

# Integrated Development Environment as a Service (IDEaaS) - Models and Architecture part of the Google Cloud Core Services

Orges Cico

Canadian Institute of  
Technology  
Rr. Andon Zako Cajupi, Zayed  
Center, Tirana, Albania

Zamir Dika

South East European  
Univeristy  
SEEU, Ilindenska 335, Tetovo  
1200, Macedonia (FYROM)

Betim Cico

EPOKA Univeristy  
Rr. Tiran-Rinas, Km. 12, 1032  
Vore, Tirana, Albania

## ABSTRACT

This paperwork proposes the implementation of a new cloud service IDEaaS (Integrated Development Environment as a Service) to become part of the existing core services. The development of an IDE environment fully supporting Google Cloud Python SDK (Software Development Kit) on the cloud infrastructure is the primary focus of this work. The objective is in fulfilling the need to have a development environment totally independent from personal desktop environments as part of the cloud core services. The paper work also addresses how this migration could become part of the rest of cloud infrastructures provided from Microsoft, Amazon and any other cloud investors in the market. This may be part of the services and business models adopted from the major existing Cloud providers as well as new entries in this important emerging technology. Two new and important cloud application business models derive from the IDEaaS: PaygoC (Pay as you go Coding) and ODC (On Demand Coding). This paper work is going to demonstrate their feasibility as well as unfold their benefits.

## General Terms

GAE Launcher: Google App Engine Launcher; IDEaaS: Integrated Development Environment as a Service; ODC: On Demand Coding; PaygoC: Pay as you Go Coding.

## Keywords

IDE as a Service, Cloud SDK, Cloud providers, Pay as you go Coding, On Demand Coding.

## 1. INTRODUCTION

This work proposes the migration of the current cloud Software Development Kit (SDK) on the cloud PaaS (Platform as a Service) infrastructure. A review among existing major cloud providers (Amazon, Google, Azure) has been performed, as well as their approaches for online software development [33]. Some have proven to be more ahead of the others but none of them have fully integrated a collaborative environment for large-scale projects. Either ad-hoc [8,21,22] or third party solutions [4] have been adopted so far. However, both choices lack proper integration with cloud providers or do not provide a proper business model which would fit best the cloud tenants and end-users. Although, advantages of utilizing online development environments have been largely understood and discussed [1,2,28].

The paper also describes the IDEaaS architecture and compare two possible models how it can become part of the different cloud infrastructures (either as a SaaS or as a new service per se). Proof of integration has also been provided through an experimental service running on Google App Engine called

GAE Launcher. The tool has been ported from its current desktop version to the cloud platform offering similar existing features as well as new ones.

It currently works with python SDK but more programming languages can be added with the same approach. The new framework allows new business model opportunities as well, which might either be a derivative of similar approaches already adopted on cloud platforms such as Pay as you go Coding (PaygoC) or On Demand Coding (ODC) for large scale collaborative projects and new outsourcing opportunities. Both proposed models have been clarified within the paper and would give cloud providers a quick economical boost in entering new open source development communities and enterprise tenants. From the IDEaaS cloud development and project management would benefit from reducing setup overheads (infrastructure, SW, HW, licensing etc.), increasing productivity in large-scale projects, decreasing costs and outsourcing opportunities. The new service would make cloud providers enter new markets, offer better services and would increase end users productivity.

## 2. BACKGROUND

### 2.1 Cloud SDK and IDE Tools

The cloud SDK commonly represents a set of tools for Cloud Platforms, which allow the end user to fully exploit the cloud infrastructure at different service level (PaaS, IaaS). Such tools are usually part of CLI or GUI interfaces mostly common in the major cloud providers. The SDK-s also support multiple programming languages and the primary objective is to allow not just development but also management of virtual clusters of computers. Each of which have similar properties to personal computers and provide primarily virtualization of basic computing components and I/O, so the systems can provide services starting from Web servers and continuing with CRM, database, computation etc. Most of the services for every major cloud provider such as Amazon, Azure and Google operate with on-demand or pay per use strategy (PAYG). The platforms are very flexible for every individual, business or governmental sector.

