over third-party sensors, supporting Bluetooth and ANT protocol transmissions. However, data analysis is primarily done through cloud processing and native processing algorithms have to be implemented with a proprietary API. The Open Service Architecture for Sensors (OSAS) framework[2] is an event-based programming system for sensor networks. It facilitates sensor nodes programming in a sensor network. To implement solutions, functionality needs to be implemented using regular coding. Another software framework designed for rapid prototyping of activity recognition applications is CRNT [19]. CRNT uses a component-oriented architecture, where complete data processing chains can be configured by instantiating, parameterizing, and interlinking components. Users of CRNT can thus develop an application without in-depth knowledge in programming. CRNT has been ported to many different PCbased platforms. The potential for utilizing this framework in a smartphone environment has not been investigated so far.

2.3 Processing framework approach

Our CRNTC+ architecture design follows a component oriented approach, including Readers, Writers, Filters, Classifiers, and others. The parameterisable components are incorporated through a run-time engine that can flexibly handle communication links between components in order to customize functionality. The architecture is partitioned into smartphone-specific and generic data processing layers to separate platform APIdependent components and those for general data handling. Besides smartphone-specific components, including many Readers and Writers, CRNTC+ incorporates all generic data processing components of CRNT [19] for signal handling and pattern processing. Moreover, several CRNT Writers, such as for file logging and WLAN communication, are directly usable. Figure 3 illustrates the layered design in a functional example. The basic architectural principles of component instantiation and data handling established for CRNT have been retained in the CRNTC+ framework. Component communication links can be routed within a layer and between layers. The architecture can be expanded by adding further components to both smartphone-specific and generic data processing layers. All components and communication links between them are configured and parameterized jointly through a JavaScript Object Notation[3] (JSON) based description. Hence, to design an application, components just need to be selected, parameterised, and interlinked only.

[2] OSAS, http://www.win.tue.nl/san/wsp/index.html

[3] http://www.json.org/