Finding solutions as growth and fragmentation complicate mobile device support.**BY MACHE CREEGER**

ACM CTO Roundtable on Mobile Devices in the Enterprise

BLACKBERRY? IPHONE? ANDROID? Thin or fat client? Carrier network or Wi-Fi? Developers of mobile applications have many variables to consider if they are going to be successful in a rapidly changing and increasingly fragmentary environment.

With rapid worldwide growth and increasingly

diverse devices and networks, supporting mobile devices in the enterprise is becoming increasingly more challenging and complex.

Application service architectures, security, connectivity, testing, a constantly changing mix of devices and platforms, and an uncertain future are among the concerns mobile application developers must face in deploying mobile device services. Change in this market is the only certainty, and developers must continually look ahead to refine development and deployment strategies to keep up.

In this ACM CTO Roundtable, five

leaders in the mobile applications field discuss the current challenges in supporting multiple devices on multiple networks for highly variable business requirements.

—Mache Creeger

Participants

Andrew Toy is past VP, mobile applications at a major Wall Street investment bank; past VP, mobile and syndication technology at MTV Networks; cofounder and CEO of Enterproid.

André Charland is the developer of PhoneGap; and cofounder and CEO of Nitobi.

Moderator Mache Creeger

George Neville-Neil is a past member of the Paranoids group at Yahoo!; and principal of Neville-Neil Consulting.

Carol Realini is past CEO of Chordiant; founder and CEO of Obopay.

Steve Bourne is CTO, El Dorado Ventures; past president of ACM, chair of ACM *Queue* Editorial Board, and chair of ACM Practitioner Board.

Mache Creeger (Moderator) is Principal, Emergent Technology Associates.

CREEGER: Andrew, when you were responsible for the use of mobile devices at a major financial institution years ago, what were the biggest concerns?

TOY: We focused on the BlackBerry. The two major problems we had were our inability to customize services and maintaining control of service reliability. The BlackBerry presented itself as a closed system; the NOC (Network Operations Center), the services, and the server software were all controlled by RIM (Research in Motion). There were very few APIs to work with and because of its proprietary nature, we had a limited understanding of its underlying architecture. As a result, when something broke it was hard to fix. Theoretically it was secure and RIM could talk about why that was true, but the same reasons that made it hard to penetrate made it dif-

ficult and expensive to maintain as a mission-critical platform. We always worried about losing email, with our only recourse being to call RIM and demand it be fixed.

While there are lot more device platforms choices today, if you look at operating systems with enterprise capabilities, the only real viable candidates are Apple iOS and, to a lesser extent, Google's Android.

Given these options, enterprise customers now believe they need to support more than just the BlackBerry. However, they are unsure how to go from the RIM world to this new and very different place. In the RIM environment everything is done for you. When you take things into your own hands, you recognize there are a lot of issues that the BlackBerry solution used to address that are now your problem.

NEVILLE-NEIL: What about compliance issues? Suddenly, you've put a huge amount of data that's probably controlled by compliance rules in the hands of people who are wandering around with their devices.

TOY: We found that lawyers can advise on industry-specific compliance requirements, but in finance everything does not necessarily have hard and fast mandated technical standards. Our experience was more along

the lines of "you have to protect your customers."

We focused on such things as avoiding client data loss that triggered financial industry-specific mandated actions. Data loss required notifying each client of the breach and potential access by anyone, including a competitor. The loss of a mobile device meant the regulatory notification requirement would be triggered if data security was not provable to some level of technical certainty. Being able to make that guarantee drove us to ensure that proper screen locks and encryption were placed on mobile devices.

It is important to create a culture that does not view the security guy as the enemy. Security should enable things otherwise not possible. If a company wants to enable financial transfers, then you need security, because without it the business will collapse under fraud and real-world attacks. Security is not a goal but a means to deliver business value and manage risk in sustainable ways.

