

An Interview With Kevin Marks, Technology Advocate, Google - OpenSocial

Hello, my name is Lon Safko, co-author of *The Social Media Bible*, published by John Wiley & Sons, the largest book ever written on the subject of Social Media. Today we are here with Kevin Marks from Google; but actually he is representing *OpenSocial*, so Kevin is going to tell us about *OpenSocial* today, so let's just get started.

LS: So Kevin, please can you tell our listeners what *OpenSocial* is and how you are changing the world with this.



KM: *OpenSocial* is a way to provide social infrastructure for the web. It is a way of expressing information about people and about their friendships that can be reused by different sites and different applications. So it is a common standard resource for the social web.

It is currently deployed to a large number of social networking sites, including *MySpace*, *orkut*, *hi5*, *Friendster*, and several others in different countries around the world, totaling over 350 million users. And we expect that to continue growing as more websites pick it up over the next few months.

LS: Wow, 350 million users and you're bringing in companies from, not just the United States, but all over the world.

KM: Very much so; we've got sites in China and Korea, European countries and South America. What you find is there are different social networks in different countries, but they will have a lot in common and we can provide a common API that fits into all of them.



LS: What a monumental task! Wow! And some of the good things about this, too, are as you mentioned, there is 350 million users out there. That's a huge number and the work that you guys are doing is behind the scenes and it's really transparent to the user. The user's themselves are actually getting a better, more easy-to-use, robust product.

KM: Well, that's the point. The problem with building a social application is you have to gather the data. So that means you have put up a form for the user to fill in. They give you their name and their date of birth and zip code and whatever else you want from them. So that's quite a barrier that you put up in front of the users. You lose a lot of them when they have to fill in that form. And then again, I guess you want to do something that relates entirely to social stuff rather than just personal stuff. You link to them often by who their friends are which means they've got to get around and find out who of their friends are members of that site.

Now, the first time you do that, that's quite fun. You join the network and you've got some friends on it and you can get around quickly, looking at them and discovering stuff about them. But if you have to do that for every site you sign up for, that becomes a chore very quickly and people are reluctant to do it. Then you get bad work-a-round's that will come up with, "Give me your password and email for your web/email site and we'll draw on your address book from that." And that's training people to give away their passwords to the website. And it is also drawing on, essentially, a broader database than you are happy sharing with them.

So the value of the *OpenSocial* model is that you can have one or more of these trusted containers, trusted custodians of your data that you store the information in. And you can run an application that you give permission to, to draw on that data so that it can know about you and your friends and use that to customize the experience that it's having.

So by delegating some of your trust to the container site, that container site is then, on your behalf, giving that information through the application making sure it doesn't get more than it needs to do the task it's trying to do, rather than giving it a free pass to access your entire online presence. So there's value for the side-developer because they will draw on this larger pool of information, and there's value for the user because they do not have to re-enter all the data.

And the value in this for the container site comes in connecting them to the rest of the web, knowing what other things you are doing apart from what you are just doing on that site.

LS: And that makes a lot of sense, too. What I also heard is that different sites really have different applications as well, because that's true in real life. I have different social situations; the people I work with, the people that are friends from work, personal friends, relatives, family members. They all interact differently with each other and exchange different information and the relationships are different.

KM: Right, and you don't necessarily want them connected all together. So you may not want to pick one site and put everything in that; though some of the sites are getting quite good at managing sub-sets of people. But quite often you'll find that the soccer team has one particular way of organizing themselves. They set themselves up as a group on some site, using that style. You may be using *LinkedIn* for professional networking and keeping track of hiring people, and so fourth. So it's quite likely that you would have more than one presence on the web, and you may want to connect some of those together and you may want to keep some of them distinct. And the point is that you should be in control of that and not have to push everything into one site or, conversely, go into every site on the web with the same bit of information over and over again.