### 2.2 Cloud Supported Programming Languages and Technologies

The set of programming languages and environments

supported proves the flexibility of current cloud platforms to support a large scale of technologies, Table 1.

**Table 1. Supported programming languages of major cloud providers**

Cloud Provider	Denomination	PaaS Supported Programming Languages
Google	Google App Engine	Go, PHP, Java, Python, Node, .NET, Ruby
Amazon	AWS Elastic Beanstalk	Java, .NET, PHP, Node.js, Python, Ruby, Go
Microsoft Azure	Azure Cloud Services	Java, .NET, PHP, Node.js, Python, Ruby

The establishment of an integrated cloud based IDE environment on demand would not just reduce the scale of licensing, but also contribute into optimizing individual, business and enterprise resources when it comes to development of large scale applications. Most developers wont be restricted from proprietary tools of third party entities and shall rely mostly on the cloud infrastructure to keep their development going.

This would not just reduce the operational cost but also help in increasing the productivity and collaborative approaches for distributed working environments. The possibility based on the similarities in programming languages and technologies could also contribute into integrating further different cloud providers. IDE environment have been widely influenced from polyglot, cross platform and collaborative software development. IDEaaS as some of the samples presented at the next section try to address the long term issue bringing the three characteristics together so that cloud development becomes as easy and efficient as the rest of the services already provided.

### 3. STATE OF THE ART AND MOTIVATION

There have been few but important initiatives related to move development environments on web browsers. Some of them were specifically related to particular needs (such as embedded computing and software development [23] others more general [25]). One of the first was founded since 2010 by providing basic features for code highlighting and more advanced related to cloud code deployment for major providers such as AWS (Amazon Web Services), Microsoft Azure, Google App Engine. Only recent environments have however tried to fully integrate the IDE with the cloud SDK by facilitating not just the development, testing and debugging but also the deployment process. Some of them worth mentioning in detail are Cloud 9 and Condevy [6, 11]. They are not the only ones operating in the market. Mainly for educational purposes we can also mention others such as CloudForge [7], CodePlex [9], Compilr [10], jsFiddle [24], Eclipse Orion, CodeAnywhere [5], Koding etc. Most of them are offering different integration levels with editors, documentation, compilation, versioning service integration such as Git [17], GitLab [18] etc.

#### 3.1 Cloud 9

Founded in 2010 proves the concept of moving IDE oriented services to the web. Thus code development and accessibility

is made easy enough and relies on browser development. Acquired from the Amazon Cloud Services, it relies on the web EC2 Operating System or other web oriented technology such as Aurora file system and the RDBMS for database management. Moreover team collaboration is made easy by full integration of repositories such as Github, Bitbucket with the online shared workspaces [6].

#### 3.2 Condevy

Condevy was originally launched in 2012 and has become since then a popular cloud based development environment. Developers can work with shared, scalable and distributed workspaces, which result in better compilation and deployment times. The environments are commonly set up as sandboxes where the entire necessary configuration is prearranged so that the development process can focus primarily towards the final application. This has become since yet the common approach for Dockers exploitation, commonly done behind the scenes also from Condevy [11].

#### 3.3 Motivation

Most of the presented solutions apart from Cloud 9 and Condevy have been adopted from educational institutions for students. Some of the major cloud providers have however become aware of the fact the benefits of services offering the development environment as a whole. The authors in [15] have correctly stated the issues arising from not adopting a development environment for cloud applications from companies. Even though they claim that developers prefer the desktop environment, the inconvenience is clear when it comes to locally configuring the development environment, maintaining, updating and deploying code in real time. Moreover, most monitoring and debugging makes sense to be fully integrated with the development process as part of the core services offered from the cloud. Finally the authors state also that new models are necessary to be adopted either with existing technology and programming languages or with new emerging ones.

