

# Cloud IDEs state of play and JPlaton Academy

Androklis Mavridis,  
Giorgos Fylaktopoulos,  
Spyros Skolarikis  
{andy, gfylak, ssk@b-open.gr}

Ver. 1.0 – November 2013

## **Overview**

*According to Gartner, the worldwide PaaS revenue is expected to reach 2.9 billion US dollars by 2016, representing a 26.6% CAGR. This fierce market competition orders the provision of integrated application development environments (IDE), which will be offered together or independently with the SaaS applications. Although, there exists a wide range of web-based IDEs the main challenge for costs minimization is yet to be addressed. We propose a homogeneous environment which “hides” the implementation and deployment details, while allowing designers/developers to focus primarily on the domain and application logic. The purpose of JPlaton Academy (JPA) is to provide a Java development framework for cloud applications facilitating the modeling, composition, configuration and expansion of new or pre-existing cloud applications. To achieve this goal JPA fully exploits the Model Driven Development (MDD) paradigm to simplify the designing process and to automate most of the development phases.*

## **Introduction**

Driven mainly by the widespread deployment of cloud solutions, there is a growing demand for developing cloud-based development environments -IDEs. The benefit is that an ever increasing number of online applications are globally available and easily accessible through any modern browser. According to Gartner, Inc. [1], the revenue from cloud-based application development environments worldwide reached 1.2 billion US dollars in 2012 from 900 million in 2011. The PaaS market achieved continuous growth with worldwide PaaS revenue totaling 1.5 billion in 2013 and expected to reach 2.9 billion US dollars 2016, representing a 26.6% CAGR.

Growth rates by PaaS sub-segment include the following: IDEs (Integrated-Development-Environment) accounting for 34.4 % of the total, followed by Services-Life-Cycle- Management applications with 12%, Business-Process-Management applications with 11.6% and Integration with 11.4%. The authors of this report claim that vendors should seize the opportunity and create new platforms for application development (IDE) and therefore cover the need for rapid SaaS implementation.

## **Challenges**

Although the “cloud approach” is considered cost effective, extensive knowledge and expertise is required for creating and deploying cloud applications. This knowledge covers topics such as specialized APIs for interfacing with infrastructures, data models, and tools for service modeling, development, and deployment. Moreover, a wide range of cloud-based IDEs are available, each introducing different programming languages, APIs, data models and development tools. As a consequence, both skills and experience acquired for one platform cannot be reused for another [2].

Another challenge is the minimization of costs both in terms of development time and in relation with the reusability of existing software components in order to create new services. The adoption of software development models, based on existing components (Component-based application development), inherently supports high adaptability and scalability and allows a faster way of constructing applications, as developers focus on basic functional components to create new, higher level of services. Especially for non-experienced and skilled developers, the ability to develop new applications through components’ synthesis is of paramount importance [3].

The fierce SaaS market competition orders the provision of integrated application development environments (IDE), which will be offered together or independently with the SaaS applications.

These platforms should cover the need for integrated web application development tools for cloud application development and provide solutions for inter-connectivity with third party applications for quick and cost effective application synthesis.

### ***State of the art in Cloud-based IDEs:***

Many online IDEs have emerged during the past years. Their type varies from simply providing training material for programming to integrated solutions for developing cloud applications. Recent approaches mainly focused on the collaborative aspect of integrated development environments and related challenges faced by developers [4]. The NodeJS<sup>1</sup> execution environment allows the programmer to write the server-side code with JavaScript and thus enrich server-side with additional functionalities. Notable Web IDE encompassing NodeJS is the Cloud9<sup>2</sup>.

The Complir<sup>3</sup> supports numerous programming languages as well as the ability to embed code libraries from independent suppliers (third party). The CloudIDE<sup>4</sup> and Codeanywhere<sup>5</sup> also support a variety of solutions, and offer development aids such as automatic code completion and debugging of applications. The Orion<sup>6</sup> supports HTML5, JavaScript and CSS, and has version control capabilities similar to those of Cloud IDE. Representing the C# family of solutions, Coderun<sup>7</sup> allows the development and debugging of C # applications incorporating Sharpkit<sup>8</sup> for the client-side code creation. Kodingen<sup>9</sup> provides an easy way for the development, deployment and hosting of applications using various languages and frameworks and supports a variety of code editors and source code control tools. Arvue<sup>10</sup> is a cloud based IDE for creating and publishing small web applications for Arvue Java framework. Finally, Neutron IDE<sup>11</sup> uses the powerful Ace code editor<sup>12</sup> as its starting point and merges the best features of SFTP clients and browser editors into one package, allowing coders to edit files on their development servers from anywhere.

