executive Summary

Smart devices promise to change how we interact with every thing around us. For years, we’ve lived with a vision of a future world where every thing is connected and online. Devices like the iPhone have given us a glimpse of the vast possibilities of always being online. Pundits tout the advantages of the Internet of Things, when everything is connected.

But putting computers in everything and connecting them will take time, at best, and may be prohibitively expensive in many situations. No matter how cheap computers get or how ubiquitous connectivity becomes, there will always be something that doesn’t justify the expense. Consequently, we’ll *always be waiting* for something to become smart.

The answer to this dilemma is to use virtual computers—*personal clouds*—to serve as proxies for many of the things we want to be smart. Personal clouds are cloud-based virtual computers that are so inexpensive to deploy that we can create an unlimited number of them. And yet, *personal clouds provide powerful, sophisticated programming and data models*, making them more than a match for the task of making everything smart.

By using personal clouds to make things smart, we can connect every thing now—and skip the waiting. Making things smart with personal clouds is cheap. So we can connect anything, right now—no matter how insignificant. We can even connect things manufactured years ago. And as devices get smarter, their sensors

and on-board computers will work with personal clouds to provide even greater utility. Personal clouds are a way to bridge both smart and not-so-smart devices and bring everything onto the Internet of Things today.

SquareTag uses personal clouds to put everything online, making them active participants in the Internet of Things This white paper shows why making every thing smart is a good idea, how personal clouds make that possible, and presents SquareTag as way to achieve this goal.

Introduction

Smart devices are personal, mobile, connected, and have an embedded computer. Mobile phones, some watches, and even a Fitbit™ are all examples of things we'd think of as "smart devices." Future-looking pundits have long given us visions of a world filled with connected things where Moore's law has rendered computing and connectivity so inexpensive that literally everything is *smart*. We wait for the day when our bicycles, tables, and even the corner bus stop are *smart*. But, what if we didn't have to wait?

This paper introduces a method for adding connectivity and computational capability to almost anything and motivate it with an example system, called *SquareTag*, that gives bicycles, cars, appliances, and anything else their own space online.

Connecting Every Thing

We're used to thinking of laptops, phones, and even special purposed devices like the Fitbit being *connected*. We imagine future worlds where swarms of devices connect to each other and do our bidding. We already see the start of that as connected things become increasingly common.

What's less apparent, however, is how *everything* becomes connected. For example, most people would scoff at the idea of the coffee table in the lounge at the hotel bar or the corner bus stop being connected in a meaningful way. We're a long way from embedding computers in small items of furniture and outdoor metal structures. Nevertheless, such connectivity isn't just possible, it's available now and has startling possibilities to make the world a better place to live and an easier place to navigate.

The first step is to imagine everything having an independent identity—a name—that it keeps throughout its existence. Through its name, the object links the activities of people, organizations, and even other things.

Making every thing smart isn't science fiction, it's doable now and has startling possibilities to make the world a better place to live.

The second step is to associate the name with a computer. The computer needn't be physical and it needn't be in the object. But by giving everything the ability to store data and run programs, we make them smart and give them the power to connect.

Connected objects create entirely new ways of interacting with the world. In [The Coming Zombie Apocalypse](http://designmind.frogdesign.com/blog/the-coming-zombie-apocalypse-small-cheap-devices-will-disrupt-our-old-school-ux-assumptions.htm) (<http://designmind.frogdesign.com/blog/the-coming-zombie-apocalypse-small-cheap-devices-will-disrupt-our-old-school-ux-assumptions.htm>) , user experience expert Scott Jenson describes a world of devices that are able to join forces to solve problems for people on-the-fly:

In a sea of these cheap devices, we'll likely need a display on our phones/tablets that lists (and most likely ranks) nearby devices that might be of interest. Selecting any one would allow that device to interact in any way it chooses. In this manner, the classic concept of "an app" will just be available on demand for any device or object a person happens to be in front of. The idea of a user downloading, managing, and launching apps will feel just plain silly.

Some of the cluster devices won't even need computation to be "smart." Just geotagging every bus stop in a city would allow people to walk up to any one and interact with it by looking up its exact location through a cloud service, creating, in effect, "websites on demand". The great benefit of these "dumb points" would be that since they're nearly free, experiences can be deployed literally anywhere and at great scale.