Next sections shall discuss how IDEaaS can act as a new service totally complementary to the previous solutions addressed so far, with all the underlying economical benefits when adopting scalable business processes in the cloud [30]. Two business models are proposed based on resource and cost management for SME-s adopting cloud services and application development as mentioned in [14] alongside the description made for the Google Cloud Computing Platform discussed from [35].

### 4. METHODOLOGY – CLOUD IDEaaS

The work also explores the possibility of adopting similar concepts for a general purpose IDE environment (encapsulating the different cloud SDK). After that this work shall provide a feasibility sample of the model, developed on the Google Cloud Platform.

#### 4.1 Cloud IDEaaS - Serving purpose and proposed models

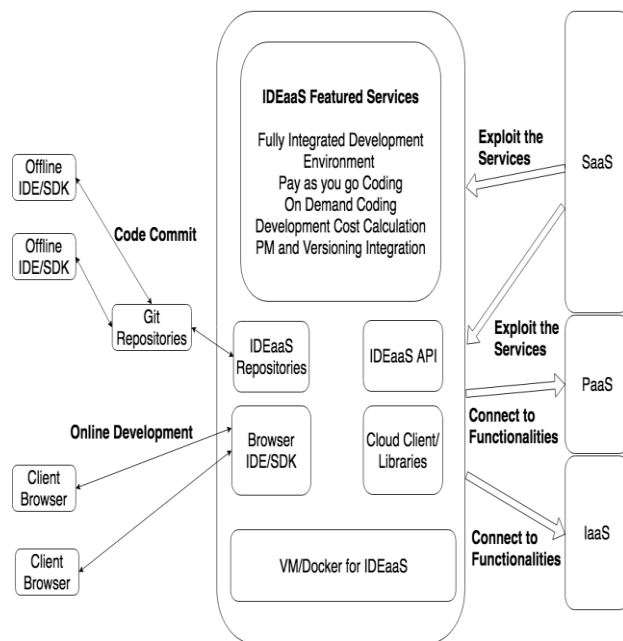
Based on the previous discussions we propose that the model should include several entities:

- Browser based SDK fully integrated with the different service layers (PaaS, IaaS) and their REST API-s or client libraries
- Browser based IDE encapsulating the SDK functionalities within the platform

–Development tool that fully exploits the pay per use or pay as you go model

–Coding Synchronization through Versioning or Agile Environments (GitHub, Jira etc.). Developers can also use other offline IDE-s with their common environment and synchronization is made from collaborative repositories

Integrating the development environment platform in the cloud and treating it as a usual service layer shall provide end cloud users not only with an online IDE, which can scale at demand, but also provide development teams with better upfront cost evaluation and help them focus on fast and efficient coding meeting project deadlines, rather than setting up beforehand collaborative environments. Community based open source coding will be facilitated even further by bringing together on one common cloud platform and adopting new coding approaches such as coding contribute on demand (Pay as you go Coding and Code on Demand) or autonomous coding techniques relying on machine learning algorithms. The IDEaaS can be adopted at business rules and corporate specific requirements related to policies, privacy and copyright protection, by intrinsically adopting public, private or hybrid cloud solutions. Figure 1 represents our proposed platform.