So the point is, as you said earlier, to try to make this stuff invisible to users would be a natural expectation for them. They can share personal and friendship information with sites which they go to on the web, but with this they get a behave-sensitivity factor with it, and no abuse.

That is a key part of *OpenSocial*; the user being in control of their information that is being shared.

LS: And I love that, because in writing *The Social Media Bible* I have some responsibility to participate in quite a few different groups. And I've got to tell you, when filling out every one of those profiles with identical information, even copy and paste, requires a tremendous amount of effort. So something like *OpenSocial* would really speed that up and help in making sure the information is transferred accurately, as well.

KM: Right.

LS: And another thing is that we were lucky enough to be able to interview Marc Canter. He's the founder and President/CEO of *Broadband Mechanics*. In a previous situation he was the founder of *MacroMedia* and *Flash*. And some of the things he's working on is *OpenId*, and he's been talking about that for a really long time.

How do *Social Media* and *OpenId* integrate?

KM: So, *OpenID* lets you prove that you own a website. That is its simplest function. It grew out of Brad Fitzpatrick's work on *LiVEJOURNAL* and other blogging systems to let people identify which blog was theirs; because blogs are a good example of URL's representing people. If you blog about somebody else, you link to their blog URL and say, "Kevin said whatever" to represent them. So that's a case where people are using their URL's to refer to each other.

And the nice thing about a URL as opposed to an email address is that you can then go and fetch that URL and get extra data from it, as well. With your email address all I can do is annoy you and send you an email, or I can give it to somebody else who can send you an email. If they want to interact with you they would have to email you their URL and you could click on it and come back to the site.

With *OpenId*, then you've got a URL that identifies you. And that means that a program can fetch that URL and use that as a route to get access to more data, whether it's *OpenId* data or even a webpage, or whether they do an "authorization handshake" using *AllOff* or *OpenSocial* and draw other data out from it that way.

So, *OpenID* is sort of a piece of this eco-system because it's a way of proving that you are your URL.

I mentioned *AllOff* just then. *AllOff* is open authorization. What that lets you do is give one site permission to have access to your data on another site. But it's a limited permission. You can give permission to some data and not others, and you can always revoke it in the future. So you're basically giving them a "valet key" that says you can do these certain things, but not everything else. And I can revoke that so you cannot have it in the future.

So that's the way for one site to draw on other site's information over time, which means that as you change the information on the custodian site, the other sites that draw on that will get the newer information, too, without having to re-enter it. So those two work together, and then *OpenSocial* draws on these underlying values too, and specifies the information content there.

So it says, "Well, here's what a personal profile looks like." It looked at many different sites and found what elements they had in common. And it defined them so we can find names and photographs and birthdates and the many other fields that we found a lot of sites had in common, including things like what your favorite music is, what your favorite books are, are you a smoker; information like that.

We log those fields by what they have in common, so we codify those. The other thing that *OpenSocial* lets you do is it lets you store data or assist events that are tied to user identify. And then it gathers that information about those events, or about that data, for both the user and the user's friends. So it solves an awkward problem that you get when you've got a large collection of data; deciding what's important to the user.

The only drawback that relates to that user and their friends is the application can do something much more interesting; because as a user you really care about what your friends think. And if I can give you some information that has a photograph and the name of your friend next to it, that's much more interesting to you than coming from a random user, or coming from a site in general. So it helps provide that kind of social filtering of whatever information that application is working on.

LS: And that's true. That point that you made about Social Network or "trusted network, is really important. I mean I think it is really the root of what's really driving Social Media, and it does for me. For example, any time I need to hire somebody or to find a particular vendor, the first thing I'll do is I'll go to my colleagues or my "trusted network" and ask around if they have any experience with a particular vendor. And if I get a good report, then at least I know, based on my respect for that colleague, that that recommendation is going to be pretty reasonable. It's a lot safer going that way than going outside of the network and just going through the Yellow Pages.

