<meta name="viewport" content="width=device-width; initial-scale=1.0; "\>Event ID: 1857862
Event Started: 12/8/2011 4:00:00 PM

>> Good morning everyone, they give for joining us today. This is the Web manager University and the mobile program manager office go presentation of a program, mobile web versus mobile apps.
>> My name is Gwen, I am director of mobile here. We are super excited today to have some really smart people to help us answer the number one question that we get in the mobile program management office which is, what should we do, sure make a mobile website or should we make a mobile app? --
>> Today we're going to be joined by Neil Bonner, who works in the information technology field at the transportation security -- he is a focus on mobile application developer, social media and collaboration platform.
>> Were alsoamong is going to be debating on the side of the website.
>> On the other side, in terms of web app, Mike Pulsifer is the manager of the department of labor , and image or in the public -- public affairs. He is responsible for Web and Web application developer, and for external and internal users. They're both very smart and very knowledgeable people to be joining us.
>> This debate will be moderated by Guthrie from the protection agency coming is the scene of technical advisor, he is the -- in the space as well as doing a lot of interesting things in technology.
>> Before turnover to do, did want to let you all know that we are structuring this debate format, we have very generous guest to take their extremist position in this debate format. Especially for people that may be online from the press, please recognize that the extreme positions are not necessarily board the positions of the taken their own jobs or in their -- I just want to be very clear about that because to make it more interesting and fun have really asked them to step back and take off because.
>> With that, wanted to know, I have a Don King hairdo on, as if a promoter here, I would turn it over to the bring to Delroy.
>> Thank you thank you.
>> Okay, everyone, will were going to be doing is going back and forth so each can speak their positions. We will then move on from there with examples and do kind of a point, counterpoint format after that. If you have an opinion, not the counter to start off, we're going to ask Neil to make his opening remarks on why he feels that the mobile web is the way to go.Neil?
>> Thank you Delroy, appreciate the introduction there. I'm just going to share my title screen for this opening statement. The first thing I wanted to do is define, what is the mobile web what we're talking about mobile applications, -- apps versus traditionally called the mobile web.
>> Quite simply, the mobile web is to take HTML ages, the standard HTML technology and format those pages for small screen sizes.
>> When we think about mobile, were generally thinking about smart phones that we carry around with us and those smart phones generally all have browsers and those browsers can display webpages.
>> Because of the small size of the screens, to take a full-size webpage that is designed to be traditionally displayed on like a desktop computer, where you have got lots of pixels in either direction, when you shrink that down into a mobile device, it really becomes very difficult to read.
>> What mobile web does, it uses that same kind of technology comes a standard HTML technology, CSS, HT and 04, the standards have been in place for well over 10 years. The format is that is easy to read and navigate using a smart phone, so using her finger to move around on the screen and through pages and so forth.
>> That in its most simplistic term is what mobile web is. There's not a lot more to elaborate other than that, other than to say this HTML technology formatted specifically for small form factors.I can pass it back to you Delroy Okay, so now we have had meals open statement. Mike, give us an idea what you think mobile apps is the way to go.
>> Sure, do have the visuals? The visuals I provided. Anyway, regardless, apparently, technical difficulty there.
>> So, native apps come over all those native apps today with the martyrs more fun than the various app stores and marketplaces that the providers provide. They're

becoming far more popular means for people to interact with the phones nowadays.
>> Native apps are the kind of applications comments are running a browser locally on the phone, independent of any other app that may require resources. Native apps can be optimized performance the best under the particular user's platform.
>> Instead of writing code that would have to run the same, hopefully with the same performance, and android or IOS device, or library window phone seven, the what -- could you are running a specific to that platform. They can access system resources such as your contact list, photos, music and videos, so you're not necessarily isolated from the rest of your device.
>> They can also make use of the device as hardware. For example, the camera -- camera cheap -- GPS etc. Each platform has is own I calls that allow you to take and post photographs, utilize the horsepower of your phone, which amazingly is probably more powerful than many of the computers that we have on our desks even 10 years ago.
>> You can use that horsepower to edit those photos, you can even use the Excel are on to -- actor movement such as if you shake your phone, there are several apps, for example can measure how many steps you have taken what you're exercising.