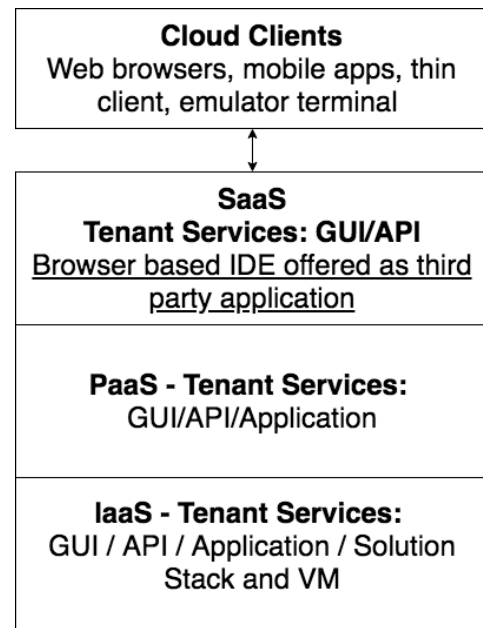


**Fig 1: Integrated Development Environment as a Service - Platform**

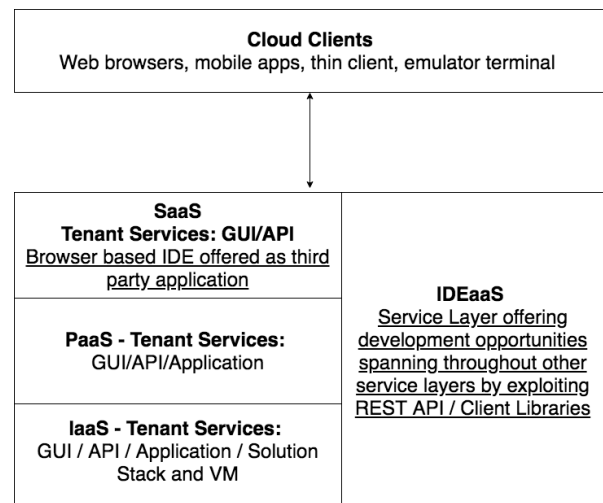
We can clearly observe the cloud layers adopting the development services provided from IDEaaS. A special focus is put on the PaaS and IaaS which clearly are the primary entities exploited within the cloud infrastructure for development purposes. Optimizing their usage have been both virtualization at VM or OS level (Docker). However, the model also supports the possibility, when applicable, to be exploited by SaaS layer where different applications can use the API provided by IDEaaS in order to integrate their own development or code testing environment. For the first time the model proposes that the cloud providers incorporate the development environment within their infrastructure; Not only making development and deployment more efficient but also bringing together cloud features into a common ground for

fast and agile software management. Figure 2 and 3 show that most cloud providers may adopt the IDEaaS service either as a third party cloud application at software layer or offered as an integral part of their infrastructure/platform. The second model would make open source contribution and development more convenient and convince more cloud based businesses to efficiently develop their products.

Other providers adopting already some similar approaches such as Amazon or the ones that haven't yet undertaken the cloud based development steps such as Azure might still benefit from the model. Proof of the concept and model adoption is presented in the following section, where actual development and integration of the IDEaaS service has been performed for one of the major cloud providers GCP (Google Cloud Platform).



**Fig 2: Browser based IDE/SDK running on Cloud Webserver VM**



**Fig 3: IDE as a Service layer incorporated into the cloud platform**

## 5. CASE STUDY - GAE LAUNCHER SDK / ONLINE IDE

The section presents the actual migration of the google cloud

python SDK-s libraries from their desktop environment to the GAE (Google App Engine). The deployment process, which was previously done from the desktop application, has been made possible to be completed from a GAE VM. Moreover, a simple code interpreter has also been developed offering the possibility to implement and link the code getting deployed directly on the cloud. GAE Launcher has also the possibility to add synchronize with local code which can be uploaded and/or part of a GitHub repository. More features can be added in the future and the development could become part of the cloud infrastructure by playing a key role and not just act as an application running on Google PaaS, which can later be part of the SaaS application pools. More functionality from gcloud tool could be part of a more sophisticated development environment fully integrated to the cloud.