REALINI: My company is about delivering consumer-facing functionality over mobile devices, and we have payment and banking services at the back end. We deliver that functionality in the U.S., as well as India and Africa. Those environments are diverse—a lot of dumb phones, a lot of smart-

STEVE BOURNE

I don't see how existing wireless carriers in the U.S. will be able to make the required capital investment to handle the increased demand for services. Certainly that is the case in the next year or two.

phones—with the mix depending on where you are in the world. The transports are also diverse. Some countries have a lot of data access. Other places, data services just aren't available.

I have worked with mainframes, client server, and Internet-based computing. Mobile is the hardest kind of computing I've experienced because it is a fragmented and rapidly changing device market. You have at least 18 platforms or operating systems, and they're in constant flux.

IT organizations may want to build that mobile expertise in-house, but that's not an effective strategy as the mobile device market is moving way too fast. Growing mobile expertise organically is hard and chances are your company would make too many rookie mistakes. Either hire or outsource to contractors with the expertise and get it right the first time.

As an IT manager, you should ask, "How fast do I need to move?" "How many platforms do I need to support?" and "In how many geographies do I need to operate?" It is important to understand that this is not just another operating system. It is a rapidly moving environment, and it's just going to change faster and get more fragmented over time.

NEVILLE-NEIL: In embedded computing, every product is different and

every customer is different. If someone powerful in the company buys an iPhone or Android, he or she then drives the IT department to support it. It's a very customer-driven model.

CREEGER: How do the IT folks avoid being buffeted by everybody coming at them at once?

REALINI: You just get used to it. If you think the world is all about iPhone and Android, just blink and it will be something else. It's going to be a fragmented environment, and it will depend on your application. If you are a large financial services firm, then you may be able to dictate that everybody use BlackBerries. You don't have that luxury if you're doing consumer-facing applications. Every current and future device is part of my world, and we must have strategies to leverage those devices even as the mix of those devices changes constantly.

An interesting question to ask is, "What chance does Android have of becoming the universal operating platform for mobile—the mobile equivalent of Microsoft on the desktop?"

CHARLAND: I don't think you can ever make that assumption about any platform. A year ago, I would have said iPhone would be the universal operating platform for mobile. Today it seems like Android, but things are

ANDRÉ CHARLAND

I want to stress the minimum viable product approach: What value can we provide to our user base and can we do this in the mobile browser?

moving too fast to say that something will not change in the future. There might be a dominant player like Android, but you're never going to be able to discount iPhone or BlackBerry.

CREEGER: How do folks decide on appropriate application architectures and the smartphone platforms they will support?

CHARLAND: I would focus on two things: the minimum viable product you can deliver to your users and what platforms are needed to support them. A lot of people look at their Web stats and assume that because people visit the Web site with an iPhone, iPhone should be the first supported platform. To prioritize a list of supported platforms you must perform basic research on your user base: poll your users, look at market trends, and do your best to forecast what phones your users will be buying.

REALINI: Three years ago India had 150 million phones, now it's 700 million. Moreover, the features of these phones are changing quickly. You could do research and then extrapolate, but you have to work quickly and constantly adjust to what's really happening in the market. It is almost like trying to track fashion or pop music.

How do people plan in such a fast-changing environment? They have to ask themselves two questions, and do

so often because the answers change as the market changes: Do I want a thick or thin client? Which devices am I going to support? Once you answer those questions and have a strategy in place, you better ask those questions again every 6 to 12 months.

TOY: The key underlying issue is this: Are you buying people their phones or are you trying to support the phones people purchase and bring into your environment? If you mandate what is used, then you can have better control.

BOURNE: Can you really mandate in a modern enterprise, even in a large regulated financial services company?

TOY: Because of the tight regulations in financial services, most certainly. For a multimedia business, I'd say no.

NEVILLE-NEIL: Small businesses are in the most trouble because they're the least likely to be providing their employees with smartphones.

CREEGER: Is virtualization going to be a solution?

TOY: Multiple use-case profiles are the way to solve the multiple-mission problem. Is virtualization the best approach? It is difficult to do power management with a hypervisor on a device with more than one operating system. This doesn't mean that it's impossible or not viable, just that it's extremely challenging.

