Cloud Application Architecture Patterns

This chapter covers architectural patterns that describe how applications have to be designed to benefit from a cloud environment. Additionally, it is described how applications themselves can be offered as configurable cloud services. Having

Fig. 4.1  Map of the cloud application architecture patterns

All figures published with kind permission of © The Authors 2014. See list of figures.

C. Fehling et al., Cloud Computing Patterns, DOI 10.1007/978-3-7091-1568-8_4, © Springer-Verlag Wien 2014
introduced cloud service models (see Sect. 2.3 on Page 42) and cloud deployment
types (see Sect. 2.4 on Page 60), this chapter describes patterns that architects and
developers can use to build cloud-native applications, i.e., applications that display
the cloud application properties introduced in Sect. 1.2 on Page 5. Following the
overview, fundamental application architectural patterns cover the architectural
principles found in most cloud-native applications to enable the cloud application
properties. Application component patterns then specify patterns on how to design
and build individual components of a cloud-native application, so that the overall
application can be built on top of an elastic infrastructure (87) or elastic platform
(91). Multi-tenancy patterns describe how cloud applications and individual
components can be shared by multiple customers, so called tenants, on different
levels of the application stack. Cloud integration patterns finally describe
mechanisms on how to integrate multiple cloud environments or cloud
environments and on-premise datacenters as well as applications both in and
outside the cloud.

4.1 Overview of Cloud Application Architecture Patterns

A cloud-native application is an application that embraces the essential cloud
properties: access via network, on-demand self-service, pay-per-use, resource
pooling and rapid elasticity. To be able to incorporate pay-per-use and rapid
elasticity, cloud native applications must be able to elastically scale to be able to
deal with varying workload (see Sect. 2.2 on Page 23). Often, the workload imposed
on different components of the same application is different.

The two fundamental cloud architecture patterns, shown at the top of the pattern
map depicted in Fig. 4.1, form the entry point to this section by describing what
cloud-native applications have to support to enable independent elastic scaling of
different parts of the application. Distributed applications (160) should be
comprised of several loosely-coupled components. Loose coupling (156) means
that application components make few assumptions about each other regarding the
format of exchanged data or the communication channels used, for example.
Components should also not be influenced by the failure of other components. In
essence, the fundamental concepts covered by these patterns are, therefore, the
decomposition of application functionality into separate components and the reduc-
tion of dependencies among these components. The following patterns describe
how different application components of such cloud-native application can be
implemented.

Cloud application components (Sect. 4.3) are characterized by three central
patterns. User interface components (175) provide application functionality to
users. Processing components (180) handle computational tasks. How this
processing can be delayed to be handled when it is most feasible is described by
the batch processing component (185) pattern. Data access components (188)
handle data stored in storage offerings (see Sect. 3.5 on Page 109). They can deal
with storage offerings at different cloud providers with different consistency levels.