KM: Right. There is a great thing Douglas Hearn wrote that said, "The vast majority of our brain is based on deciding who to trust." We evolve this complicated empathy and evaluation mechanism in our heads to decide who to trust because that was vital to our survival. And so, that's not something you can straightforwardly replicate in a computer; deciding whose information you trust. But if we have the information tied to the people that you recognize, then you can apply the trust that you have in your head. So you know that one friend is good for recommending restaurants and another one is good for recommending books to read or one would be a great one to tell you where to get your car fixed. And you know that about those people. So when you ask the question you can rate their expertise on their "individual" selves without the computer having to do that for you, because you are relying on the knowledge that you already have about those people in your head to evaluate it.

LS: That's really psychologically profound. That's pretty cool. I never actually really thought about it in psychological terms, but that's absolutely right. And if we can actually get an API, a platform, a technology, that can help assist in sorting all of that, that would be a tremendous, helpful tool.

KM: And to tell the computer all that information and knowledge you have about people, it would be really hard and wouldn't be worth your while to do. But you have all that tactic knowledge of the people you've met and what you think about them and you can use that to tally the information you see coming in.

Conversely, once you've connected to a set of people and you're getting updates from them and information that can drive your opinion of them in the future. That's one of the things that you see in activity streams in *FaceBook* or in *Twitter*, or *FriendFeed*. These applications give you a flow of information about people. If they're people you don't know, it just looks like a noise and a mess. If it's people you do know, it becomes interesting signals and bits of knowledge about them that you can take in and then respond to, or not respond to. But if you look at the public *Twitter* stream, it's a bunch of random weirdness from strangers. If you look at your personal one, it's, "Oh, my friend's in New York today!" And that's a very different type of information because the context is supplied by you and you've conveyed the context of who you're interested in.

And the value of the Social Networks is that they have all these different context groups, or individuals, which means any application that's re-sending info can present filtered through those contexts so that your friends mediate the information and they can make it make sense to you. And that's the sort of value that's broad, but it's hard to do without a rich set of connections and information about "who knows whom". So being able to bridge that information from the sites that have lots of it, to any given individual application, is what *OpenSocial* provides.

LS: One of the other things that I'm hearing, too, is there has been a lot of concern...I mean there is always a lot of concern and paranoia about personal information being distributed through the internet, and actually kind of losing control. And there's an example, and I won't name the name of the company, but they made a little bit of a mistake when people can go and check all the websites that their friends have been surfing. Well,

maybe there's some websites that their friends didn't want them to know that they were surfing. And that, kind of, put a scare into this, kind of, "open-social"/open-information" exchange, because giving personal information is one thing, giving behavioral information is another way to collect. Such as the information that Google uses when you are using *Gmail* and it's pulling key words out of your email and serving the appropriate advertising.

But what I'm hearing here is that *OpenSocial* really isn't, kind of, a free-for-all information-exchange, that the users themselves really can pick what's personal, what's private, what's behavioral or what's in it. There is control on this.

KM: The point is that you want to make sure that information is not shared without your permission. You don't want two different identities joined up. You don't want your *LinkedIn* profile connected to your *SuicideGirls* profile, necessarily; unless that's something you've chosen to do. So part of this is that people "facet" their online existence, they segment it by using different sites for different purposes. And sometimes there's value in that and sometimes you want to keep them separate, and you want to stay in control of that.

And there's a danger in putting everything into one specific site and trusting that to everything...because it can cause [15:39.5] to leak without realizing that's a problem to you. You know, you can decide to change the information flows and then accidentally cause you some embarrassment, because you thought that things that were in two separate compartments, no longer are.

So one of the things that are part of the design of *OpenSocial* is this idea of the user giving permission and/or authentication before any connections can be made; but also because of this separation between the application and the container that's supplying the data. That's "policy layer" there, as well. So different containers will impose different policies to fit in with the social undersign and the social contract they have with their users.