### ***The approach of JPlaton Academy:***

- 
- 1 <http://nodejs.org>
  - 2 <http://cloud9ide.com>
  - 3 <https://compilr.com>
  - 4 <http://cloud-ide.com>
  - 5 <https://codeanywhere.net/>
  - 6 <http://www.eclipse.org/orion>
  - 7 <http://coderun.com>
  - 8 <http://sharpkit.net>
  - 9 <http://kodingen.com>
  - 10 <http://dev.vaadin.com/wiki/Arvue>
  - 11 <https://www.neutrondrive.com/>
  - 12 <http://ace.ajax.org>

Trying to address the aforementioned challenges, b.Open Ltd<sup>13</sup> introduces the **JPlaton Academy (JPA)**. JPA, currently in prototyping phase, is a cloud based "multi-tenant" application development environment which allows: multiple **users** to operate multiple **applications** provided by multiple **cloud providers**, implemented by multiple **developers**.

The purpose of JPA is to provide a Java development framework for cloud applications that facilitates the modeling, composition, configuration and expansion of new or pre-existing applications based on implementation through reuse of key "elementary" software components. The basic idea is to address the need for a homogeneous environment which "hides" the implementation and deployment details, while allowing designers/developers to focus primarily on the domain and application logic. To achieve this goal and to allow the reuse of existing software components, JPA fully exploits the Model Driven Development (MDD) paradigm offered by JPlaton<sup>14</sup> which separates the business logic from the development details. This separation simplifies the designing process as it reduces the actual development lifetime by automating most of the development phases.

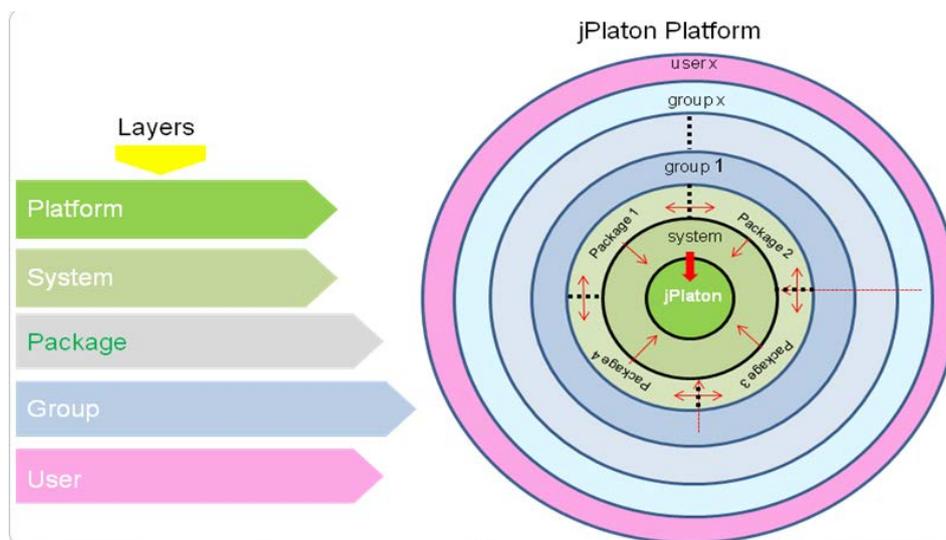


Figure1. JPlaton Reference Architecture

Hence, in JPA the implementation of applications is made by the composition of smaller software components (programming units) whose functionality is contained in XML files which describe the objects' properties and behavior. The intended functionality is defined and modeled at abstract level, through Sequence and Collaboration diagrams (UML 2.0). The organization of the programming units is achieved by creating sets of functional layers hierarchically structured. Each

<sup>13</sup> [www.b-open.com](http://www.b-open.com)

<sup>14</sup> <http://www.b-open.com/index.php/en/jplaton-technology>

layer may add new functionality or alter (update or remove) the functionality of previous (inner) layers. The number and nature of the layers is determined by the number and nature of functional requirements of each application.

The development of applications in JPA encompasses the following steps, shown in figure 2:

1. **Search and Load programming units:** The analyst can search for existing programming units in the Repository. The programming units can be basic or complex (composite: Derived from the synthesis of other basic units). For the composite ones, existing collaboration and sequence diagrams (if any) for each of these is loaded in order to assist the analyst in the planning process.
2. **Modeling:** JPA offers a modeling suite through which the analyst can:
  - i. Create data-mapping diagrams: The programmer can create diagrams for mapping the desired data elements (from existing or new databases) to existing or new programming units.
  - ii. Synthesize Diagrams: Selecting the desired programming units and performing the data-mappings, the analyst can build new models to describe the desired application.
  - iii. Design the UI: The suite allows the design of new GUI or the modification of existing one.
3. **Implementation - Processing Code:** The developer can:
  - i. Build code automatically from the models created above, and write code for new programming units or for expanding the functionality of existing ones. The environment gives all these state of the art assistive tools such as code auto completion, syntax highlighting, composition, compiling etc.
  - ii. Test the generated code for potential logic errors in debug mode.
4. **Deployment - Controlling:** JPA connects to JPlaton application server for:
  - i. Deploying the newly created application.
  - ii. Controlling new versions of the application (version-control).