But it's not just about quick access; these dumb points explode the classic concept of an app by focusing on the precise node I'm current in front of. I don't need a 'city bus app', I just need the app for this particular bus stop, which shows me, without any interaction, the next 3 buses. The same is true for classic store apps. Instead of firing up the GPS and figuring out which Starbucks I'm in, the on demand page shows, without effort, the one I'm currently standing in, complete with today's special right there at the top. This approach works not only for my current location but now adds significant depth to any mapping application: I can now 'peek into' stores when I zoom into a street, seeing mini pages for each location right in context of the map itself.

Living in a sea of smart devices and "dumb points", as Jenson calls them, changes people's interaction patterns and user experience in several important ways:

- As Jenson notes, the idea of an *app* becomes "ridiculously quaint". Things provide their own interface, creating the experience on demand and in context. This independence from apps is important because as nice as current smartphones are, the notion of apps for everything doesn't scale to a world where literally every thing is connected.
- Over the past five years, more and more of our world experience has moved onto a small screen. People are seen hunched over their phones, peering at them,

absorbed in the user experience on their phone. A world of connected things reverses this trend, making the things themselves worthy of interaction and opening up our experience to the larger world.

The coming sections of this paper will show how we can build a connected world of “dumb points” and smart devices now, using readily available technology.

A Smart Bike

Imagine a bicycle for a minute. Bikes aren't smart in the same sense as a smartphone. I don't know of any bikes that have embedded computers. But if a smart bike existed, what could a smart bike do?

Now, imagine the bike being connected to its manufacturer, the bike store that sold it, and its owner. These are some of the *stakeholders* of that particular bike. From its earliest point in being, the bike would be able to keep track of data about itself, things like its specifications, when it was made, and even the provenance of the components used in its manufacture.



The bike would keep track of inventory data such as when it was delivered to the bike shop, who assembled it, it's current configuration, its price, and when it was bought and by who. If you change out the wheelset when you buy it, the specs for the new wheels will be associated with the bike, rather than the originals. Want to understand the gearing ratios and what it would mean to change out the freewheel? The bike can tell you.

The bike has it's own interface, creating an experience on-the-fly according to context. The bike would manage its own maintenance, keeping track of what's been done and what needs to be done. The bike alerts its owner when something needs done—*the bike is its own TODO list*. A smart bike knows about the recall status of all its parts and can request maintenance based on usage and part replacement history. Since the bike is keeping track of it's own data, any maintenance records written by the bike shop end up on the bike. If you go to a different shop, they see the maintenance history regardless of where any previous maintenance was performed.

And if a smart bike gets stolen, a “find my bike” service reunites bike and owner.

The owner can tell the bike to schedule an appointment with the bike shop or even put together a riding group.

The bike's also linked to a GPS unit so that it knows where it's been. Of course the GPS can also be linked to the owner's personal cloud. Apps on the bike tie into the owner's favorite fitness sites or show the gas that's been saved.

A smart bike will alert its owner when it needs a new set of tires or when the tires she normally buys drop to a new low price on Amazon.

Suppose the owner would like to upgrade some of the bike's components. She might not have the time to look all over the Net, but she doesn't have to. The smart bike will do it for her. She creates the wish list, set some prices on the various components and lets the smart bike do the shopping for her over the next six months. When a component with the right price is found, the owner is merely asked for permission to confirm the purchase. The new components work with the bike because the bike knows what will fit. Just like Amazon won't let you buy the same Kindle book twice, the bike won't let you buy parts that don't work with it.

If the owner decides to sell the bike, the bike can manage its own listing. When the bike is sold, the bike keeps its smarts. The new owner can see the maintenance logs and anything else specific to the bike. The previous owner's personal information, however is not intermingled with the bike's and thus there's no privacy concern.

And, of course, it's not just bikes, it's everything. Literally anything at all could be *smart*. The secret to seeing how a smart bike is possible is to realize that the computer doesn't have to be in the bike. Rather the computer can be in the cloud. Any bike, even one you bought years ago can be made smart by giving the bike a personal cloud. Personal clouds, cloud-based virtual machines, can be created that are cheap and facile enough to literally make *every thing* smart.

The secret to seeing how a smart bike is possible is to realize that the computer doesn't have to be in the bike. Rather the computer can be in the cloud.