So, for example, on home sites the assumption is that anything you say on the profile is public and visible to everyone. Other sites are only visible to a select group of friends. You can just choose different levels of permission for different fields. So when you install an application, there's an expectation for your public side that could see everything. So there's an expectation on the private side that it could only see things that it's been permitted to see. Now understanding that as an application programmer, is hard and you will be liable to make mistakes if you are just told, "Oh, you're not supposed to show that to people." But by just calling API and saying, "Can I have information about this? And the container then imposes its policy and says, 'well this information's public. Yeah, you can have that. That's private. We're going to have to ask permission of the user before you can have that.'"

The [17:16.2] doesn't need to know that detail and doesn't even need to know that it's been denied the data as it might just not exist there. So there's this sort of separation there between the [17:26.9 ap] and the container, which provides a trust boundary. So you can say, "Okay, I've trusted this container with a lot of information, but I'm only giving somebody some of that over the (lines) that can convey the application.

LS: So this is controllable by the user?

KM: Probably. It is controlled by the user potentially, or it's controllable by the container because the container sets policy; so there's a

general policy for all users and takes that expectation into account. The way any given site behaves has something to do with its history of the different kinds of users that have come there and they way they shape that site.

LS: Expectations.

KM: There's a set of expectations there. You know, you've got to (turn) the service that you click through when you join the site, so than nobody can reach those, but the fact of what you do on that site defines the information that site has and the expectations you have for that site. And the sites can understand those expectations and not violate those for you.

So when an "ap" runs in that environment it will take on the context of that environment and the social mores of that environment. So it will see more or less information, depending on what you decided to share with that site; or more of your friends' information in the same kind of way.

LS: Well, I like that and I think that when people realize that, you know, *OpenSocial*, *OpenID*, that really there's some controls on it and information isn't getting exchanged freely. I'm excited to hear that because, yes, certain information to my *MySpace* account...well, I want that as public information, but maybe only for my friends, because as you know, you can't view a lot of information in *MySpace* unless you invite a friend. And one of the things that I find interesting, too, is that after talking to HR managers an awful lot of them actually go in and look at clients of their potential employees. *MySpace* page is to really find out what they are doing on their off hours. So, if you've got a lot of pictures, if you're out there being drunk at the local party last Friday, probably doesn't play well with an HR manager who's looking to hire you.

So these different levels of information, I think, is really important.

KM: Well, that's part of this compartmentalization issue and that's "understand" who has access to what. And people may not understand directly what they are sharing with whom. That can be a problem with this site, but that happens often even without all this detail. That's true.

That reminds me of another aspect of this, which is that the other thing that we are seeing with *OpenSocial* is by the "abstracting" way the social infrastructure...that can actually apply to more than one kind of contact. So, most people talking about social network sites up to now...they're the ones that took this up first. But we are starting to see other kinds of sites that work with *OpenSocial* as well.

One example is...like "dashboard" sites. If you think about *IGoogle* or *MyYahoo*, those are much less personal profiles sites. The information dashboard goes on the web that is drawing information from outside.

LS: And match-ups...

KM: Yeah. And basically all sort of little things that you imbed in there that tell you stuff about the world. Those have the same problems that the websites do, which is, for each of them you've got to say what actually is a "news zip code"... "I care about news about this topic." And you tell them this stuff about yourself and they don't really have any context about you.

So by applying that...by creating this “shared” context across the whole site, then these individual applications can draw on that without you having to re-enter the data. So that’s an area we’re going to see expansion of soon when *iDoBlog* launches and we see this to be on other dashboard-type sites.

LS: That’s a really good point; too, because, every little widget requires pretty much the same information about me in order to function properly. And, in fact, should just be automatic. I mean *Apple* uses *Keychain* for passwords. I love that! And “*RememberMe*” on a website because it saves me the trouble of having to remember to put in all those passwords every time.

KM: Right. So you can imagine the environment having that context for you and supplying it to different application.