Right now a mobile operating sys-

tem manages power and does a lot of things under the covers to maximize battery life. Take away the operating system's direct link to the hardware, and you lose the ability to effectively manage battery life. That is a huge blow to the value of the phone.

While you could migrate power management (or the management of any limited resource) of a mobile phone operating system to a hypervisor, you would then be stretching the definition of the traditional hypervisor to something more like an operating system of operating systems—effectively, a very fat hypervisor.

NEVILLE-NEIL: It will probably not happen near term, but it might happen on Android because it has gigahertz phones. Apple will never let a hypervisor execute on an iPhone.

TOY: A sufficient solution might be more like the Unix method of multiple users. You would have one box with multiple users logged in. Each user has his or her own experience; all users run concurrently; and there is one kernel and one operating system.

REALINI: One of the biggest trends in emerging markets is that users have multiple SIM (subscriber identity module) chips in their pockets to optimize the costs of their calls. Carriers have different pricing to different destinations, and users pick the chip

that minimizes cost for a specific call. Effectively, they are creating multiple profiles similar to what has been discussed, but instead of maximizing security, it's minimizing charges. In India if a phone does not have dual SIM chip modes to allow the user to change personalities, it will not sell.

Mobile phones are personal—somebody calls me and I know it's for me. We all have multiple personas: businessperson, mother...A personal device must evolve to support these multiple roles.

TOY: In the business world, some of those personas can be very tightly managed. For a company employee, that personality can be made to function under a formal corporate security policy.

NEVILLE-NEIL: You are going to see more of the iPhone architecture in most smartphones—a combination of Jails (<http://www.freebsd.org/doc/handbook/jails.html#JAILS-SYNOPSIS>) and Mac frameworks. These control structures are in Mac OSX and FreeBSD. While I don't believe Android has these functions as yet, and RIM certainly doesn't, smartphones will migrate to this type of approach because virtualization is too heavy and loses control of the lowest layer.

These Apple technologies isolate applications from each other, and

“

GEORGE NEVILLE-NEIL

The Apple architecture, which is a nonsharing design, is the right place to start in developing a next-generation mobile operating system.

”

“

ANDREW TOY

Security is not a goal but a means to deliver business value and manage risk in sustainable ways.

”

all their APIs have the ability to control where information flows. This introduces some problems when one would like to share data but cannot. Those issues notwithstanding, the Apple architecture, which is a nonsharing design, is the right place to start in developing a next-generation mobile operating system.

TOY: When I was building applications for enterprise-owned BlackBerries, we often asked RIM how to access a particular piece of data. RIM would say it was not possible because it was insecure. That mentality seemed paternalistic in that RIM was going to protect us from ourselves by not allowing certain functions to be implemented regardless of the business need.

CREEGER: Carol's experience shows that network connectivity can be very uneven worldwide. Just take a look at Africa. How do you work with issues such as intermittent connections or long latency when they are the rule rather than the exception?

REALINI: It's more widespread than just Africa. We have one U.S.-based carrier that gave us 36 duplicate SMS messages in the past 30 days. If you're deploying applications that use mobile phones, you cannot depend on a rock-solid 24/7 proprietary network even in developed countries.

CHARLAND: We're focusing on ex-

treme applications with extreme security or other situations where you have to deal with really poor phones and poor networks. It is important not to lose sight of the big swath in the middle, especially in North America and Europe, with a reasonable average for phones and networks and relatively low security requirements. IT managers are going to be faced with this type of environment far more frequently, and their challenge is to build out to different device platforms.

We see many clients deploying HTML5 browser-delivered applications for some things and then native installed applications with a wrapper such as PhoneGap. It really depends on the devices they're targeting and their use cases.

NEVILLE-NEIL: Carol, how do you deal with the software management problem? How do you manage versioning?

REALINI: Our approach is either to purchase or build tools to help us be efficient. We figure out how to work with the most devices efficiently and how to create reference ports. You have to target what devices you believe people are going to use and be efficient about doing a reference port because it's not just one device, it's multiple devices.