>> Native apps can also go for the user of important news or updates on their own. Rather than having to open up an app to find out what it has been aching to tell you, native apps can actually tell you what they need to, when they need to with a centralized notification interface.
>> Native apps basically bring the power of the desktop into your hands. That is my opening statement.
>> Okay, thank you. Neil, can you not give us a few reasons or so on why you think that the mobile web is the way to go?
>> Sure, I would be happy to. So, there are some really compelling reasons why you want to go mobile web. Firstly, and I group these together, they're fast, cheap and commodity skill sets. They are all related.
>> Fast means to develop a mobile application can take some time. If you think back to what I said in my opening statement, we are talking about HTML technology. This is technology that has been around since the mid-1990s, the current standards of HTML 4 and the evolving HTML 5, have been around 10 years or sooner.
>> Commodity skill set, virtually everyone in the IT field knows how to do HTML. As far as the cost associated because the skill sets are fairly pervasive, it's fairly cheap and you can do this very quickly.
>> If you have got your requirement and held down, fast, cheap and commodity skill set is number one with a bullet as too far as why you'd want to use the mobile web as the choice for your mobile application.
>> Number two is that metrics. It is easy to measure usage with mobile web. You can use a standard analytic products like Google, Google analytics. That allows you to integrate with your existing analytic products because everybody has a website and on your website it is highly likely that you are using some type of analytic tool to measure the usage, how many page views do you get, how many visitors do you get, what devices are they accessing your pages on?
>> That same set of analytics that you are using for your public website, you use for your mobile web. It is there, it is built in already. The technology supports that. It is just dropping that code into your mobile web.
>> So, metrics and analytics is a huge reason as well.
>> I third reason why you would want to use mobile web as your mobile solution is that it is nearly 100% compatible with smartphones. All smartphones I guess by definition have a web browser. Any phone that does have a web browser and that web of browser is let's say, someone is, up to date.
>> For example, we have got some older Blackberry devices here and some of those older Blackberry phones, their web browser is a little bit spotty so you can get some differences of their in phones.
>> By and large, you are one of percent or you are close to 100% compatibility with smartphones. You build it, one and everyone can see it. You do not have to worry about, well, this audience is -- that is years away iPhone only, that is just one segment or you are talking about building for android, Blackberry, build it for whatever.
>> With mobile web, you get 100% compatibility out of the box. Fast, cheap and commodity skill set is number one.
>> Number two is Metrix come you to metrics really right out of the box and all

management is going to want to know is who is using it, how they're using it, what devices are they coming from.
>> Three is the compatibility, website compatibility. It is pretty hard to argue with those three things and I would like to see Mike try and try to shoot down of that argument. I do not think he could.
>> M?
>> -- Mike?
>> That is going to be quite easy. As far as the cheap -- it really does not get any more, place the CBEs languages that the native app platform use.
>> To see but -- like which has been around for decades. The core of any computer science program, and any of the colleges, and even the community colleges around us.
>> For android, even the.net and Objective-C are based on sea bass languages. Is actually pretty easy to move from one language to another. Any competent developer should be able to pick up any of those platforms no problem.
>> The -- are so solid nowadays that's it is so easy to get something built quite quickly.
>> As far as metrics, metrics are really great for static websites. Problem is, when you get into the applications themselves, there is a bit of a creepiness especially if you are looking at what people are, what specific individuals are accessing, what type of information they are requesting and what they're actually doing.
>> As far as compatibility is concerned one of the things that we have seen is that that compatibility that we hope it would be there because the vast majority of the mobile platform are using Web Kit-based browser, is not quite there because they are all using different builds of the Web Kit.
>> We have seen different behaviors in Safari, IOS, and the android browsers. Of course as you have mentioned, Blackberry, Blackberry is a really big problem especially in the older phones that do not have the Web Kit is to browser.
>> So, I consider that shot down. Do you want to go into these three simple reasons to go native?
>> Yes I'm a please.
>> Only three simple reasons to go native, one of the once you get out of the big cities, cellular and wi-fi access can be really spotty or even nonexistent. This is something that I see on a daily basis when I am taking the train home.