Another kind of container that we’ve designed and see being built is...if you think about doing the HR in [21:44.9] and you think about customer relationship management; sites like *SalesForce* or Oracle’s *CRN*. They have large databases of people and information about them. And they have a similar issue of wanting to be able to do innovative things with them, you know, with applications around them...and what we found when talking with them that open searches [22:02.3] fits that as well. In that case you still want the same thing. you can ask them not only about the person you use in the system because you have logged in to use it, so you know about them and their friends and colleagues, but because you were a database of customers, you can have access to the profile because the profile page is, in effect, the customer page, the information about the customer that the company has created about them.

And you can use that as an input to the applications, too, and again, draw on the information about the person using the site and the information about the customer and use [22:33.7] applications, as well.

So it was the same model, actually, that fits in several different places. That is this idea of the viewer of the page and the owner of the page; and who the owner is varies depending on what kind of site this is.

There could be one session where it could be your profile in a dashboard, where the owner and the viewer would be the same. And in the *CRN* system the viewer is the user; and the owner is, potentially, the customer that you have information about.

So, coming out with this common abstraction we found that it did fit in a very broad range of possible sites. This meant that the applications can really move between many different kinds of sites and still make sense.

LS: Wow! That’s exciting. Again, I never really thought about applying this open social to all these different areas, these widgets, to social networks as passwords in ace, any form that you enter information into, on the internet.

KM: So, that’s right. You mention widgets. That seems to be first interaction in *OpenSocial* and the thing that is widely deployed now. It was designed running gadgets inside sites. So you’ve basically got an application imbedded inside another site. And there is a bunch of advantages to that, because that means that the imbedded application gets access to all the sites’ users and the authentication is much easier because the site is managing that for you...because it’s wrapping that up in the gadget inside it. So that was the direction that we’ve now got coming, deployed to a lot of sites.

But something is coming in the next generation called *OpenSocial Version 0.8* which is coming out over the next month or so. And that has what we are calling “rest API’s” which means that you can do server-to-server calls; which means that one web server, or one web application on one web server, can call into the Social Network remotely, find out information about you, and use that while you are surfing the web. And it can feed updates of what you’ve done on your back site back to the home site.

Now, obviously, that is one of these places where you’ve got to be careful about permissions and authentication, which is why it’s taken more work to do this than just the [24:37.8] applications. But we’ve got that so, now with [24:40.6] opted to distribute the information. So now this, instead of just being able to work on gadgets imbedded in a single site, you can have this work-spanning knowledge of who you are by drawing on the most sites which you keep up to date...and by feeding the flow of what you’ve been doing elsewhere on the web into that stream. So, if when you go and use a book review site, that could be fed back to your Social Network so that your friends can know your books views from an external site, as well.

So that’s something that’s new in Version 0.8 and that should extend the range of social groups to a much broader range of websites, as well.

LS: And that’s also exciting. I saw a presentation done by a company here recently, which kind of scared me; but then when I thought about it, it was technically fairly simple. The demo was that somebody was looking for a Honda Truck, and they viewed a couple of Honda Websites and then eventually then drilled down into the site, found a particular model, a make, a year and that’s the particular truck they looked at. Now they just decided, “Well, let’s go and see what Nissan has that’s comparable.” So when they clicked on just “Nissan.com” instead of going to their homepage, that it captured that behavioral information, and didn’t make them click 10 clicks down. It actually took them to the make, that model, that year, exactly the comparable truck that Nissan made compared to the Honda, based on behavioral information, which served up a “best-possible” page instantly!

And then, of course, when they went to Toyota the exact same thing happened.

KM: Well.....that seems to escape what we’re trying to do, but again then I can see some potential there. There they’re, to me, running something client-side that is keeping track of what you were doing, to do that.

LS: Yes.

