

Chapter 2

Cloud Workflow System Architecture

In this chapter, we will present the general cloud workflow system architecture. System architecture in general decides how the system components are organised in different layers and how they communicate with each other. In [Sect. 2.1](#), we will first introduce the general cloud software architecture. Afterwards, [Sect. 2.2](#) will present the general architecture of cloud workflow system. Meanwhile, for each of the general architecture, the concrete architecture of a commercial system is also demonstrated.

2.1 General Cloud Software Architecture

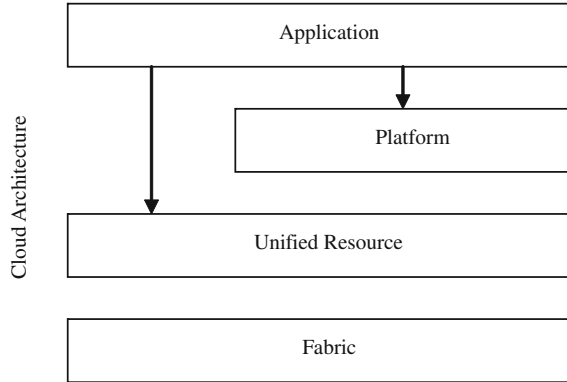
2.1.1 Cloud Architecture

There is so far no unanimous cloud software architecture. Nevertheless, in comparison to the conventional five-layer grid architecture, Ian Foster et al. proposed a representative four-layer architecture for cloud computing which has been accepted by many researchers and practitioners (see [Fig. 2.1](#)) [32].

The fabric layer consists of the raw hardware resources, such as the basic computing units, storage disks, and network bandwidths. Similar to grid computing, at this layer, most resources are heterogenous. For example, in a cloud data centre, the underlying physical machines can be commodity PCs, workstations, and supercomputers.

The unified resource layer consists of heterogeneous resources which are usually in the form of virtualised resources. In this layer, the underlying physical machines have been abstracted/encapsulated usually by virtualisation tools so that they can be exposed to upper layer and end users as integrated resources, for example, a virtual computer/cluster, a logical file system, a database system, and so on.

Fig. 2.1 General cloud software architecture [32]



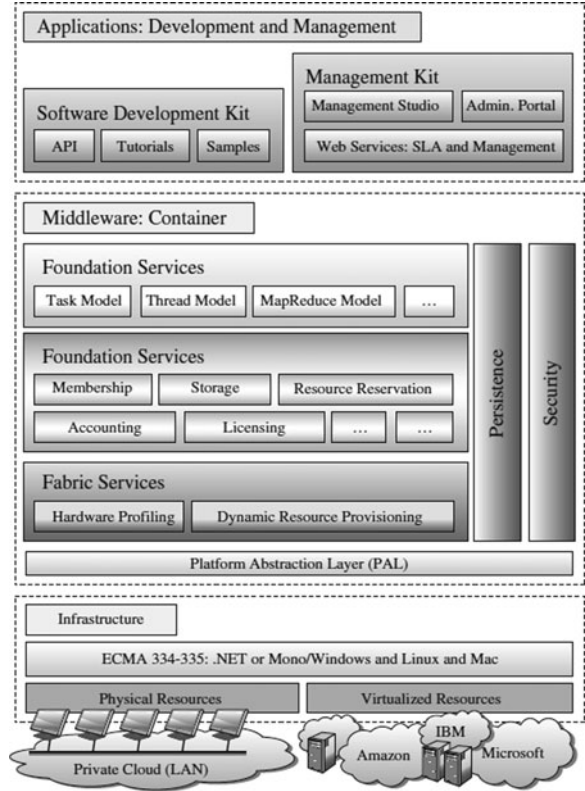
The platform layer consists of a set of resource management tools and middleware services on top of the unified resources. The platform layer can provide a development and/or deployment platform, for example, a Web hosting environment, a workflow modelling service, and a scheduling service, and so on.

Finally, the application layer consists of the user applications which can be any kind of applications such as cloud workflow applications, social networking tools, and e-commerce websites.