>> It is too easy to get the idea that PCs the represent of the rest of the company -- country. It is really an incorrect associate. On the way home, I lose signal several times. There are several areas where signal is weak it, even around Gaithersburg. That is not exactly rule.
>> Back were I live in Martinsburg, I get 3G access with one provider, but Verizon does not provide 3G access. It is even inconsistent between the providers.
>> Because bandwidth is often To and speed are unreliable, it is best to avoid downloading user interface bits unnecessarily. Native apps can have these elements built-in, greatly reducing the quantity and weight of what gets downloaded.
>> The apps only download what they need, when they need it.
>> And of the reason is security. Here in the government, we have got these really steep IT security requirements that we have to deal with. As a system owner, the last thing I want is additional security headaches due to personal at that file information or other sensitive information being stored on our systems, storing user data locally using apps for -- that are sandbox rather than on agency servers along with API used can ease the IT security strain on the existing agency system.
>> We have having the paperwork to do, we do not need more.
>> The third reason I have for going native is that going native gives users that UIs that make sense for the device. Granted, android might be the wild West right now, but other platforms and detailed conventions are regularly enforced and they are in force for the user benefits.
>> The example here, I have three different apps that interface with Wikipedia. Wikipedia has a mobile web interface, though when you -- but when you have a native applicant top of that, is a much richer interface, there is more power that is given to the user and it makes using Wikipedia that much more easier.
>> With these three different examples, we have got IOS, android and Windows phone seven. With a mobile web interface, they would look the same for everybody. They would not conform those platforms conventions.
>> A more concrete example of differences between the platforms can be demonstrated with our labor stats app. IOS has its own convention for the tab-based interface

with the tabs based on the bottom, android has them on the top.
>> Those are the three reasons why going native really is the way to go. One is inconsistent and are -- unreliable cellular access, security and the native UI conventions that are different between each of the can -- platforms.
>>Thank you, Mike. You made a very easy for me. Sorry to interrupt you Delroy, did you want to say something?
>> I was going to sink Mike and say Neil, you must be teed up now for letting Mike know why he is wrong. Can you jump in now and give us some examples?
>> Sure, couple of examples, this is like shooting a fish in a barrel. It is easy to parse through some of these arguments here.
>> One, Mike talked about location and signal not available everywhere. This is a good screen to put up there. Here is the Mike TSA application, if you're not familiar with that, we'll talk about that little bit later. It is an application designed to be a trusted traveler companion as you go about traveling to the nations airports and whatnot.
>> By way of an example am a useful applications do need to interact over the internets and grab real-time information that you are pulling down to help you make decisions, to help you know when things are going to occur and so forth and so on.
>> To say that you do not need a signal for a native application is like, what are you going to do, play and groupers on its? I guess if you're going to play a groupers and that is your application, then you do not need a signal to -- they've to transmit your score back.
>> If you're playing words with friends, like my buddy Alec Baldwin, you need to be online because you are sending data with other players, was things like that. The application that are really interesting are the ones where there are some components or features of that application that aren't grabbing information in real-time or semi-real-time.
>> Weather is a mobile web or a native app, you're really going to need a signal there. The last of the signal -- the loss of the signal is certainly an issue I think in practical terms everybody is going to need one to make full invention of an application.
>> And optimizing download, you can take a look at this at here. You can make it look very a track is and optimize the graphics. These pages are very lightweight and it really doesn't require anything like a heavy download would be.
>> I think the download argument really does not apply if you can optimize your pages and your graphics, which you can.
>> The other thing on to talk about, is IT security. I am shocked that you would say that the advantage of a native application is it is easier to secure. I think the truth of the matter is it is just the opposite. I -- when you're talking about IOS applications for Apple, or if you're talking about android applications, you are running these applications on particular devices.
>> There are so many, there is so much in the way of having to lock down areas pieces of the communication between the device and let's say a backend service that you're offering.
>> If you're checking, like IRS has an application where you can check on the status of your tax return and so forth, you want to make sure that that application, the date of application is secure and it cannot be compromised in some fashion to get access to other Texas turns that you are not authorized to see or things like that.