By making the bike smart—by giving it a virtual, online existence—we have enabled the bike to form relationships with a vastly expanded set of stakeholders from the manufacturer, to the shop that sells and services it, to its owner, and even to other things like the GPS unit. In *Shaping Things* (<http://www.amazon.com/dp/0262693267/>), Bruce Sterling says (ppg 105–106):

Only a limited number of people can interact with any particular physical object. A real, physical thing is too small, too parochial, too limited to remain the center of importance for a large number of people. A real, physical object occupies too small a piece of space and time. Most people in the world will never be able to see it or touch it. Its ability to interact with people is sharply limited. So only a limited number of people can contribute their skills and their insights to the process of that object's development. ... The object's virtual representations, however, can have stakeholders.

By increasing the footprint of physical objects, giving them a virtual presence, we make them more interesting to a wider group of people and organizations. Because the bike is merely a physical manifestation of a much larger and capable virtual

thing, we've made it more valuable. The virtual representation can attract a tremendous amount of attention. For example, the smart bike can be continuously updated with new apps as the owner's needs changes. None of the functionality or even the data schema for the bike is fixed. In fact, the smart bike is a *platform* upon which developers can write apps.

And if there's value in a smart bike, just think of the possibilities when your cars, appliances, everything else are smart too.

Introducing SquareTag

Kynetx has developed a system, called *SquareTag*, that makes things smart—just like the smart bike in the last section. To get a feel for SquareTag in action, imagine the following scenario.

Police find an abandoned bicycle. They notice that there is a small aluminum tag on the bike with a [QR Code](http://en.wikipedia.org/wiki/QR_code) (http://en.wikipedia.org/wiki/QR_code) stuck on the frame. They take out their smartphone, scan the tag and see a screen that says “This bicycle was reported missing; please contact the owner.” along with a red button labeled “Push to Contact”. The information might also include reward information if the owner desires.



When the policeman pushes the red button on his screen, he's given a form that he can fill out telling the owner how to claim the bike. The message is delivered to the bicycle's owner in a secure manner, using a channel of the owner's choosing. The owner receives the message and happily recovers her bicycle.

This successful reunion of the bicycle with its owner came about because the owner gave the bicycle an identity by affixing the tag to the bicycle and claiming it by scanning the tag. The process allowed the owner to record relevant information about the bike and link it to a communication channel. One of the key features of smart things is they keep track of themselves. Just like the lost bike who knows who its owner is, tools and household appliances could know where they were and find themselves. SquareTag promises to make the physical world as searchable as the online one.

In addition to helping with recovery if the bicycle is stolen or lost, the SquareTag could also provide functionality like we discussed in the previous section. The bicycle has become a programmable platform with the ability to run applications, store any relevant data, and communicate with it's stakeholders and other network services.

SquareTag links things to personal clouds so they can be smart.

Privacy and security are important aspects of SquareTag. The bicycle and its owner weren't reunited because there was a big database of bike registrations

maintained by the police. Rather, the bike carried its own data and knew who owned it. The data is more private since it is kept in a space controlled by the owner. The data is more secure because it's never released except when the owner decides.

Of course, a SquareTag could be affixed to almost anything and its functionality would change depending on what it was affixed to and how its owner configured it. Unlike a QR code that goes to a Web site with static functionality, SquareTags are associated with a general-purpose, cloud-based virtual computer and the functionality they display is determined by what apps are running behind the SquareTag. Changing or adding to those apps adds new functionality to the SquareTag.

How SquareTag Works

The heart of SquareTag is an identifier that links the tag to the a individual cloud-based computer. Once a thing, like your bike, has an identity, untold functionality can be layered on. Of course, we've had bar codes for a long time. But bar codes identify the class or type of the thing, not the thing itself. Knowing the class of something certainly gives us useful information about it since things of a type have many attributes in common. But the class can never tell us everything we need to know about the instance—the thing itself.



For example, going back to our bicycle, a bar code might tell us that the bike is a [Specialized Roubaix](http://www.specialized.com/us/en/bikes/road/roubaix) (http://www.specialized.com/us/en/bikes/road/roubaix) , but it can't tell us the owner, or that it had its drive train replaced last spring, or that it was at the top of South Fork canyon on June 16 at 4:30pm. To do that, the bike needs an identity that is independent of every other bike. What's more this unique identity must be tied to a means of computation—a *personal cloud*.