2.1.2 Example: Aneka Cloud Architecture

Figure 2.1 provides a high level abstract view of the general cloud software architecture. To illustrate it further additional details, we take the Aneka cloud as an example. Aneka project (<http://www.manjrasoft.com/products.html>) developed in the Cloud Computing and Distributed Systems (CLOUDS) Lab, University of Melbourne, is a software platform and a framework for developing distributed applications on the cloud. Aneka has now been commercialised by Manjrasoft Pty Ltd (<http://www.manjrasoft.com/>) as a technology to enable.NET-based enterprise cloud computing. As shown in Fig. 2.2, the fabric layer of Aneka cloud can contain the physical resources in the private cloud and virtualised resources in the public cloud provided by such as Amazon, IBM or Microsoft. The unified resource layer and the platform layer are represented by the Platform Abstraction Layer (PAL) and its core is the Aneka container. The Aneka container is the building block of the middleware and represents the runtime environment for executing cloud applications. There are three classes of services in the container, viz. the fabric services which provide the access to the cloud resources, the execution services which are responsible for the scheduling and executing applications, and the foundation services which are the core management services in charge of metering applications, allocating resources, managing available nodes,

Fig. 2.2 Architecture of aneka cloud [17]



and keeping the services registry updated. In the application layer, Aneka provides a tool for managing the cloud, allowing administrators to easily start, stop, and deploy instances of the Aneka container on new resources and then reconfigure them dynamically to alter the behavior of the cloud. For more details about the Aneka cloud, please refer to [17, 75].

Corresponding to the different layers, clouds in general provide services at three different levels (IaaS, PaaS, and SaaS [32]) as follows, although some providers can choose to expose services at more than one level.

Infrastructure as a Service (IaaS) [19] provisions hardware, software, and equipments (mostly at the unified resource layer, but can also include part of the fabric layer) to deliver software application environments with a resource usage-based pricing model. Infrastructure can scale up and down dynamically based on application resource needs. Typical examples are Amazon EC2 (Elastic Cloud Computing) Service (<http://aws.amazon.com/ec2/>) and S3 (Simple Storage Service) (<http://aws.amazon.com/s3/>) where compute and storage infrastructures are open to public access with a utility pricing model; Eucalyptus [61] is an open source Cloud implementation that provides a compatible interface to Amazon's

EC2, and allows people to set up a Cloud infrastructure at premise and experiment prior to buying commercial services.

Platform as a Service (PaaS) [17, 46] offers a high-level integrated environment to build, test, and deploy custom applications. General speaking, developers will need to accept some restrictions on the type of software that they can write in exchange for built-in application scalability. An example is Google's App Engine (<http://code.google.com/appengine/>), which enables users to build Web applications on the same scalable systems that power Google applications.

Software as a Service (SaaS) [58] delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model. Salesforce (<http://www.salesforce.com/>) is an industry leader in providing online Customer Relationship Management (CRM) services. Live Mesh (<http://explore.live.com/windows-live-mesh>) from Microsoft allows files and folders to be shared and synchronised across multiple devices.

Although clouds provide services at three different levels (IaaS, PaaS, and SaaS), standards for interfaces to these different levels still remain to be defined. This leads to interoperability problems between today's clouds, and there is little business incentives for cloud providers to invest additional resources in defining and implementing new interfaces. As clouds mature, and more sophisticated applications and services emerge that require the use of multiple clouds, there will be growing incentives to adopt standard interfaces that facilitate interoperability in order to capture emerging and growing markets in a saturated cloud market.

2.2 General Architecture of Cloud Workflow Systems

2.2.1 Cloud Workflow System Architecture

In this section, we will present a general architecture of a cloud workflow system. Obviously, as typical cloud software itself, the architecture of a cloud workflow system should be consistent to the general cloud software architecture. Therefore, as shown in Fig. 2.3, the general cloud workflow architecture can be a mapping of the general cloud system architecture.

Specifically, the application layer consists of cloud workflows (workflow applications for real-world business processes).

