



Cloud Computing for the Battlefield

featuring Grace Lewis interviewed by Heidi Brayer

Heidi Brayer: Welcome to the [SEI podcast series](#), a production of the Carnegie Mellon Software Engineering Institute. The SEI is a federally funded research and development center at Carnegie Mellon University in Pittsburgh Pennsylvania. To learn more about the SEI please visit our website at sei.cmu.edu. A transcript of today's podcast will be posted on the SEI website. My name is Heidi Brayer and today I'm pleased to introduce you to [Grace Lewis](#), a senior member of the technical staff in the [Research, Technology, and System Solutions initiative](#) at the SEI. Grace's recent work includes architecture and design of service-oriented systems, guidelines for engineering systems of systems, testing and compliance in SOA environments, service-oriented architecture, governance, and cloud computing. In today's podcast, Grace will be discussing software that her team is working on for soldiers who carry handheld devices to support their mission needs while they are in the field. Welcome, Grace.

Grace Lewis: Thanks, Heidi.

Heidi: Let's start off by having you tell us about some of the challenges that soldiers face in using handheld mobile computing devices to support their mission needs in the battlefield.

Grace: There are mainly three challenges. The first challenge is that no matter how powerful mobile devices are these days, they offer less computation of power than a conventional desktop or a server computer. The second challenge is that there are computation-intensive tasks—for example image recognition, doing maps, using GPS—that really take a heavy toll on battery power so battery consumption is a big problem in mobile devices. The third challenge is that soldiers in the field in tactical environments have to deal with unreliable networks and bandwidth.

Heidi: So tell me a little bit about the exploratory research that your team is working on to help address some of these challenges, specifically your use of cloudlets.



Grace: So the easiest way to think about cloudlets, which is the main component of what we're doing, is as a very small private cloud. In essence cloudlets are localized, lightweight servers, very lightweight, that are running one or more virtual machines. The idea is that soldiers can offload expensive computations from their handheld mobile devices onto these virtual machines, and what happens is that the computational-intensive task that would normally execute on the mobile device are now executed on the cloudlet and helping to conserve battery power.

One of the advantages of cloudlets is if you look at other work in the field of offloading computation to conserve battery power in mobile devices, which is a technique commonly known as cyber foraging, what you see is that this work relies on either conventional environments or environments that really create a very tight coupling between the applications that are running on the handheld devices and the servers onto which computations are offloaded. Because we are using VM technology or virtual machine technology, the cloudlets really decouple the mobile applications from the servers. Because of the use of VM technology, we really provide greater flexibility in the type and platform of applications; and this also reduces set-up and administration time, which is really critical to systems at the tactical edge.

When I mentioned that cloudlets were localized servers, what I meant was that cloudlets are located within proximity of the handheld devices that use them. What that means is that latency is decreased because we're using a single hot network and potentially lowering battery consumption by using Wi-Fi instead of broadband wireless that consumes much more energy. So you can think about, for example, a cloudlet running a tactical operational center which is called a TOC or inside a Humvee, and the soldiers with their mobile devices interacting with the cloudlet on either the top or the Humvee. Now from the security perspective, which is a big concern in these environments, because the cloudlets use Wi-Fi networks, you can take advantage of all the securing infrastructure that already exists; so you can take advantage of existing security policies, or you can establish rules such as that the cloudlet can only be accessed from certain particular handheld devices, or you can use whatever encryption techniques are used within your organization.

Heidi: Now, as I understand it, this also relates to your work on conserving battery power in mobile devices, specially your use of dynamic VM synthesis. Can you please tell us more about this?

Grace: Sure, each mobile app has two components: It has a client portion, and it has something called the application overlay that is almost like the server part of the application. What is inside this overlay is the computation-intensive code, so what happens when a soldier executes the application on the mobile device, the overlay is sent at run time to the cloudlet and it's applied to one of the virtual machines that are running in the cloudlet. This is what is called dynamic VM synthesis, the integration of the overlay with the virtual machine that is running.



Now, the application overlay is pre-generated by calculating the difference between a base VN, which is a clean VM with nothing on it and the base VM with the computation intensive code installed so in essence the only cup link that really exists between the mobile app and the cloudlet is that you have to use the same version of the virtual machine both to generate the client part as well as the overlay.

Heidi: And as you conduct this work, are there specific concerns that you have given that you are working in a tactical environment?

Grace: Oh absolutely. You can refer to a tactical environment as pretty hostile. It's characterized by unreliable networks and bandwidth. You can lose cyberforaging platforms because of explosions, and you also need a rapid deployment because you have soldiers that are running mission critical applications and have to provide very quick responses to what they're doing. For example, imagine this scenario: You have a soldier that needs to execute a computation-intensive app. This app is configured to work with cloudlets so what happens at runtime is that the app discovers a nearby cloudlet located say in a Humvee and offloads the computation intensive portion of the code to it which is the overlay. Due to enemy attacks, network conductivity, maybe an explosion, let's say the mobile application run by the soldier loses contact with the cloudlet. What happens then is that at runtime the mobile app can detect another cloudlet that is in the environment and, because of dynamic synthesis, you can have the app running in a short amount of time because the only thing you need to do to get the app running again is to do dynamic VM synthesis. You don't have to do any special configuration on the app or on the cloudlet. So definitely having this flexibility makes it very opportunistic because now you can respond very quickly to problems in the field.

Heidi: Now as I understand it, part of your research involves the creation of an android app for face recognition applications. Can you explain a little bit about how that will work?

Grace: Sure, the application that we are building as a proof of concept for our work is a face recognition app. Basically what the face recognition app does is first it locates a cloudlet via a discovery protocol that we have implemented. The second thing it does is that it sends the application overlay to the cloudlet where dynamic VM synthesis is performed. Then what happens is the application simply starts executing; it captures images, and it sends them to the facial recognition server code that is now residing in the cloudlet so the computation intensive code is executing in the cloudlet and not on a mobile device. The application overlay itself is a very simple facial recognition server. It is written in C++ and what it does is that it processes images from a client either for training or for recognition purposes. If it's running in recognition mode, what it does is it returns coordinates for the faces that it recognizes as well as a measure for confidence.



For example, I think that this is Heidi, and the confidence that I have is 95 percent. The first version of the cloudlet that we have created is a very simple HTTP server. and what it does is it receives the application overlay from the client, decrypts it, decompresses it, and does VM synthesis to dynamically set up the cloudlet.

Heidi: So what is the status of your research so far? Are you collaborating with anyone and what stage are you at, and what will the end result be?

Grace: As part of our research we are collaborating with [Mahadev Satyanarayanan](#) who we call “Satya,” the creator of the cloudlet concept. He is a faculty member at [Carnegie Mellon University \(School of Computer Science\)](#). The first phase of our work what we had been doing is building the cloudlet prototype that I just described. Now we're on our second phase where we are conducting measurements to see if these computations in the cloudlet really provide significant reduction in battery power consumption. Also what we want to gather is measurement-related bandwidth consumption that globally transfers as well as VM synthesis because we want to make sure that it is truly an optimized solution. So finally if we're successful, our third phase is that we will create a real cloudlet, and we'll have it running in our concept lab, and that will serve as some form of baseline to explore other ways to take our computation to the tactical edge.

Heidi: Grace, thank you so much for joining us today. This recording and a downloadable transcript are available at sei.cmu.edu/podcasts. If you have any questions, please email us at info@sei.cmu.edu.