KM: Right. So that’s...yeah, I could sort of see how you could do it that way. But that isn’t part of this model. This model is, sort of, is server-to-server data, with some assumption that stuff is stored server-side. So, you could use things a bit like that, but it would be...it wouldn’t be [26:51.6] as implicit as that. You would have had to actually stored the preferences in one site and then draw them out in another site.

LS: Sure. But, I mean, it could very well go that way where pages are served up very specifically.

KM: Sure! The other thing is that if you’ve given the site permission to know something about you, it can make and customize the display. And whether that’s personal demographic or reference information, or whether it’s...”here’s a set of your friends and what do we know about them? Have they interacted with this site?

One of the problems you've got with sites is that, if you think about comments on sites, there is a problem that... when there is a small number of comments, it's fine...and when there's a large number of comments they tend to fail because you cannot publish or read them all. And the normal mode of showing you the comments is them being in chronological order with the most recent at the top, or something; or maybe even in some "who-replied-to-who thing". But that doesn't act as filter...a good filter a lot of the time.

But actually at that point, you don't want to see the comments. I think with a restaurant review site, you don't want to see how numbers review the restaurant. But if a couple of your friends review that restaurant that would be really interesting. And if it could know who your friends were and popped those right on the top, that would be much more valuable for that site than for you, then showing everything and making you have to flick through it yourself.

So it can be, you know, subtle adjustments to have this information as presented, but by drawing on this information about you, that is an example of where it could be useful.

LS: That's very powerful. And for the people that might be a little bit afraid of it, too, I mean we've been kind of use to it on at least one level of this. Whenever you go to Amazon.com to order a book, it comes back and says, "Here's the other types of books that people who bought this book, also bought; and you may be interested based on behavioral information of all the other purchasers."

KM: Right!

LS: Okay.

KM: But I can get mixed up on intent as well. I don't know if it's a good thing because of purchases. That's actually a pretty strong thing because they are actually giving their money for something. But it tends to get a bit confused because if I've been buying things for my kids as well, it would then show a random mixture of things that are things for me, and things for them.

LS: This will stick to your network.

KM: Yes, because the behavioral signals are, otherwise, clear. An *Amazon* that knew what my friends were reading would be really interesting, too. I can actually see that that would be powerful.

LS: That would be really powerful. And I would be interested in that as well, because if I respect somebody, I would be curious to see the types of books that they are reading, because they may interest me as well.

KM: And that's using it in exactly the kind of application we've seen on *OpenSocial*. One of the sites wonders "what are your friends reading"-type of application.

LS: Wow, I like that a lot.

KM: It is quick and easy to tell what you are reading and it will poke you every now and again and ask, "Have you finished that yet?"

LS: (Laughter) that's remarkable!

KM: And then you get this sort of “flow of recommendations” from there, which is, “Did you know your friends are also reading this book?” Or, “Would you like to suggest this book to somebody else?” And they just exist by doing that and then sending you off to Amazon if you want to buy the book, and they take the referral thing. I thought that was, absolutely, a very clever bit of application design.

LS: That’s very, very clever. Wow. Kind of “contentural” sharing through trusted networks.

KM: Yes!

LS: Wow. That’s very powerful. I mean, I’m just thinking that you have this tool set, this skill set; all the different applications that you can apply to it, and it just makes like on the internet that much easier for all of us. That’s exciting.

Now, about...you actually work for Google, but you’re herding cats. You’re actually bringing in perceived competitors such as Yahoo, and so trying to get all of these kinds of companies into one room. How’s that been? Is everyone, kind of, trusting one another? Was it tough at the start? Do you have a critical mass?

KM: Well, it was kind of interesting in that when we first started talking about this, how quick they were “in”; how happy they were to jump on board and get involved. I think that it’s...intellectual problems are hard. It’s always hard to get people to agree to do something. But if you can draw them together you can find the common interests. And then what’s been going on since this? That was the first launch of this last November in which people said, “Yes, we are really interested in this” and we got to work on it.