We invest to get a superior user experience on 80% of the installed base of phones. This is done by an applica-

CAROL REALINI

Mobile is the hardest kind of computing I've experienced because it is a fragmented and rapidly changing device market. You have at least 18 platforms or operating systems, and they're in constant flux.

tion on the phone or on its STK (SIM application toolkit) where the carrier distributes the application as part of its SIM chip.

CREEGER: You have several different ways to develop applications. How do you make those kinds of decisions?

REALINI: You have to look at things early and often because this is a moving target. We use the 80/20 rule, with 80% of the devices providing a good to great user experience and the other 20% providing adequate user experience.

TOY: The key is to have a tiered strategy and not go for the silver bullet. Don't say which device is the right one. While all devices could probably be supported, you have to ask, "What is the right functionality to have on each platform, and what is the minimum functionality required for any device?"

REALINI: The CTO of my company puts things in three buckets:

- ▶ I know it works because I've certified it.
- ▶ I think it works because the device manufacturer said it's totally compatible with earlier implementations.
- ▶ I have no idea if it works because multiple changes have happened.

This is important because your consumers and/or employees have to be able to make a decision from the thousand or so devices that could be pur-

chased. You have to help them identify the 200 or so devices that will probably work well and the 50 or so devices that have been certified.

TOY: With the world changing so fast, you have to make the effort to keep those buckets up to date and revisit your categorizations frequently.

REALINI: There's a cost to making sure things are in the right buckets. We had a specific credit-card application in place when the iPad was launched. At that point, everyone was told that all iPhone applications would work on the iPad. That falls into my second bucket: I think it should work because Apple told me iPhone applications work on iPads. Well, guess what? It didn't work.

That experience taught us that we have to say to our partners, such as the credit card company, that we *think* it works, but if you want to be sure we better go through a three-week certification process.

NEVILLE-NEIL: In targeting one or more platforms for an application set, you should use minimal surface area to maximal effect. Android or iOS has the system-call complexity of a modern workstation operating system—thousands and thousands of possible APIs. When trying to design portable software, use the:

- ▶ Fewest APIs possible—this limits

the complexity of porting the software to new devices.

- Oldest APIs—they have been around long enough to be supported by many different device variants.

- Best tested APIs—they will be the most reliable.

CHARLAND: Ideally, in cross-platform software development projects, we first target BlackBerry, as it is the most minimal platform. We negotiate a minimum operating system release level with the customer, typically pushing for at least 4.6. Currently, BlackBerry is at version 6.0, and if that is acceptable, it makes for a much richer application platform.

We focus on 4.6 because there are still a lot of enterprise users on it. We target what we can, using the browser as an application and build up from there to Android and iPhone. It's important to stick to this philosophy and not start with an iPhone application and try to work back to BlackBerry. That approach often leads to emulating iPhone features on a BlackBerry, at best an extremely painful effort.

NEVILLE-NEIL: Apple does try to make it easy to move things from the Mac desktop environment to iOS, but it's not the same environment and you get a poor user experience. The same thing will happen taking desktop/server Linux developers and putting them on Android.

CREEGER: Smartphones are not the only devices that we're talking about here. Not all mobile-specific devices are necessarily phones, such as iPads. How can you broaden this advice for those kinds of devices?

NEVILLE-NEIL: We have already been through this with the Palm Pilot, and in a lot of ways those lessons have been forgotten. When the Palm Pilot came out IT departments went nuts. A personal handheld device that contained a large proprietary address book and was subject to loss or inadvertent disclosure on an Internet site was not what they wanted to hear about. One should be careful about placing persistent proprietary data on a mobile device.

CHARLAND: I want to stress the minimum viable product approach: What value can we provide to our user base and can we do this in the mobile browser? The browser paradigm is a

familiar concept to IT departments.