>> Our expenses are IT experts group here at TSA suggests exactly the opposite of that there is a significant IT security component that the folks that are responsible for granting accreditation and authority to operate in ATO, have to go through. They will put you through the ringer when you build an application weather is for the public or for internal use as far as ensuring that it is rock solid secure.
>> We talk about the mobile web, on the other hand, it is HTML pages, like you're publishing an HTML page or an application that is externally facing. The IT security groups are familiar with those kinds of technologies and what is the proper way to secure those things.
>> If you're talking about an ASP type application that is public facing, that is easy. These guys have gone through those things for 10 or more years. Security is fairly simple.
>> A couple of other things that might do..., but will he actually did mention, is cost. The cost of building a native application is significantly more expensive than

a mobile web. Skill sets are big to argue. To try to find someone that knows subjective see which is the language that Apple uses for building IOS devices and using X. code which is the IDE for that, that is not a skill set that is readily available.
>> If you try to hire someone like that, if you have gone through that process, you will understand just how difficult it can be and if you get someone that promotes themselves as being a competent builder of IOS applications, ask to see what they have built.
>> More times than not, you will be fairly underwhelmed with what is out there. Difficult to find good people and we are all after the same good people, so they are in high demand and therefore they're going to be very costly to do.
>> Because you do have to go through extensive IT security reviews, and to build it into testing you have to have different devices, you have to worry about what version of the OS on that smart phone you are building for and so forth, it just takes much more time to do, so the slower and more costly. Skill sets are more go to find.
>> Again, gets into metrics. Management wants to know the utilization. So if you're spending a fortune on a native app, they're going to want to know how many people are using this app and just by saying, I went to the iTunes connect portal and I see that we have got 20,000 downloads, that is not going to fly. They may have downloaded and deleted the app because your app is crappy.
>> They want to know, is it being used? How often is it being used? Those are the kinds of things that management wants to know. It is hard to measure the success of the program if you're not measuring things.
>> Mike really did not address metrics other than to say that it is creepy. Well, you may think it is creepy, but we made leadership does want to know about usage.
>> Limiting the audience, getting back, for the advantages of for the mobile web, you can set those are the disadvantages of building a native app. If you build one for IOS, you haven't done anything for the large installed base and that there is a large installed base of Fort android phone. If you want to hit the base, now you have to build, find a skill sets just like you did for let's say you're IOS, Apple application. You have to do the same for android and potentially for windows seven phones if you choose to try to take care of that audience.
>> They all have different design considerations and whatnot, so you have just multiplied the cost tremendously. Sorry, Mike, that was a good try but I think I shut you down pretty good there.
>> Delroy Thank you Neal.. Neil is wanting if you could go into why HTML 5 might be our savior in the future in terms of what it will bring to the table and the mobile environment?
>> I would be happy to. What I was talking about before, being technology that has been out there 10 or more years, really talking about HTML 4 and the cascading style sheet standards, or CSS as it is commonly referred to, but what has been in development for the last four or five years is a new standard of HTML, HTML 5.
>> HTML 5 brings a whole new set of capabilities to the table that we have not had before. Really, HTML 5 is designed to augment your browser based website and make it more applications like.
>> That is the difference I want to to understand. The traditional HTML that we have built and so forth, it generally is that HTML 4 set of tags and codes.
>> That is designed primarily for website, is putting up there. HTML 5 is more about application and how you interact with a site, with an application. It is really application.
>> For example, it has do tags like the video tag. You do not need plug-ins anymore, if you want to play a particular video in a certain format, you need of all of these plug-ins for but the video tag, with the audio tag, with the canvas tag, there is no need for these plug-ins, you do not need flash, you do not need some of these other things because it is built into the specification.
>> Also, HTML 5, has off-line capabilities. Mike mentioned, what happens when you don't have a signal? Well with HTML 5, there is an ability to store data locally or to use the application as a self-contained application. You do not need to be connected all of the time.
>> There are some capabilities to do that data by the elation locally. It also offers Dragon --. That is new. If you use Google Melany want to take a message and drop it in a folder, that kind of simple drag and drop manipulation can be done with

HTML 5. That is more application like.