## 5.1 Migrating GAE Desktop SDK to the Cloud

The migration, based on the concept stated in [29] of the Google App Engine Launcher python SDK from desktop to the GAE environment has involved the modification of several SDK libraries and their adoption with client libraries [13]. In particular the appcfg.py has been adopted to run on the GAE VM (Virtual Machine) environment. Common modifications relate to limitations of the cloud environment such as lack of file system, spawning of processes etc. Other cloud services such as Google Cloud Storage and SQL have been used instead of local filesystems. And the application has to run as a single web service process. The major changes to the code are related to modifying the appcfg.main(argv) [19,20] as below:

```
def main ( argv ) :
    logging.basicConfig ( format = "( %(asctime)s %(levelname)s %(filename)s : %(lineno)s %(message)s )" )
try :
    result = AppCfgApp ( argv ) . Run ( )
    if result :
        sys.exit ( result )
except KeyboardInterrupt :
    Status Update ( ' Interrupted . ' ) sys.exit ( 1 )
```

which is then passed a list of arguments representing the functionality to be run. Common tasks are:

1. Authentication and Deploy Arguments:
 

```
argv = [ ". / g o o g l e / a p p e n g i n e / t o o l s / a p p c f g . p y " , ' u p d a t e ' , ' / g a e l a u n c h e r / ' + ' u s e r @ g m a i l . c o m ' , " — e m a i l = " + ' u s e r @ g m a i l . c o m ' , " — p a s s i n " ]
```
2. Rollback Arguments:
 

```
argv = [ ". / g o o g l e / a p p e n g i n e / t o o l s / a p p c f g . p y " , ' r o l l b a c k ' , ' / g a e l a u n c h e r ' ]
```
3. Creating/Deleting/Editing project data/source stored on Google Cloud Storage (or git repositories). Preliminary condition is that the project should already be a registered project on the Google Cloud Console.
4. Getting logs. For each operation performed the user can visualize current logs. For this purpose the google-cloud-sdk logservice.py module has been exploited. One example displaying error logs is as follows: `tot logs =`

```
logservice.fetch(end time = end time, minimum log level = logservice.LOG LEVEL ERROR, include app logs = True)
```

The first step in accessing the application as a SaaS solution (Model presented earlier in Figure 2), was to perform the user authentication with the backend. There are two methods adopted for this purpose :

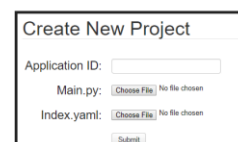
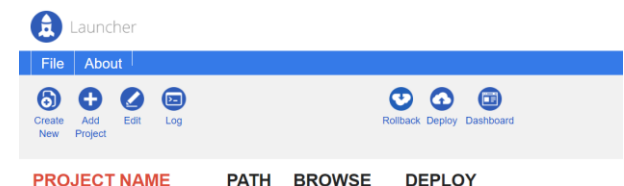
1. Exploiting the username and password for the user as part of the list of arguments upon project deployment. Drawback is that end user has to provide the credentials to the application, alongside full permission to other different services. However this is not the best approach.
2. OAuth 2 for web server authentication is a more secure approach. The reference [27] provides further information how the OAuth2 solves the problem for future references to the API. Moreover there exist full support for integrating it with Django Framework (GAE Launcher as PaaS Application). This is also the best option for future implementation of IDEaaS based on the model in Figure 3 in the previous section.

Next section discusses further opportunities along side the features offered so far for the integration. The entire code has been integrated into Django Framework 1.11 based on Python 2.7.

## 5.2 Migrating GAE Desktop SDK to the Cloud

The environment currently allows users to perform common tasks already available from the desktop GUI SDK application (Google App Engine Launcher):

Figure 5 displays some of the currently developed features. Further information can be found on <http://gae-launcher.appspot.com>.



