A complete pervasive application is composed of several architectural layers requiring different abilities, from low-level hardware programming to the design of high-level abstractions.

Part II mainly focuses on the fundamental technological issues beneath a pervasive data management system.

In Chap. 4, the different devices composing a WSN are introduced and described by means of an abstract conceptual model. This allows to query and send commands to the devices using a single database-style language. Examples will be shown using PERvasive LAnguage (PerLa), a declarative structured query language (SQL)-like language and infrastructure for data management in pervasive systems, mainly oriented to monitoring applications but, with its context management extension, also suitable as a support substratum for the deployment of autonomic systems.

The following two chapters discuss the main approaches to information flow processing: the first, introduced in Chap. 5, is more related to the viewpoint of the database research community (data stream management systems (DSMSs)); the other, introduced in Chap. 6, looks at the problem from the software engineering perspective and is more related to the distributed systems community (complex event processing (CEP) systems). The main differences among the two paradigms lie in the different ways of considering the stream:

- In DSMS, the stream is seen as an infinite set of timestamped tuples, ordered by their own timestamp. In some advanced implementation, tuples could also be annotated with an expiration time that lets the system know when a tuple has to be discarded.
- In CEP, a stream represents an infinite sequence of events. Each event is characterized by an identifier of its type and a timestamp; the CEP paradigm provides event consumption policies to let the system discard an event (or a set of events) after having used them for detecting a given sequence rather than using an event expiration time.
While DSMSs propose an SQL-like query paradigm, trying to keep those systems as similar as possible to traditional relational databases, CEPs are more like traditional publish/subscribe systems, usually allowing querying by means of logical rules. None of the two approaches exclude the other; scenarios exist in which one performs better than the other: CEPs usually perform better in fields like surveillance, while DSMSs are better suited for data analysis.

Chapter 7 addresses the problem of data integration in data streams; this requires a new generation of data integration solutions tailored to meet the requirements. Often separate organizations and, even, departments of the same organization collect and manage data independently creating so-called data silos, which must be bridged and managed through a uniform query interface. Two extreme solutions to data integration—data driven and query driven—are considered.