A funny thing happens as the bicycle becomes increasingly linked to features through its identity. The bicycle becomes a physical instantiation of its identity —instead of the other way around. The virtual bicycle is accessible to more people than the physical bicycle ever can be. The manufacturer can issue recalls or modify component lifetimes. The bicycle shop can schedule maintenance with the bike. Developers can build applications that expand the functionality of the bike year over year.

Identity enables SquareTag to link physical objects to personal clouds.

While we've initially chosen to use QR codes as the means of not only identifying objects, but also of linking them to their personal clouds, SquareTag isn't about QR codes. They're simply the most convenient way to affix an identity to most things right now. As more phones have NFC capabilities, they will likely be the preferred

method for discovering a thing's identity. Some applications might be better suited to [RFID](http://en.wikipedia.org/wiki/Radio-frequency_identification) (http://en.wikipedia.org/wiki/Radio-frequency_identification) . And, of course, smart devices with embedded computers have their own identifiers that can be used to link them to a personal cloud.

Personal Clouds

In a recent white paper [From Personal Computers to Personal Clouds: The Advent of the Cloud OS](http://www.windley.com/cloudos/) (<http://www.windley.com/cloudos/>) , I describe a system for building *personal clouds* that are capable of interacting as peers with other network services. Personal clouds are general purpose computers that can run multiple applications, access data, and communicate on behalf of their owner. Don't be confused by the term *personal* . While these cloud-based virtual computers *are for people* they are also for *any other entity that needs a cloud-based, virtual computer*.

Personal clouds have three important properties that make them especially applicable to the problem of connecting every thing:

1. Personal clouds are enabled by identity. The identity links the cloud and associated computation, data, relationships and communications to a specific entity.
2. What's more, these cloud-based virtual computers are extremely cheap to spin up and run. They are lightweight and can come into existence to service a single request and then be suspended, awaiting the next. Consequently, we can afford to create as many of them as needed. They're just identifiers and a little bit of storage.
3. Personal clouds provide a programming model that enables developers to build new applications for anything that has one, making personal clouds flexible and customizable. You can add software-based features to your bike that are different from mine. Everything becomes a programming platform.

Personal clouds are so cheap to create and run, we can afford to have one for every thing.

These features make personal clouds, as we've described them in the [personal cloud white paper](http://www.windley.com/cloudos/) (<http://www.windley.com/cloudos/>) , perfectly suited to a world where everything is connected.

Events

Personal clouds are event-based. An event is an indication or notification that *something happened*—a state change occurred. Events often mean that an application should act. Events provide a powerful [metaprotocol for defining the various interaction scenarios](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) that will be present in a personal cloud.

Events augment the traditional request-response programming model of the Web with one that drives action independent of the user. This is a critical component of any system that promises to upset the power structure of the client-server model. David Siegel eloquently explains why in [Apple and the Cloud: A Cautionary Tale](http://www.xconomy.com/san-francisco/2012/04/06/apple-cloud/?single_page=true) (http://www.xconomy.com/san-francisco/2012/04/06/apple-cloud/?single_page=true) :

Adaptability is event driven. It's very different from the demand-driven systems we have today. If something happens in front of you, whether you're on a bike path, driving down the freeway, or flying at 30,000 feet, the system (all participants and their equipment) adjusts. When you take a pill, don't take a pill, hit a golf ball, reschedule an appointment, get in your car, or walk near a store that has something on your shopping list, the event triggers a response and keeps other people up to date automatically. In an event-driven world, we don't know which apps we need, and it won't matter. A piece of code sitting in the cloud that is perhaps almost never used is nevertheless ready to respond to something unusual, and we may only learn about this software service after we needed it. An event-driven world is designed to change as the data changes.

An event-driven world is designed to change as the data changes.

As David points out, event-driven architectures are adaptable in ways that demand-driven architectures aren't. Events enable late binding of functionality leading to extremely dynamic and responsive systems.

Personal Channels

A personal cloud listens for events on *personal channels*. A personal cloud can have an unlimited number of channels, allowing a channel to be assigned individually to any one or any thing with whom the personal cloud interacts. This gives the owner of the personal cloud complete control of inbound events. If a contact isn't behaving as the owner would like, the personal channel can be shut down without affecting any of the other relationships that the personal cloud has. Personal channels represent the relationships a personal cloud has with others.