TOY: If possible, I will go with a browser-based application, but it is not always a viable choice. The only platform that allows you to keep your application behind your firewall is BlackBerry. Yes, you can run VPN (virtual private network) on the iPhone, but iOS locks up all your other applications. Plus, iPhone will not support two-factor authentication, which is becoming an industry requirement. While I agree that one should look at doing a browser-based application first, aka thin client, it's a challenging approach and will not always work. A lot of the time the juicy stuff you're trying to access with your thin client is on your intranet and behind your firewall. Today only BlackBerry gives you an easy path to get there.

REALINI: A thin client has inherent advantages if the network is powerful enough to support it. Right now in the U.S. we have huge network capacity issues. While you can talk about how thin clients can get by the firewall, there is the issue of whether the network is going to be fast enough to make that model viable.

I think the network problem in the U.S. will be fixed and will eventually be fast enough to handle the demand. So in the future, when everyone has a smartphone and the network's going to be fast enough, why wouldn't we all want thin clients?

NEVILLE-NEIL: You're going to have to battle for control. I want my data on my device and not on someone else's server. It makes perfect sense for sensitive corporate data not to be under my control, but it makes no sense for me not to have control over my own data.

REALINI: So, thin client implies that my data is in the cloud?

NEVILLE-NEIL: Yes.

CREEGER: And that means you're giving your data to Google or other data aggregators.

REALINI: Everyone should care, but I am not sure they will care as much as the technical community.

NEVILLE-NEIL: There are people who care, and more will care as more data compromises happen.

CREEGER: Is anyone pushing a fat-client approach today that focuses on the use of mobile-phone platform cycles?

TOY, NEVILLE-NEIL: iPhone.

REALINI: Do we agree that a thin client has inherent advantages in the right environment?

CREEGER: Today, a thin client is desirable because the cloud is in ascendance and people are not sensitive to data security.

TOY: For the enterprise, personal data privacy is not a problem because it is not your data; it belongs to the company. Enterprise IT guys are going to favor thin client. They want to keep the company's data inside the data center to control access better, including revocation.

BOURNE: As mobile devices become more ubiquitous, I don't see how existing wireless carriers in the U.S. will be able to make the required capital investment to handle the increased demand for services. Certainly that is the case in the next year or two. Cellphone data transport is limited, and in the U.S. at least, carriers are not making great money on those services. How does Wi-Fi as an alternative transport layer fit in?

NEVILLE-NEIL: The urban U.S. usually has good Wi-Fi coverage. Practically all mobile devices have Wi-Fi, and people building applications would be crazy not to take advantage of that.

For the purposes of authentication, cellular phones are attractive as each one has a hard-to-duplicate ID. Plus, there are many things a carrier can do to secure data across a cellphone network that cannot be done with random Wi-Fi access points. Lastly, when you touch a Wi-Fi access point, unless your data is encrypted, everybody else is touching your data as well.

REALINI: If wireless networks don't get better, will we get to the point where smartphones are really just connected Wi-Fi devices?

My iPad is useless as a connected application, and I have stopped using it because it is too slow for some applications. If we have a situation where users have powerful devices but the network is unreliable, they will learn to roam on Wi-Fi in the same way Africans learned to carry two SIM chips.

If that becomes standard practice and carriers don't solve the problem, the cellular network will diminish in importance. People will defect from their networks and start connecting

to Wi-Fi. We'll see a shift from cellular devices to Wi-Fi devices.

CHARLAND: Regardless of whether you are doing browser-based or native mobile applications, you have to design them for a spotty connection. You can't always assume there is network connection, nor should you think that there's never a network connection.

CREEGER: What are the most important issues you would stress to our readers?

REALINI: I would stress: (1) If you don't already have seasoned, in-house, mobile expertise, rent or buy it but don't try to grow it organically. (2) Be prepared to deal with a highly fragmented environment. (3) Do your best to define what you will and will not do. In mobile, one has the opportunity to achieve huge scale if the right things are done on the right devices in the right way. Making a mistake means fragmentation and getting bogged down. (4) Expect dramatic change all the time. Along with fragmentation, mobile is moving at a much faster rate than one sees in IT. (5) As you plan to develop new software, continually ask what the market is going to look like in 6 to 12 months so you know what you are getting into.