And then over the last eight or nine months, that’s what’s been going on. There is a whole bunch of engineering going on in this space. A discussion group, two open-source implementations in the *Shindig/Apache* project; which has been built by many different companies. And then they are now deployed in different Social Networks. So we’ve got, you know, *hi5* and *orkut* running the same basic underlying code to do the *OpenSocial* layer, and then we’ve seen that *Friendster* and some of the European networks are running the same code from the other branch of *Shindig*.

So, there is actually a lot of collaboration on the call code as well as on the specification. By being able to do that work on the net that makes it actually much easier to coordinate this work, and have stuff happening that way. By making it open source, you’ll see that this interesting benefit is that people can get involved with this without actually having to come and talk to us.

So we found on a couple of occasions that some implemented *OpenSocial* specification and they joined the group afterwards, and told us they’d done it.

LS: Laughter.

KM: That was pretty good. And now they’ve said they are contributors to specification and we are working together that way.

And another company called *OnLabS* from Korea, they came to visit here and I was expecting to just give them the information and have them say, “This is wonderful and we’ll have to think about it.” And they said, “On,

no, we already built it and we downloaded from *Shindig* and we've got it up and running. We've translated it into Korean and we've put a website in Korean so that Korean people can do applications for it, as well."

So the power of "open-sourcing it" from the web is that putting stuff out there, it can find, again and in the same way, it can find the people who are interested in it. They can feed it back into the community. So that's something that we've seen that is very strong about this. And the nice thing about the Open Source community is that it is mature enough now to have these kind of practices and models like the Apache Foundation. That means that we, sort of, know how to do this kind of large-scale collaboration between two different companies now. And that's, you know, the way the web works...is through a set of agreements between different people who do the same kind of thing. So this is building on that model and then doing the same thing for Social that we've done for connectivity with PCIP and switch documents with them, and actually HTP and HTML.

LS: And one of the things I am going to stress is that I hear you mentioning, not just companies, but countries. Because this is just not trying to herd the American cats, because we're U.S.-centric; but this really is trying to pull together...and I'm going to use the term...trying to pull together the entire world...into a standard.

KM: Very much so. I mean, that's part of...you know...to some extent code is the common language; and you can run code and it doesn't care what language you are speaking as a user. And there are cultural differences and things like that, but there is a bunch of commonalities between people that actually means that different sites can work together on a standard, even though they may have a different set of users and they may have to adjust how they behave a lot on the particular sites for it to work.

We are seeing that connection between the different sites. We are also seeing that we can develop a different community in different countries, as well. So there will be a popular, local social network that everybody's talking about, and they build an application for that and then they may think, "Oh, we can take this one, or the other one. We could take this from another country, or we could partner with a U.S. company, or we could partner with a German or Korean company, to move to their world."

We are starting to see those kinds of extensions form, as well. Because you've got this shared platform that people can build upon, it means that the effort of moving it from country to country still takes some work to translate it and check for cultural misunderstandings, and so on. But that's much easier when you can actually get the code running and then have a conversation with someone from that country who has done something similar already.

So we are seeing that community grow, and we've got a series of Google Developer Days, which we're organizing across Europe and Asia, in the fall coming out, and we expect to see even more of this. We going across Europe and then we are going across Asia and meeting developers in each country and working with the Social Networking companies in these countries, as well, to bring those together.

A lot of this is not so much organization as just making...creating a space for people to get together, and so we're helping them do that.

LS: Wow! Again that's such a...if someone approached me and said, "How would you like to try to manage a project like this", it would scare the heck out of me. But it sounds like that because it's the right thing to do, it benefits everybody, it seems to be spreading virally and people really are cooperating because they know it is in the best interest of everyone, including themselves, personally.