>> It has support for what is called SVG graphics which are more vector-based graphics with the web know, everything is bit maps. When you scale a window up and down, bit maps, if you're stretching it, it gets to look kind of crappy. With vector-based graphics, you get crisp resolution no matter what size you go to.
>> Another thing that HTML 5 is that has semantic tags. These are tags that make sense, things like header, footer, section, article, now if for navigation, those are semantic type tags, video is also a semantic type tag in sure some of the other type of tags that were in the older versions of HTML.
>> So, I talked about, there is new API, scripting API, there is this canvas tag that I had talked about before. You can specify a portion of your screen is a this is your canvas. You can do to the drawings. Maybe you have got a chart or pie chart that represents some data.
>> You define a portion of your screen is the canvas and all of a sudden you get all of this cool animation and whatnot that can occur on that.
>> There are also times media playback so you can synchronize some events as far as the video playing or audio playing and so forth. There is document editing, you can actually edit documents within the HTML 5 construct.
>> Again, what type of things like that, you're really talking more about application and that again is what it is all about, creating an application it is HTML framework.
>> There's also a couple of extensions that are not technically a part of HTML 5 but a lot of browser manufactures and whatnot are kind of treating these like HTML 5.
>> These things are either in there now on some of the browsers or are coming shortly. Some of those things come a fairly excited like geolocation am I think that is one of the things that I think is an advantage for a native app. Native app has a geolocation capabilities built into it.
>> With these extensions, it will allow you to do some geolocation, so your device is in JavaScript with your permission of course and can communicate to the backend server as far as where your location is a maybe bring up a map, shows where your friends are located nearby, things like that.
>> Some of these extensions also include the ability to Web storage, file API which means you can save, save files, upload files. There is a Web SQL database, and Index database and API, again all of the things I'm talking about here are more application type of things.
>> I think in the future as we gothe next year or 2 Down the Road, and HTML 5 and some of his extensions become more widespread, we're going to see more and more applications, more web applications take advantage of this and I daresay that folks that use these will realize that they are using a web application, a mobile web application as well as -- as opposed to a native application.
>> Delroy, how's that? Does that sound good as far as an HTML 5 introduction, explanation?
>> That sounds great, Neil. Mike, can you jump in here and explain to Neil why he is wrong to think HTML 5 is going to change the landscape?
>> Well it is not the savior, least not yet, not by research of the imagination. Granted it does have some benefits over the current actual standard on the web. He calls of video and audio tax, I would love to see those used and adopted more.
>> One problem that we have with that is that there is no real standard of video, audio codec that has been decided upon just yet. You have gods apple and Microsoft in one camp and Google and Firefox in a completely other camp.
>> As for stream them as applications, there is one barrier here. The users are having to learn a new behavior. That is actually the process of downloading these web apps, or I'm sure what most nontechies would consider websites to their phone. Then you have that the question, go to security can, what about the data that is actually being saved? How secure is this? Is the data encrypted in any format, these applications sandbox or is the data sandbox? We have questions or as well.
>> Semantic tags are great for the web in general but I'm not sure that I see the applicability in a mobile context. One thing he did mention, there is greater support for geolocation, but one of the things that people are using their mobile phones nowadays, more and more, is to interact with their environment.
>> One great example is the camera. People are using their mobile phones nowadays as their primary camera. You look at Flickr and the model or the camera that is most popular in terms of uploads is actually a mobile phone.

>> The HTML 5 really does not help in that context is also the lack of accelerometer support. One other issue that I would have with going lock, stock and Burrell to the mobile web, with HTML 5, is how to update the apps on the users and devices?
>> One of the nice things about the native app environment platform is that users can get the updates to the apps on their phone quite easily. And with minimal effort. How would a user be notified that an update is required for their downloaded mobile app and how would a user actually updated?
>> Actually, my, users usually does not have to come and have an icon on their phone desktop screen, they click on it and they are connected right to a web server that has the most updated to version. We need to just push out updates to one location, or web server and then the next time the user clicks on the icon to launch the application, they have got it. There is no difficult issue of how you distribute applications out there and keep them up to date and whatnot like there is with the native app.