**Fig. 5: Part of the currently developed features of the online SDK**

Development server wouldn't bring added value since the cloud virtual environment provide enough resources to handle development and production scenarios. While application staging could be of better profit. Moreover other major features have been integrated such as:

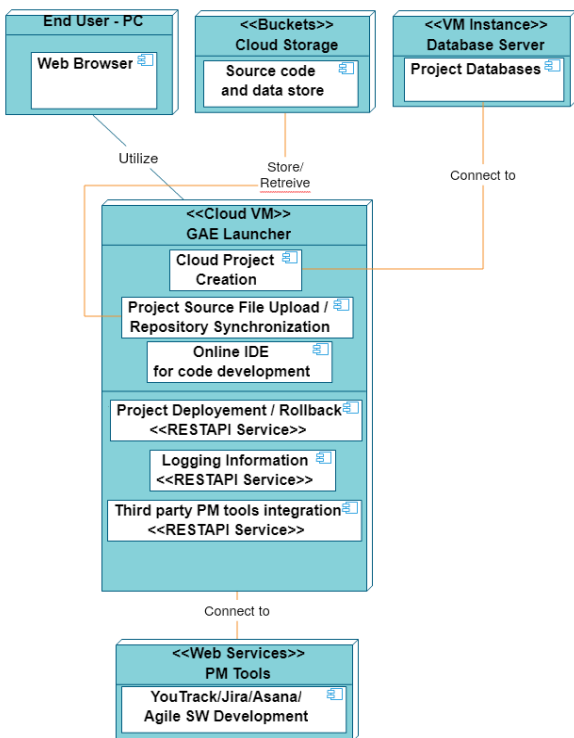
- Browser based python development environment with Django framework
- Github project synchronization

Further integration with cloud based PM tools such as Asana and Jira are part of the features that would contribute into new cloud based coding concepts such as:

- Pay as you go during project development
- Code on demand, increasing outsourcing possibilities of project functionalities
- Project cost estimation
- Optimization of collaborative and community coding for large scale projects relying on cloud platforms

### 5.3 Adopted Architecture

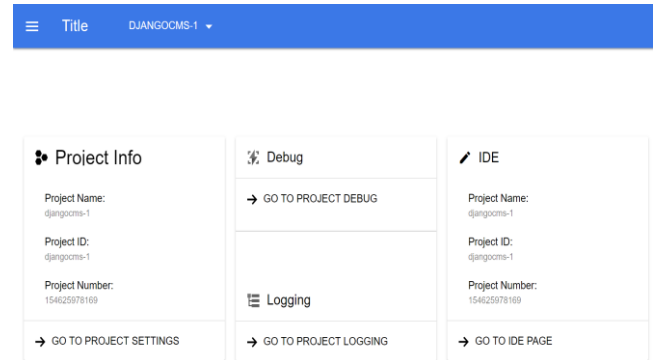
The adopted architecture, involves different entities in collaboration with the current cloud services. An overview of the integrated services is represented in Figure 6. We can clearly observe the integration of the features for efficient cloud development and deployment. Although the current architecture adopts more to the model in Figure 2, it can still be easily integrated into the current cloud console by featuring more infrastructure services such as Storage, Compute, StackDriver (logging, debugging) etc. This allows several users to exploit an integrated development platform where many features are part of the cloud but development oriented. Integration would mean adoption to the model in Figure 3, which can be part of a new core cloud service (IDEaaS).



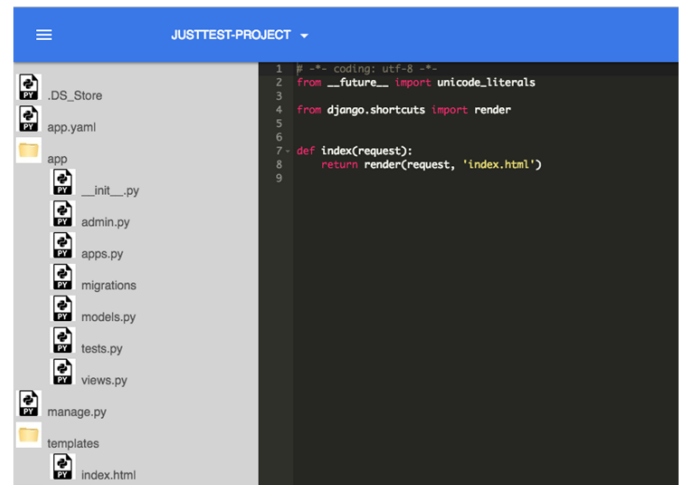
**Fig. 6: GAELauncher adopted architecture based on IDEaaS models**

To achieve so in this work it has been further developed the IDE to fit with the cloud console debugging environment and implementation pipeline (development, testing, debugging and deployment). Figure 7 and 8 represent the current state of the development.

