PAAS

PAAS-Definition

- Platform as a Service providers commonly offer a development and deployment environment that allow users to create and run their applications with little or no concern to low-level details of the platform.
- The traditional development environments can be used to design and develop applications, which are then deployed on the cloud by using the APIs exposed by the PaaS provider.
- PaaS offerings may include facilities for application design, application development, testing, deployment and hosting.

PAAS-Defintion II

- The capability provided to the consumer to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider."
- NIST goes on to say, "The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment."
- Said another way, PaaS provides developers with easier ways to create and deploy software onto cloud infrastructure. Those "easier ways" typically exist as GUIs, sandboxes, programming languages, shared services, APIs and other online tools for software developers.

PAAS Users

- PaaS consumers can be
  - application developers who design and implement application software
  - application testers who run and test applications in a cloud-based environment
  - application deployers who publish applications into the cloud
  - application administrators who configure, monitor and manage applications deployed in a cloud
- PaaS consumers can be billed according to
  - the number of PaaS users,
  - the processing, storage and network resources consumed by the PaaS application;,
  - and the duration of the platform usage

Requirements of a PaaS Platform

1. High Scalability and On Demand Provisioning

- PaaS is expected to scale applications across the hardware and the extent of scaling can be stretched to include the last hardware resource available for deployment.
- This provides users of PaaS a feeling of infinite scalability. In addition, the application provisioning should be an automated task that needs no IT intervention for deployment and delivery.

2. High Availability

- PaaS platforms should provide a runtime environment for applications that features failover and load balancing capabilities.
- The important question is “how is it different from a traditional clustered, load-balanced environment?”
- The answer is that failover and load balancing capabilities should be scoped across the cloud rather than a few dedicated machines, as is the case in a traditional environment.

3. High Reliability

- A successful PaaS platform should provide this reliability to all services/components deployed and running under them.

Requirements of a PaaS Platform

4. Optimal Usage

- One of the core requirements of any cloud computing platform is optimal usage of resources.
- In the case of PaaS, optimization specifically applies to resources utilized for executing applications. To apply resource optimization, the PaaS platform should have components that monitor application execution and usage.
- Another purpose of monitoring is to provide chargeback to the users. The second reason is undecided about being required across all scenarios but the former is certainly needed

5. Auto-Scaling

- The on-demand scenario could be based on a user request or in response to an increased load. In the latter scenario, the cloud because of its elastic nature expands and adds more resources to meet the increased demand.
- This requires the PaaS to auto scale the applications in the newly added computing resources.