>> Touch&é;, my friend.
>> [Laughter].
>> I have got him speechless. [Laughter]. Does this mean I won?
>> I am sorry, Mike. Go ahead.
>> No, I'm just chuckling.
>> We have been going back and forth I think we have found by now that there is no savior coming along to make this easier. This is something that everybody is going to have to choose to do on their own when they move into this arena. Why do new -- why don't we start out with Mike and let folks know when designing for native application, what is the best way to get started, somebody starting from scratch, what is the best way to go about?
>> Sure, there is some basic steps. First step is to commit to a great app. I know this may seem like a really simple, simplistic type of thing to say, may seem fairly obvious to some people. But really, there is a lot out there, web, native etc. that is just thrown out because it was just good enough.
>> Creating a nap for the sake of have a -- having an app is not enough. Good enough is not good enough anymore. There's too much of that. Two standout, what you really need to do is do great work and you need to commit yourself to really be storing the adage of good enough for government work. To do that through action that the public can -- and you can create a great app rather than just an okay or good or mediocre app.
>> The next up is a really important one and that is to develop the user story, to understand the user, the motivation for using this app, why they would use it, and how they would benefit.
>> It really just asking the essential questions behind storage. Ask who, what, when, where, and how. Those are the basic questions. The answers to not even have to be literal. They can be fairly abstract.
>> For example, the answer to where can be walking down the street. When could be, again and apps tracked concept.
>> What you want to do, got to take those answers, next slide please, and turned it into a coherent elevator pitch. An elevator pitch is the page where, if you're walking into an elevator and you have to sell your boss on something, you have maybe 30 seconds before the elevator reaches that for were he is getting off. You have got 30 seconds to make your case.
>> You want to get -- you want to take your answers, turn it into computer elevator pitch is basically the user story. If you're other pitch is not clear or it fails to sell your idea, really what you want to do is either start over and look at those questions again or just stop altogether.
>> Maybe you do not have a good case for a native app. We will just use the one example was a story that we developed to support a native app that we developed here at the Department of Labor. That was the DL timesheet app that we worked with, closely with the HR division on.
>> Is look at 517. In this particular story, we have Jane. She's been working a lot of hours and chose to be sure that she's getting the credit she deserves. She uses an iPhone. At work, she enters her start, stop time and including breaks protected of the week she can see where she may be entitled to overtime pay. If she wants she can forward a report of herself as collected hours.
>> That is the users a story that explains what the app would do, why this person would use it and how it would benefit them.

>> From this, we can develop the requirements that the developers could use to build the app.
>> The next step would be to learn the mobile design the best practices and in UI guidelines. I've put up here, and image from the book For the common designing great iPhone apps. Do not get thrown by the word iPhone. Many of the best practices in this book really expand platforms.
>> Anybody who has seen this book probably can recall the rule of thumb. That is not something that is specific to iPhones, it is something that would apply to anybody using a touch-based phone, and not even, with not even just benefit people using native apps that I believe also if you are developing a web app.
>> I must add, read, learn, no, live and love the UI guidelines for the individual platforms. Android has a couple of pages, Microsoft has their own, Apple has their own interface guidelines.
>> The plat form vendors put those there for the users of benefits. Yes, there are constraints that developers must, in the mid-the designers must learn to live with, but is often said that great artists embrace constraints.
>> What do not exist they impose their own. That is one great creativity has a chance to force.
>> Affords a step, if I may, based on the user story decide whether you're going to go native Web or even both.
>> Thank you Mike, now, Neil could you let people know what are the best ways to get started when they are doing native Web development?
>> Sure, for could switch over to my presentation, I think slide to has, going mobile or the approach.
>> Our approach is fairly straightforward and logical. You can say that it is universal to many kinds of applications.
>> The first thing to do is really identify the high value content and information that users want. What we mean by that, everybody so excited about mobile these days and the technology. Oh, we have have a mobile lab or this group wants one, or this agency wants one and a hurry and Russian app out there thinking that there is this time to market constraint that they could put anything out there and they can declare victory.