Features offered are similar to the previously discussed SDK but further integrating the platform in Figure 1 to the Google cloud core services.



**Fig. 7: Integrating IDEaaS to the Google cloud core services**



**Fig. 8: Online IDE as part of Google cloud core services**

New concepts arise from IDEaaS, such as on demand and Pay as you go Coding, project management and version control part of the common cloud services, which can contribute also to new business models for cloud based information systems. The following section discusses such opportunities and economic models related to them.

### 5.4 Business Models for New Cloud Based IDE Services

The IDEaaS includes two major economic model opportunities:

1. Pay as you go Coding (PaygoC)
2. On Demand Coding (ODC)

Both models shall help cloud software managers optimize their PM flow, outsourcing and costs along with larger collaborating communities. While large enterprises shall never again suffer from collaborative development overheads for configuring their shared coding environments.

Model 1 ( PaygoC ) - Pay as you go is not a new business model to the cloud but when it comes to IDEaaS it can be directly adopted for coding utilizing resource hours. Thus making it easy to evaluate development costs without any infrastructure or licensing upfront costs. Moreover portable operating systems (Docker) might not be required since many of the environment setups are acquired as API services or libraries. Of course upfront configuration might be needed per project basis, but any further configurations on the online

IDE will be shared among all the project developers. Customer pricing might rely on per use basis, typically charged by the hour, of the development environment and exploitation of other existing services from the cloud console.

Key Partners	Key Activities	Value Proposition	Customer Relationships	Customer Segments
<ul style="list-style-type: none"> <li>PM online tools &amp; frameworks (Jira, ASANA etc.)</li> <li>Cloud Providers (Amazon, Azure, Google etc.)</li> <li>API, Third Party Libraries</li> </ul>	<ul style="list-style-type: none"> <li>Online Code Development &amp; Deployment</li> <li>Revenues generated from Cloud Developers</li> </ul>	<ul style="list-style-type: none"> <li>Coding Flexibility</li> <li>Real time Cooperative coding</li> <li>Faster application deployment</li> <li>Information sharing</li> <li>Improved freelancing opportunities</li> </ul>	<ul style="list-style-type: none"> <li>Cloud End User and Developer Services</li> <li>Cloud Console Platform</li> </ul>	<ul style="list-style-type: none"> <li>Freelance Developers</li> <li>Small and Medium Cloud oriented companies</li> <li>Large cloud based enterprises</li> </ul>
<b>Key Resources</b> <ul style="list-style-type: none"> <li>Developers</li> <li>Project Managers</li> <li>Cloud Applications</li> <li>Cloud Infrastructure</li> </ul>		<b>Channels</b> <ul style="list-style-type: none"> <li>Common cloud customers</li> <li>Professional oriented social networks</li> </ul>		
<b>Cost Structure</b> <ul style="list-style-type: none"> <li>Development and Maintenance costs</li> <li>Integration with third party services</li> <li>Contractual agreements</li> <li>Dedicated and shared cloud infrastructure costs</li> </ul>		<b>Revenue Streams</b> <ul style="list-style-type: none"> <li>Application Development based on PM and Developers online payments</li> <li>High revenues based on the number of projects developed</li> </ul>		

Fig. 9: PaygoC and ODC business model canvas

Model 2 (ODC) - On Demand Coding permits outsourcing services to be facilitated and optimized whenever coding expertise is needed on demand. The major opportunities that might be risen from the model could integrate employment platforms with cloud development environments. Making it possible to develop large open source development cloud based communities fully integrated with existing freelancing platforms. Cloud based outsourcing would definitively provide better security, development policies, project cost evaluation and avoid over budgeting. Figure 9 describes the PaygoC and ODC business model canvas. It is clearly observed from the proposed business model the different potentials arising from integrating several new actors in the cloud development in one common place.