6. Admin /Management Console and Reports

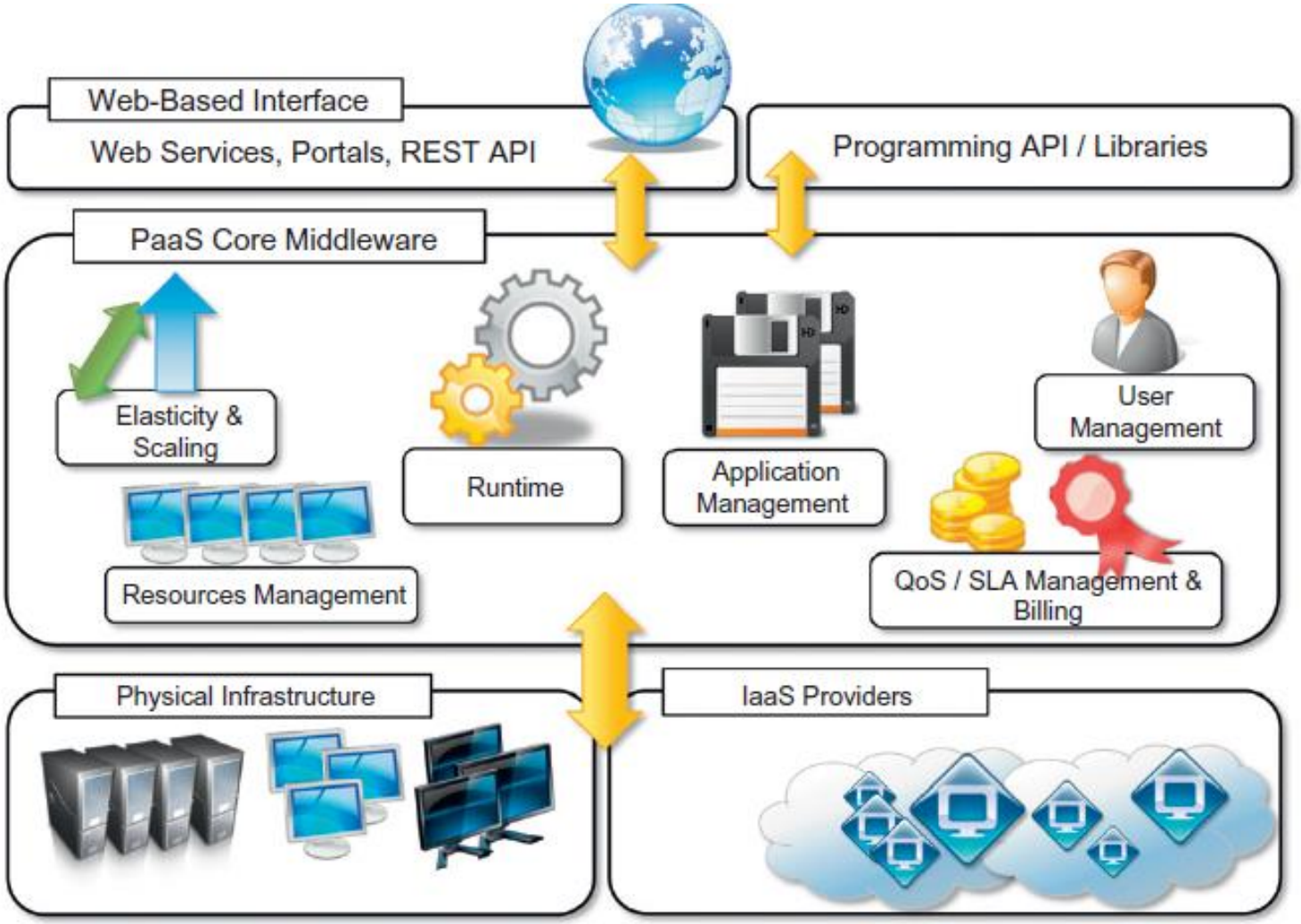
- PaaS platforms should include some form of a user interface through which all application components/services can be tracked and monitored.
- PaaS platforms should also have reporting capabilities to provide statistics related to application usage, execution and provisioning.

Requirements of a PaaS Platform

7. Multi OS and Multi Language Support

- An organization might have different OS and applications written using different languages.
- PaaS platforms should enable runtimes which can run on multiple OS (Windows, Linux etc.) and should be able to run applications created in different languages (Java, .Net, C++ etc.).

The Platform-as-a-Service reference model



The Platform-as-a-Service reference model

- **PAAS Core MiddleWare :**
  - PAAS constitute the middleware on top of which applications are built.
  - Application management is the core functionality of the middleware.
  - PaaS implementations provide applications with a runtime environment and do not expose any service for managing the underlying infrastructure.
  - They automate the process of deploying applications to the infrastructure, configuring application components, provisioning and configuring supporting technologies such as load balancers and databases, and managing system change based on policies set by the user.
  - Developers design their systems in terms of applications and are not concerned with hardware (physical or virtual), operating systems, and other low-level services.
  - The core middleware is in charge of managing the resources and scaling applications on demand or automatically, according to the commitments made with users.
  - From a user point of view, the core middleware exposes interfaces that allow programming and deploying applications on the cloud. These can be in the form of a Web-based interface or in the form of programming APIs and libraries.

The Platform-as-a-Service reference model

- PaaS providers can provide a runtime environment for the developer platform
- Runtime environment is automatic control such that consumers can focus on their services

Dynamic provisioning

- On-demand resource provisioning

Load balancing

- Distribute workload evenly among resources

Fault tolerance

- Continuously operating in the presence of failures

System monitoring

- Monitor the system status and measure the usage of resources

Platform-as-a-Service reference model -Features

- The specific development model decided for applications determines the interface exposed to the user.
- specific programming languages and frameworks are made available in the platform, as well as other services such as persistent data storage and in memory caches

Programming Models, languages and Frameworks

1. Some implementations provide a completely Web-based interface hosted in the cloud and offering a variety of services.
   - It is possible to find integrated developed environments or rapid prototyping environments where applications are built by assembling mash-ups and user-defined components and successively customized.