In our white paper [The Personal Channel](http://www.windley.com/pchan/) (<http://www.windley.com/pchan/>) , Drummond Reed and I describe the function and benefits of personal channels. Among others, channels exhibit the following important properties:

- Personal channels provide separately revocable, separately trackable authority to share between personal clouds.
- Channels use a combination of the [Event eXchange Protocol](#)

Personal channels create a relationship network linking smart objects to their owners, manufacturers, and each other.

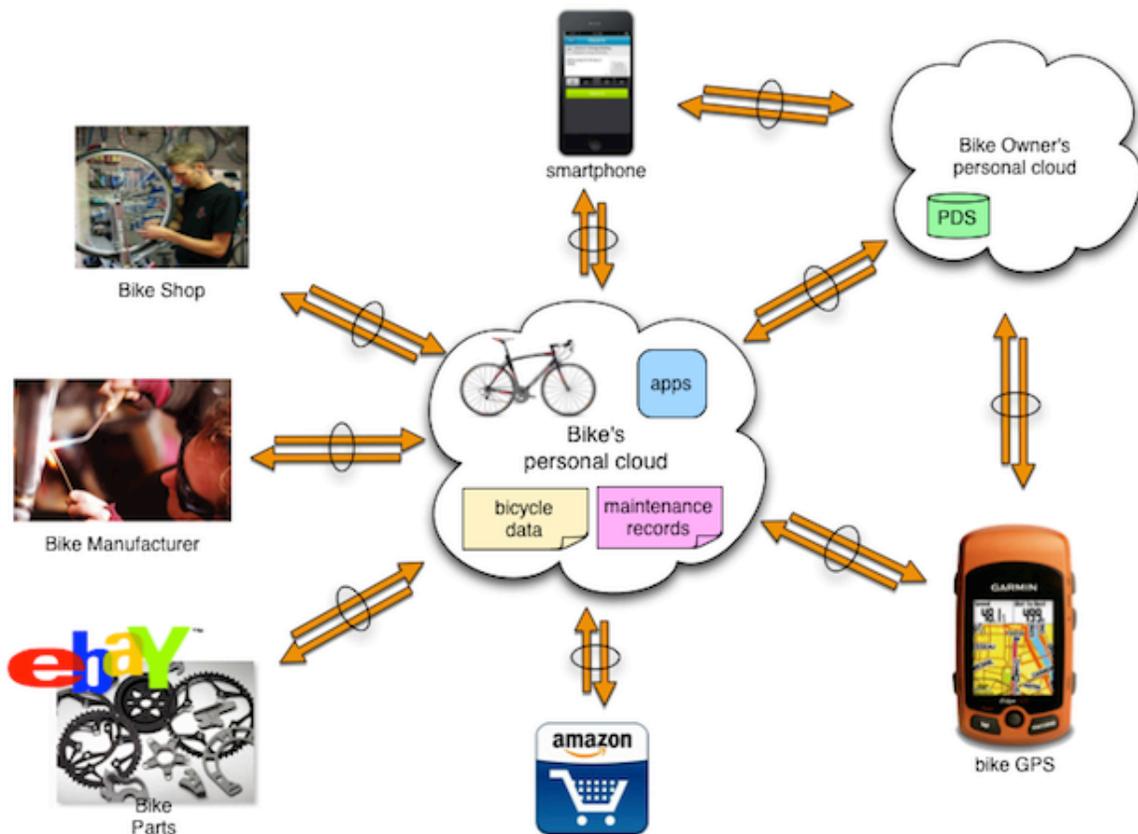
(<http://developer.kynetx.com/display/docs/Event+Exchange+Protocol>) (EXP) and [XRI Data Interchange](http://developer.kynetx.com/display/docs/Event+Exchange+Protocol) (<http://developer.kynetx.com/display/docs/Event+Exchange+Protocol>) (XDI) protocol giving them metaprotocol capabilities. Channels are ways of doing something instead of a place for doing something.

- 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.

Like email, channels form a point-to-point network between personal clouds all speaking the same protocol. Unlike an email server, whose sole function is usually email processing, a personal cloud is more like a general-purpose computer; it has an operating system that runs applications, processes events, and manages data under direct control of its owner.