KM: Well, it's setting up the system such that you can collaborate on the thing. You know, you can agree to agree and you can really disagree. So, you can find the stuff you have in common and you can agree with that. And then you can find the places that were a little bit different and specify a way to ask the question. So the programmer can say, "What can I do on this thing? Can I send messages to people, can I see who's browsing the page or do I have to log in first?" Ask those kinds of questions and then write an application that does different things, depending on the answers, and then it will work across a higher range of sites than just having a singular function about how behavior works.

LS: Wow! Those are some really incredible insights, and as not being an API developer or a code developer any longer, I wasn't really aware...I mean I heard the term and I've read some articles about it...but I really didn't understand how profound it's going to effect me as a user. And, again, the most important thing that I'm seeing here is that all of this really is going to benefit the user, but it's also going to be done in a very intuitive, very protected and very transparent way.

KM: Well that's the thing; it has to be, you know. The other thing about creating a collaborative project like this is that it has to proceed by agreement. So if there isn't an agreement then that piece won't proceed. So, as long as you can not make those bits block the other bits, it can keep growing. So we started out with something fairly straightforward, it was just exchanging ID's and photos and names, and then we went through and looked at all the sights and saw what they had in common, and grew that to a broader profile.

And then in the later generation, we've added a few more boosts on that profile and we may find it a bit further. And one of the things we added was a way for one site to represent an account on another site. So they can actually bridge it together. So there is a series of these things that you add and you say, "Okay, we can see that this stuff makes sense." And then you find, "Oh, someone built this; and this other thing will be useful, too. Can we add that into this thing and grow the agreement, as well?"

As long as you start from a core of stuff that actually works and then grow it, you're much more likely to get somewhere interesting that if you try to define the whole space to begin with, you know, trying to start with the whole universe and specify that. But you start with a core of agreement and see how you can grow the agreement.

LS: Makes a lot of sense...that really makes sense. Wow!

Well, lastly, is there anything else that you would like to add? Again, our listeners are users and probably very few of them really a familiar with *OpenSocial*. Is there anything you want to summarize...what the future is or what they can look out for?

KM: The thing to think about is the big, sort of, model shift. It is to stop thinking, "I'm doing a website", and to say, "I'm doing a web application that will go and find its users. How can it separate itself from the information about the users it needs. You see, "build to that model" means that you can take the application and move it through all these different sites with 10 to 50 million users, and more coming, rather than have them all have to go to you site and tell you everything over and over again.

I suppose that what I am saying is that no website is an island anymore. So you can take your application and move it through other people's sites and help it find the people that are interested in it. That's the sort of "key"

transition here, is moving from, “I’m a [39:38.2] file on the web, come and find me; to, “I’m an application that moves through the web to find the people that are interested in me.”

LS: Wow. That is awesome! So basically when you go home at night and you get in your car and you turn the key and look in the rearview mirror you say, “I’m changing the world!”

KM: (Laughter)...well, little bits of it! That’s the idea.

LS: A piece at a time.

KM: ...making the web better.

LS: I love it! So, Kevin, is there any place at all that people can find more information? Is there any blogs; is there websites, if they’re interested in getting more information about the work that you are doing?

KM: You can go to www.opensocial.org. That’s the best place to start. That has information about *OpenSocial* and links to development resources and up-to-date news, and so on.

LS: Excellent! That’s really excellent. So I would really like to thank you Kevin. Thank you so very much, really, for being here and for sharing your time with us and talking about *OpenSocial*. I learned an awful lot. I truly appreciate it.

And this has been Lon Safko, co-author of *The Social Media Bible*, published by John Wiley & Sons. Be sure to check out our other valuable social tactics, tools and strategies that can be found in *The Social Media Bible*, as well as its companion website, www.thesocialmediabible.com.

For more information about me, Lon Safko, or to learn how I can speak at your company or at your next event, please go to www.lonsafko.com.

And, Kevin, again thank you so very much for being here and thank you from our listeners.

KM: Thanks very much. Appreciate it.

LS: Thank you.