>> That is really not the case. You have got to identify what is it that the users want? What is it to your agency or your organization does that the public, in this example, really want? You can take a look at TSA, the my TSA application, we have got a lot of experience as far as the types of things that the public have asked us.

>> Things like, they won't basically the public wants to get through the security checkpoints with the least amount of stress and hassle. That is the overall objective.
>> They have questions about what are TSA policies regarding issues, or liquidate, liquids and gels, what sorts of things can I bring to a security checkpoint versus what do I have to check and put in my checked baggage and things like that.
>> The first of is the most important step and that is really thinking through content, what do people want from your organization? What do they want to see?
>> Once you have gone through that step, there is a subset of that content that may be the from a mobile device. For example, if your audience, your users to a great deal of planning and have to do a lot of planning, planning can be done and generally is done by the desktop computer or laptop type computer where you are thinking things through and making plans and so forth.
>> There are some things that lend themselves more to a mobile device. For example, say you're traveling and you are visiting friends and they give you a bottle of wine, really nice bottle of wine. Well, wait a minute, can I take a bottle of wine on the plane with me? It has alcohol in it in a close level. One of the rules of that?
>> Those types of questions to run themselves more towards, hey, I'm not at my desktop computer, I cannot do this time -- this kind of planning, I need that type of information.
>> There is some information that lends itself to the mobile device. Not all of the information that you have on your public website is appropriate for your mobile website because when people are up and about, they are not going to be wanting to look at some dry statistics from eight years ago about such and such a thing.
>> That point is important, think about the device itself. And number three, as

devilish -- and resources. What did it do you have available, what devices do you want to target, do want to just target everyone with a mobile web solution or do you want to consider, as Mike is arguing for, something that is more specific to a device, like an Apple iPhone and what are the reasons for that?
>> Then, once you identify what the resources you have available, that may be like data, funds, I have this pot of money to do that, what can we do?
>> Number four, this is more a good project management. You need to build a project plan with timelines, rules and responsibilities of who's going to do what. So before you really start your project, you need a good project plan. It is very important to have that so that you can manage well and so the bigger ship knows what you are doing, and went to expected and what are the activities that take place.
>> Finally, updating the approval and agreements. In some cases, you may have to have legal agreements. If you're going to be placing an application on an app store, are you legally able to do that? Do you have to higher developers, people of some skill sets, there's of documents related to that.
>> You want to get these things answer really before you start your project because you do not want to have to start and stop and this time and this credibility because you have not done a great planning.
>> That is kind of an overall approach that we take as far as the launching into a mobile project. Delroy Thank you very much. Mike, can you go into what are the apps stores, where the imported and how to best use them?
>> Sure, we ask stores are the infrastructure that the different platform vendors put in place to make it easier for people to download apps to their mobile devices.
>> I'm the ILS I, you have one, that is the iTunes app store. Blackberry has the app world, Microsoft has their own and android has the android work it.
>> -- Market.
>> There also other apps stores and marketplaces, famously, Amazon has their own. One thing that I would suggest, if you could bring up slide 12, is that when you do enter into these agreements, that you do so at a level, so in our case, we have got the department of labor has several agencies. We went into a make him -- Department a level.
>> The benefit that were sitting there, is that any agency within the department of labor, whether as OSHA, the employment training administration, etc., -- if they release an app, goes to the Department of Labor a count and what you see here circled on the screen is the built-in cross promotion that you get by going through this one account.
>> You can see more iPhone apps by the Department of labor. You can easily cross promote apps within your department or agency is centralizing the account on that particular app store marketplace.
>> The benefit to do so this, what was so centralizing this, in some cases getting the legal agreements, I will say this and I will say from it experience, can be very painful. One of the selling points of has really made it easy for the other agencies within the department to get onboard with this centralization, that control the apps store accounts is that they do not have to go through this. It has already been done, somebody has already weathered the storm.
>> Thank you very much, Mike. Now could you go into what is an API and how to use them when developing an app.
>> So, API. We can go to slide at 21. I pulled this definition from PC mag because I did not want to have, do not want to give a definition in necessarily cater to the technical long list of that this one had reduced the unnecessary amount of jargon to the point where it is actually kind of meaningful.