CHARLAND: (1) Define the minimum viable product for internal and external customers. (2) Consciously choose which devices you have to support. Don't just say all; do the market research, look at the market trends, talk to customers. (3) Determine the cross-platform user experience; then pick the solution that allows the design to get close to a single application. No application is ever totally cross platform. You will have differences and they should be documented. (4) Determine if the application can function in a Web browser (including HTML5) for the devices being supported—if not today, then in the future (check W3C and other standards bodies). Also, research whether a hybrid approach such as PhoneGap is feasible. (5) Determine your plan to test all the different devices on the different carriers. It is barely good enough to buy all the devices and have one individual who can test on every device. You'll need either to have a more comprehensive testing plan or to hire a third-party testing service.

TOY: (1) Trying to make everybody

happy is an unsolvable problem. One needs to define tiers of service and decide how many tiers will be supported. (2) Define the key issues that are important to your business—functionality, security, and ubiquity are three good concerns to start with. (3) For each of your tiers, define the level of resource devoted toward the support of each issue and the devices you will be supporting at that level. Trying to make every device tier support every device at the maximum level is a recipe for failure. It's fine to say that the CEO must use only a BlackBerry and cannot use an iPad to access important documents. It is also fine to say that someone further down the security stack can just synchronize an iPhone. Applications must fit into a dimension of that tiering, and perhaps people lower on the security stack just don't get certain applications, or any applications.

CREEGER: How do you suggest IT manage and track information service consumption and the threat environment?

TOY: One way is to go thin and keep information assets behind a firewall in much the same way folks have done with thin desktops. Alternatively, you can emulate what folks have done with laptops and install end-point security products to impose control directly on the device.

NEVILLE-NEIL: You have to think about what data is important to your business and its continuity. You need a disaster recovery plan that is sensitive to different types of disasters. You have to decide which data goes where, on which device, and to which people. Most computer security is trying to decide those questions, and the same will be true in mobile as well. Also, use the oldest, most established, and minimal API set possible. It will just make your life much easier in the end.

CREEGER: Does anybody have any idea what the world is going to look like in two to three years?

TOY: Tablets are the biggest workplace changer in the next two to three years inside a company and on the Internet.

CHARLAND: While I don't think native applications will ever go away completely, from a developer's perspective, the majority of applications

will be Web-based. Tablets will play a bigger role, but they will blend more with laptops, and I feel the phone will always play a bigger role than the tablet.

NEVILLE-NEIL: We're going to see more splitting of the network space. More people will have personal area networks, accessing a MiFi, their phones on the cellular network, whatever. You're going to see devices talking to each other a lot more.

Applications will move from the phone to the tablet. The tablet will become the primary consumption device for media and the sweet spot for consumers. I think kids will lead the way.

In the corporate space, you won't see consolidation around Android or iOS, and both will maintain a varying percentage of the marketplace unless or until somebody produces a new game-changing killer device.

Lastly, we'll have a lot more thin client in the enterprise space. It's just an easier way to control access to data.

REALINI: Today, companies interact with customers primarily in person or on the Web. In the future, mobile is going to be the most important way those interactions take place. Smartphones will become richer and more powerful, because we're going to expect and demand it. It's only natural that a lot of customer-facing applications will be mobile. I think mobile is going to fundamentally change the types of services that can be delivered; how efficiently those services can be provided; and what types of customers can be engaged. I think mobile will create vast new markets to broaden the reach of commerce way beyond its traditional scope.

Related articles on queue.acm.org

Mobile Media: Making It a Reality

Fred Kitson

<http://queue.acm.org/detail.cfm?id=1066066>

Four Billion Little Brothers?: Privacy, mobile phones, and ubiquitous data collection

Katie Shilton

<http://queue.acm.org/detail.cfm?id=1597790>

Mobile Application Development: Web vs. Native

Andre Charland, Brian LeRoux

<http://queue.acm.org/detail.cfm?id=1968203>

© 2011 ACM 0001-0782/11/09 \$10.00