2. Other implementations of the PaaS model provide a complete object model for representing an application and provide a programming language-based approach.
   - This approach generally offers more flexibility and opportunities but incurs longer development cycles.
   - Developers generally have the full power of programming languages such as Java, .NET, Python, or Ruby, with some restrictions to provide better scalability and security.

Platform-as-a-Service reference model -Features

- PaaS providers usually support multiple programming languages. Most commonly used languages in platforms include Python and Java (e.g., Google AppEngine), .NET languages (e.g., Microsoft Azure),and Ruby (e.g., Heroku). Force.com has devised its own programming language (Apex) and an Excel-like query language, which provide higher levels of abstraction to key platform functionalities.
- A variety of software frameworks are usually made available to PaaS developers, depending on application focus.
- Providers that focus on Web and enterprise application hosting offer popular frameworks such as Ruby on Rails, Spring, Java EE, and .NET.

Persistence Options

A persistence layer is essential to allow applications to record their state and recover it in case of crashes, as well as to store user data

- Traditionally
  - Web and enterprise application developers have chosen relational databases as the preferred persistence method.
  - but may lack scalability to handle several petabytes of data stored in commodity computers

Platform-as-a-Service reference model -Features

- In the cloud computing domain,
  - distributed storage technologies have emerged, which seek to be robust and highly scalable, at the expense of relational structure and convenient query languages.
  - For example, Amazon SimpleDB and Google AppEngine datastore offer schema-less, automatically indexed database services

PaaS implementations Categories

- PaaS solutions can offer middleware for developing applications together with the infrastructure or simply provide users with the software that is installed on the user premises.
- In the first case, the PaaS provider also owns large datacenters where applications are executed;
- in the second case, the middleware constitutes the core value of the offering.
- It is also possible to have vendors that deliver both middleware and infrastructure and ship only the middleware for private installations.
- It is possible to organize the various solutions into three wide categories:
  - *PaaS-I,*
  - *PaaS-II,*
  - *PaaS-III.*

PaaS implementations Categories-PaaS-I,

- The first category identifies PaaS implementations that completely follow the cloud computing style for application development and deployment.
- They offer an integrated development environment hosted within the Web browser where applications are designed, developed, composed, and deployed.
- This is the case of Force.com and Longjump. Both deliver as platforms the combination of middleware and infrastructure.

PaaS implementations Categories-PaaS-2

- In the second class we can list all those solutions that are focused on providing a scalable infrastructure for Web application, mostly websites.
- In this case, developers generally use the providers' APIs, which are built on top of industrial runtimes, to develop applications.
- Google AppEngine is the most popular product in this category. It provides a scalable runtime based on the Java and Python programming languages, which have been modified for pro- viding a secure runtime environment and enriched with additional APIs and components to support scalability.
- AppScale, an open-source implementation of Google AppEngine, provides interface-compatible middleware that has to be installed on a physical infrastructure.
- Joyent Smart Platform provides a similar approach to Google AppEngine.
- A different approach is taken by Heroku and Engine Yard, which provide scalability support for Ruby- and Ruby on Rails-based Websites.
- In this case developers design and create their applications with the traditional methods and then deploy them by uploading to the provider's platform.

PaaS implementations Categories-PaaS-3

- The third category consists of all those solutions that provide a cloud programming platform for any kind of application, not only Web applications.
- Among these, the most popular is Microsoft Windows Azure, which provides a comprehensive framework for building service- oriented cloud applications on top of the .NET technology, hosted on Microsoft's datacenters.
- Other solutions in the same category, such as Manjrasoft Aneka, Apprenda SaaSGrid, Appistry Cloud IQ Platform, DataSynapse, and GigaSpaces DataGrid, provide only middleware with differnt services.

