4. Discussion

4.1 Conclusion

In this thesis we described two frameworks for distributed smartphone computing, one for applications with compute intensive tasks and another one for applications that take contextual sensor information into account. Both frameworks provide a common structure for the development of distributed smartphone applications, thereby extending the possible distribution model options for distributed smartphone applications. Both frameworks have in common that they put the smartphone application at the center and only make use of other resources when applicable. This contrasts with the traditional approach for smartphone computing in the above areas, where a centralized web server is part of the distributed environment. The frameworks that we described do not need such a centralized component.

In Chapter 2 we described the Cuckoo computation offloading framework. Computation offloading can be used to transfer compute intensive tasks to other resources. Transferring tasks away from the mobile device has the potential advantage to save on energy usage and/or computation time. Also the other resource might be capable of performing the task better (i.e. with a higher quality).

From two case studies, eyeDentify and HDR we derived the requirements for the Cuckoo framework. We found that the framework should have a fall-back implementation that can be executed on the device itself in cases of disconnection, and also that the framework should allow this local implementation to be different from the remote implementation. Furthermore, we require the framework to make smart decisions, based on the current context of the device, the history of previous executions, and the parameters of the current execution.

The resulting framework is created for the Android framework and exploits Android’s activity/service inter process communication, by intercepting calls to a service at runtime. These calls pass through Cuckoo’s Oracle which decides based on the above requirements whether or not to offload. At build time Cuckoo uses code generation and code rewriting and integrates into the default Android build process of the recommended Android IDE, Eclipse, through a plug in. Transforming an application into a computation offloading application is simplified to a single mouse click followed by creating an implementation for remote resources. The latter can be as simple as copy-pasting the local implementation. In addition Cuckoo offers developers the means to fine tune the offloading decision process, by offering methods for method weights, return size prediction and energy estimation.

Not only does the framework help creating and running the application on the mobile device, it also comes with an offloading server that can be started on virtually any remote resource. To help users manage their remote resources and to enable Cuckoo’s runtime to discover such resources, Cuckoo also has a separate resource manager application.

Evaluation of Cuckoo showed that the build tools drastically simplify the creation of computation offloading applications and that the runtime overhead is low enough,
such that computation offloading indeed can lead to less energy usage or lower execution times. The components that together form the input on which Cuckoo's Oracle makes its decision prove to be good enough to lead to high quality decisions.

In Chapter 3 we discuss the SWAN framework. SWAN simplifies the development of applications that use input from sensors. Application developers can use SWAN's domain specific language SWAN-Song to create complex context expressions that in turn can be registered to the SWAN Evaluation Engine. This Evaluation Engine takes care of efficiently evaluating all the expressions of all applications on a device that use sensor data. It informs the applications that registered the expressions only when the expression value changes. The application itself is then responsible of acting upon such a change.

SWAN supports a wide variety of sensors, from the on device sensors such as the accelerometer, to external sensors over Bluetooth, to network sensors that get their data from the web, to data sensors that inspect data sources on the device, such as the calendar. It is possible for third-party applications to add sensors to the framework. These sensors in turn can be used by yet other applications. The SWAN framework also has a sensor maker tool, which can, based on an JSON input file, generate a large portion of code needed for a new sensor.

Furthermore, the framework has built in user interface components for configuration of sensors and expressions, which can be reused by third party app developers.

We constructed a sub framework, especially targeted at network sensors that exploits the technique of communication offloading and further structures the development of those sensors, and in addition also makes these sensors much less communication intensive at runtime by offloading polling to a remote resource. This remote resource will notify the device once an update has been detected through a push message.

To broaden what can be expressed in the context expression abstraction we expanded the SWAN-Song language to add a location to each sensor predicate. This resulted in cross device expressions, expressions that take not only the local context, but also context on other devices into account. We do not forward all sensor data to a central server or directly to the device that registered the expression, but make use of the processing power of the involved devices and execute the evaluation of cross device expressions in a distributed way.

We believe that through developing the Cuckoo and SWAN framework, we have contributed both new insights as well as very practical tools that make structured realization of truly smart applications more viable.

### 4.2 Future Research Directions

Although we have put our best effort in providing complete frameworks, it is inherent to research to limit the scope. Furthermore, sometimes we were limited by software and/or hardware to do the research that we wanted to do. In this section we outline future research directions that can build upon the existing research.