Personal channels exhibit a property called [rich sharing](http://www.windley.com/archives/2012/05/rich_sharing_and_personal_channels.shtml) (http://www.windley.com/archives/2012/05/rich_sharing_and_personal_channels.shtml). Rich sharing characterizes functional human communication. Because channels support rich sharing, they are extremely flexible and can be used for many purposes. Personal channels provide a messaging system for personal clouds that provides access-controlled, filtered, trustworthy notifications, data exchange, and sharing.

The following diagram shows the channel relationships that a personal cloud might have in the bike scenario that opened this paper.



Personal channels raise events into the bike's personal cloud

Note that the bike's owner also has a personal cloud and the bike and its owner are

linked using a personal channel. In addition, the bike has a relationship with its manufacturer, the bike shop that Alice favors, a GPS unit, and others. The owner of the bike is in control of the bike's personal cloud and can add or delete channel relationships at will.

For example, if the manufacturer abuses the relationship with spam, the owner can delete that channel and sever the relationship without jeopardizing any of the other relationships that the bike's cloud has. When the [bike is sold, control of the bike's cloud is transferred to the new owner](#)

(http://www.windley.com/archives/2012/06/buying_a_motorcycle_a_vrm_scenario_using_personal_clouds.shtml) and a new channel is opened between the new owner's cloud and the bike's cloud.

Applications installed in the bike's cloud can respond to messages and events on the channels in appropriate ways. The owner can install new applications to customize the operation of the bike's personal cloud.

A Real Internet of Things

Through SquareTag, we can connect any thing, any where to the Internet, making it smart. And we can do it at great scale. Literally everything can be connected, not just to the Internet, but to each other. By giving things—dumb or smart—an identity and a personal cloud, we unlock vast possibilities.

When someone places a tag on something and scans it for the first time, they have no idea that a vast edifice of personal clouds and channels is springing into existence. Quite the contrary. The end-user experience is rather mundane—a few Web forms or an app on a smart phone. Nevertheless, the thing has suddenly joined the [Internet of Things](#) (http://en.wikipedia.org/wiki/Internet_of_Things) and linked itself to its owner and anything else that matters. The thing has a virtual existence apart from its physical manifestation. We can write apps for it and change its behavior. We can ask it questions and expect it to answer.

When we give everything a virtual existence by means of an identity and a personal cloud, the entire world becomes our user interface.

When you give everything a virtual existence by means of an identity and a personal cloud, the entire world becomes your user interface. You can ask your screwdriver where it is. The shirt you just bought tells the washing machine the right temperature and detergent. Your motorcycle schedules a needed tune-up before the trip you just added to Google calendar. Your electric car can coordinate charging with your air conditioner to reduce the overall load.

While we could solve any one of these problems with a point solution, the vision of SquareTag is bigger than any single interaction. SquareTag aims for ubiquitous

connectivity of anything at all. A real Internet of Things requires an any-to-any solution, not a one-to-one solution. Personal clouds and personal channels provide the means to create any-to-any interactions so that *every thing* can be active, smart, and participating in your life.

Finding Out More

While SquareTag is a service from Kynetx, based on personal clouds and personal channels, the underlying technology is open and anyone can build services or systems that interact with things made smart through SquareTag.

You can discover more information about the concepts and technologies we've discussed from a variety of sources including the [From Personal Computers to Personal Clouds](http://www.windley.com/cloudos) (<http://www.windley.com/cloudos>) white paper, the [Personal Channels](http://www.windley.com/pchan/) (<http://www.windley.com/pchan/>) white paper, 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 is open source](https://github.com/kre/Kinetic-Rules-Engine/) (<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.

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/>) .

Credits

The picture of the girl on the bike is from [Ayanami_No03](http://www.flickr.com/photos/ayanami_no03/6037203114/) (http://www.flickr.com/photos/ayanami_no03/6037203114/) on Flickr and used in accordance with the Creative Commons license.

The picture of the abandoned bike is from [krakow.bicycles](http://www.flickr.com/photos/krakowbicycles/3406367897/) (<http://www.flickr.com/photos/krakowbicycles/3406367897/>) on Flickr and used in accordance with the Creative Commons license.