>> Basically, an API is a language and message format used by an application to communicate with the operating system or some other program.
>> So, android has them. I owe us has them. The API in IOS is built-in, called cocoa touch. Other platforms have them. They are calls that you can make within your application code to the operating system, for example, this is a bring up the camera, saved as a file.
>> You're interacting with built-in components that you do not have to build yourself. APIs are also used to access data, not just system resources. One nice thing is federal agencies have them. A benefit to these type of APIs, is one, you have a wide data. Johnny to download data sets are data files to update the data in your app.
>> Does looking at some of the data sets that the Department of Labor provides even

within the employment and training, some of those it is are in excess of 800 MB in size. Nobody's going to want to download an app, 100 makes in size especially if it has really specific and limited functionality.
>> Which was built to do is reach out to API and get the data life. Sometimes these data states -- sets are updated. The second benefit is that you automatically and always have up to date data in your apps as long as the API is up to date.
>> A third benefit is you can have control access to the data. You can expose your agency data through centralized warehouse for lack of better term, with -- this is one thing that we are really seeing the fruits of. Within a couple of weeks, we will be making the data from the Bureau of Labor Statistics available through our API.
>> -- Is where protective of their data for very good reason. What they are able to do is provide our API and only our API very specific access to very specific data. In doing so, and providing the data to our API, we are able to make that data available to everyone. Everyone is able to access the OS and data a while BeOS is still able to maintain a real tight lock on security.
>> This is an example, was up on the script there, is the developer community site where we have our API, we have several data sets available and we have also made available a software developer kits, and SDK for several platforms to make it easier for developers to get started. You do not have to be experienced or inexperienced developer to even get started in making an app using our data.
>> You could be someone who just picked up a book yesterday and learning objectives here and learning to program and android and you have a great idea. Within a weekend, you can get started and you are you using our data.
>> Thank you Mike Turk now I would like each of you to quickly, we're running out of time, tell us about your app and the website and also go into whether you have ever done one of the other types from what you're arguing. Neil you have our are doing for websites, have you ever done a native app and Mike --
>> [Indiscernible - multiple speakers].
>> I would like to jump in quickly here. I would like to ask a little more pointed question which is, in 20 or 30 seconds or less, would either of you ever do the other type of tool? Meal, which you do mobile apps, Mike would you do mobile website?
>> Well, I have to confess, I went and we have done a native app as well as the mobile web. Even though I have been arguing very strenuously for mobile web, we are a big believer as well as native application. They actually have their place in there. I think the answer is all of the above.
>> Like?
>> Same. I have to make the same confession as well. We have been making a big push into it is apps, but we have also got a major effort underway to make sure that all of our web apps are compatible and easily usable on mobile devices.
>> I knew it. I knew there was no easy answers here. You guys have given us a hard time this whole time. Gosh. So we're pulling your leg.
>> We are right at the noon timeand want to thank Delroy for doing a great job of moderating this program. I really want to thank both Mike and Neil . It is really hard sometimes to take extreme positions because this is a nuanced area and I really appreciate their coming up and laying out the issues very clearly on both sides.
>> If you listen to their arguments, the arguments are true, there are good uses for both mobile and, mobile web and made of apps. Those of you who are looking for a silver bullet today, sorry there is not going to be one.
>> In closing, wanted to let you know that a lot of folks did mention, they did have some questions that we did not actually get to. What we're going to do, we are going to go ahead and compile those questions and we're going to do a blog post on our mobile gov blog that will address those. Those folks -- we will send you an e-mail link to the blog posts word answers more of these questions as well as the wealthy posting informationin the audio and video of it as a program.
>> I wanted to thank Delroy forholdingg, being a good referee and keeping the guys from killing each other, I wanted to especially thank both Mike and Neil for in helping us to see that this is actually more complex, and it always comes down to audience, mission and what you're trying to accomplish.
>> Thank you.
>> Thank you one.
>> There will be an evaluation that will be sent to you should give everybody all fights -- five.

>> Thank you.
>> [Event Concluded]