The process is shown in the following figure.

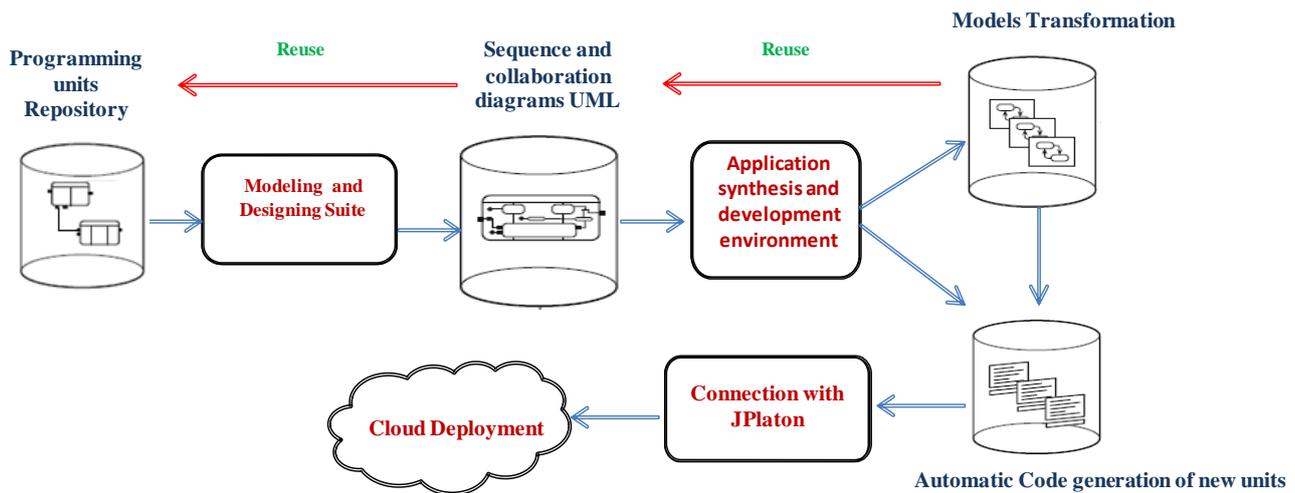


Figure.2 The JPlaton Academy development process

### Conclusion

Numerous web-based development IDEs exist offering great assistance to developers seeking seamless and fast development of cloud-based solutions. Nevertheless, this plethora of tools requires knowledge and expertise that is not easily transferable from one solution to the other. Moreover, they do not provide a holistic approach to cost minimization through extensive reuse of pre-existing models and components.

JPA addresses these shortcomings by incorporating the knowledge repository in which software development models and existing components are stored and are available for reuse. Exploiting the inherent adaptability and scalability of JPlaton application server, all intended functionality is loaded hierarchically into distinct layers that can be modified and/or extended seamlessly even by non-experienced/skilled developers. Hence, JPA covers the whole range of developers and the whole spectrum of business needs for specialized solutions/applications.

We plan to release the first JPA working version by the end of November 2014. B-Open will provide periodic updates of this working report presenting the progress of JPA and related technical advancements.

## References

- [1] F. Biscotti, Y. Natis, M. Pezzini, T. Murphy, P. Malinverno, M. Cantara, D. Feinberg, W. Schulte, T. Friedman, J. Thompson, B. Lheureux, E. Thoo and B. Huang, "Market Trends: Platform as a Service, Worldwide, 2012-2016, 2H12 Update," Gartner, 2012.
- [2] L. Kats, R. Vogelij, K. Kalleberg and E. Visser, "Software Development Environments on," Delft University of Technology, 2012.
- [3] S. Kächele, J. Domaschka and F. J. Hauck, " COSCA: an easy-to-use component-based PaaS cloud system for common applications," in *Proceedings of the First International Workshop on Cloud Computing Platforms (CloudCP '11)*, ACM, New York, NY, USA, 2011.
- [4] W. Liang, G. Kui and S. Wang, "CEclipse: An online IDE for programing in the," in *IEEEWorld Congress on Services (SERVICES)*, 2011.

## Authors

**Androklis Mavridis**

Comidor Product Manager

**Giorgos Fylaktopoulos**

Head R&D

**Spyros Skolarikis**

CEO

**b.open - Business Open Software Applications Ltd**

*Laskaratou 11A Thessaloniki Pylaia 54250, Greece*

<http://www.b-open.gr>

[info@b-open.gr](mailto:info@b-open.gr)

+30 2310 402522

+30 2310 403652

skype: comidor.support

<http://www.comidor.com>

[support@comidor.com](mailto:support@comidor.com)



Copyright © 2013 b-Open Ltd. All rights reserved. b-Open and the b-Open logo are trademarks or registered trademarks of b-Open - Business Open Software Applications Ltd.