The platform layer is the cloud workflow system which provides a development and running platform for cloud workflows. All the system functionalities of a cloud workflow system such as workflow management, cloud resource management and QoS management are included. The application layer and the platform layer are usually self-maintained by the business organisation.¹

¹ A cloud workflow system can be encapsulated as a platform service, i.e. PaaS (platform as a service). In such a case, the platform layer is maintained by external cloud service providers.

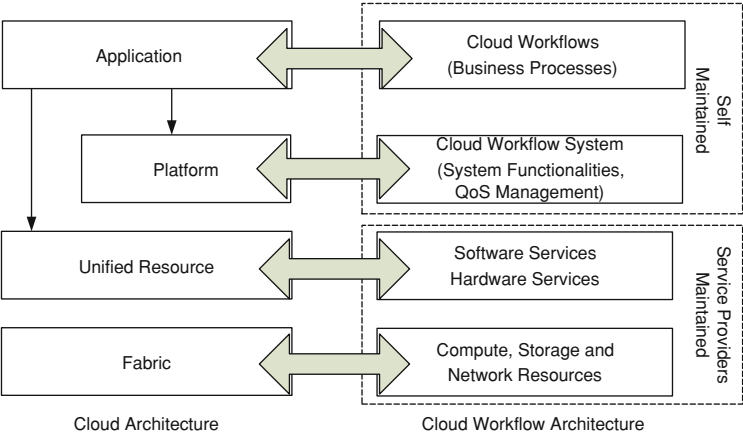


Fig. 2.3 Cloud workflow system architecture

The unified resource layer consists of both software services and hardware services that are required for the running of cloud workflows. Specifically, SaaS can provide massive number of software capabilities for processing different business tasks, while IaaS can provision on-demand and elastic computing power to meet the resource requirements for processing business tasks. In practice, software and hardware services can also be integrated together and encapsulated to be delivered as VMs (virtual machines).

The fabric layer is composed of low level hardware resources such as computing, storage and network resources. The unified layer and fabric layer are usually maintained by external cloud service providers.²

2.2.2 Example: Window Workflow Foundation Architecture

Here, as an example, we demonstrate the Windows Workflow Foundation (WWF, <http://msdn.microsoft.com/en-us/netframework/aa663328>) which is part of Microsoft.NET framework providing a foundation for developing workflow applications. Microsoft has developed the Windows Azure to help developers build, host and scale applications through Microsoft data centres. Windows Azure (<http://www.microsoft.com/windowsazure/>) is an operating system that serves as the development, service hosting, and service management environment for the Windows Azure platform. The Windows Azure platform consists of an infrastructure of hardware, software, network, and storage resources. Developers can

² The fabric layer can also be a virtual collection of local computing infrastructure (i.e. private cloud) and the commercial computing infrastructure (i.e. public cloud), i.e. hybrid cloud.

Fig. 2.4 Architecture of WWF based cloud workflow system



build and deploy applications as a hosted service for Windows Azure by using the .NET Framework. Therefore, the Windows Azure platform is one of the ideal hosting environment for WWF based workflow system.

Figure 2.4 depicts the architecture of a WWF based cloud workflow system. The fabric and unified resource layers are built upon the Microsoft data centres with Windows Azure, so that they can offer basic computing, storage and network resource to the upper layers. The platform layer is WWF. WWF consists of three main components, viz. base activity library, runtime engine, and runtime services.³ The base activity library contains frequently used workflow activities such as assign, delay, invoke and so on. The runtime engine is the heart of WWF which consists of the runtime classes and services required for the workflow execution. The runtime services are responsible for such as scheduling activities, event handling, exception, tracking and so on. At the application layer, the visual designer provides customers a tool for flowchart-based workflow modelling with activities such as if, sequence, pick, and parallel. In addition, customers can also add custom-built activities into the customer activity library for use in the workflow.

³ A Developer's Introduction to Windows Workflow Foundation (WF) in .NET 4, <http://msdn.microsoft.com/en-us/library/ee342461.aspx>.

The Design of Cloud Workflow Systems

Liu, X.; Yuan, D.; Zhang, G.; Li, W.; Cao, D.; He, Q.; Chen, J.; Yang, Y.

2012, XIV, 97 p. 33 illus., Softcover

ISBN: 978-1-4614-1932-7