PaaS implementations Categories

Table 4.2 Platform-as-a-Service Offering Classification

Category	Description	Product Type	Vendors and Products
PaaS-I	Runtime environment with Web-hosted application development platform. Rapid application prototyping.	Middleware + Infrastructure Middleware + Infrastructure	Force.com Longjump
PaaS-II	Runtime environment for scaling Web applications. The runtime could be enhanced by additional components that provide scaling capabilities.	Middleware + Infrastructure Middleware Middleware + Infrastructure Middleware + Infrastructure Middleware + Infrastructure Middleware	Google AppEngine AppScale Heroku Engine Yard Joyent Smart Platform GigaSpaces XAP
PaaS-III	Middleware and programming model for developing distributed applications in the cloud.	Middleware + Infrastructure Middleware Middleware Middleware Middleware Middleware	Microsoft Azure DataSynapse Cloud IQ Manjrasof Aneka Apprenda SaaSGrid GigaSpaces DataGrid

Key Characteristics of PAAS

1. Multi-tenant architecture
   - Multi-tenant architecture is one of the key elements of PaaS model.
   - A multi-tenant platform is one that uses common resources and a single instance of both the object code of an application as well as the underlying database to support multiple customers simultaneously.
   - Although the technology stack in a multi-tenant deployment is shared, customers' user experience should be comparable to that of a customer using an application developed and managed by dedicated resources.
2. Customizable/Programmable User Interface
   - The PaaS offering should provide the ability to construct highly-flexible user interfaces via a simple "drag & drop" methodology which permits the creation and configuration of UI components on the fly.

Key Characteristics of PAAS

3. Unlimited Database Customization
   - Data persistence is core to many applications and facilitating the creation, configuration and deployment of persistent objects without requiring programming expertise is a key characteristic of a powerful “cloud platform”.
   - Thus, the PaaS offering must support the construction of objects, the definition of relationships between the objects and the configuration of advanced data behavior all from within the comfort of the web browser via a “point and click’ declarative paradigm.
4. Robust Workflow engine/capabilities
   - Successful business process execution via process automation is the primary objective of any business application. A “cloud-platform” must offer a business-logic engine that supports the definition of workflow processes and the specification of business rules to engender process automation.

Key Characteristics of PAAS

5. Granular control over security/sharing (permissions model)
   - Granular secure access to appropriate data is one of the counterstone of the PaaS model.
   - Just within the PaaS model, secure access models might span multiple logical domains.
6. Flexible “services-enabled” integration model
   - However given the complex IT environments that permeate most enterprises today, the PaaS offering should leverage Service Oriented Architecture (SOA) principles to enable seamless integration of “cloud” application data and functionality residing in the “cloud-platform” with other on-premise/on-demand systems and applications

Other Characteristics of PAAS

- *Support for custom applications*
- *Provision of runtime environments*.
- *Rapid deployment mechanisms*.
- *Support for a range of middleware capabilities*
- *Provision of services*
- *Preconfigured capabilities*
- *API Management capabilities*.
- *Security capabilities*
- *Tools to assist developers*
- *Support for porting existing applications*
- *Operations capabilities.*

Benefits of PaaS

1. Improving the development life cycle:
   - ✓ Effectively managing the application development life cycle can be challenging.
   - ✓ For example, teams may be in different locations, with different objectives, and working on different platforms. When it comes time to integrate, test, and build the application, problems can arise because developers are working on different platforms with a different configuration than the operations team is working on.
2. Eliminating the installation and operational burden from an organization:
   - ✓ Traditionally, when a new application server or other middleware is introduced into an organization, IT must make sure that the middleware can access other services that are required to run that application. This requirement can cause friction between Development and Operations.
   - ✓ With PaaS, these conflicts are minimized. Because the PaaS environment is designed in a modular, service-oriented manner, components can be easily and automatically updated. When PaaS is provided by a third-party organization, those changes occur automatically without the user having to deal with the details.
3. Implementing standardization
   - ✓ PaaS enables development professionals and IT operations professionals to use the same services on the same platform. This approach takes away much of the misunderstanding that happens when the two teams with different responsibilities aren't in sync

Benefits of PaaS

4. Having ease of service provisioning

- ✓ A PaaS provides easy provisioning of development services including build, test, and repository services to help eliminate bottlenecks associated with non-standard environments.
- ✓ This in turn improves efficiency, reduces errors, and ensures consistency in the management of the development life cycle. Additionally, PaaS provides ease of provisioning in runtime services that include application runtime containers for staging, and running and scaling application