Statistics also prove that cloud usage and net income is increasing with AWS being at the top list while the others follow such as Azure, Google, IBM etc. Opportunities arise for this business model to be successful in the market. A good comparison, Figure 10 represents Google and AWS cloud earnings taken from Statista [32]. From the graphics it is clearly observed that AWS [3] is way ahead w.r.t. GCP (Google Compute Platform) in earnings. However they both prove they have been keeping a solid growth rate during the previous fiscal year.

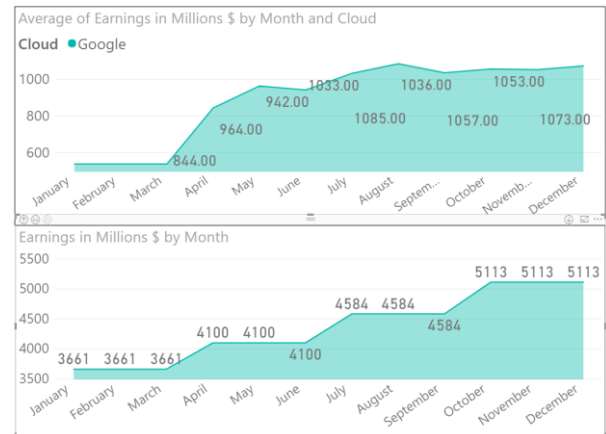


Fig. 10: Google and AWS cloud earnings taken from Statista [32]

Figure 11 represents estimates of IDEaaS contribution to the current growing trends for cloud development. The estimates have been based on a community of approximately 2 Million cloud developers (accounting for 10% of global developers), with a daily overload of 4.2 hours. The final earnings for three different charges of 0.5, 1 and 1.5\$/h to account up to 0.1, 0.2 and 0.3 Billion estimated yearly income increase. Proving the potentials of IDEaaS to consolidate itself as a stable business model for cloud users.

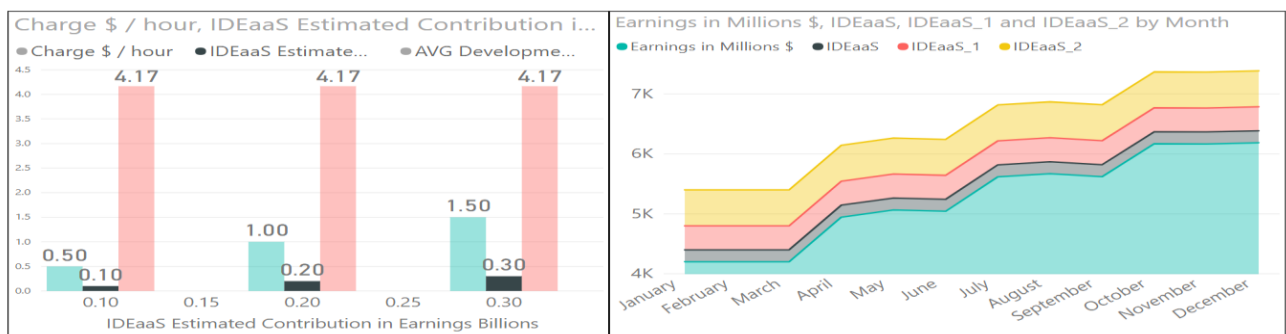


Fig. 11: IDEaaS contribution to the annual earnings for Google Cloud referring to Statista

The section has proven both the viability of the model as well as the possibility to become a key actor in cloud computing services. Next section discusses further opportunities arising from the proposed model.

## 6. DISCUSSIONS AND CONCLUSIONS

This paper work proposes a new cloud service which will have important impact how cloud development will be perceived in the near future.

Most of the Cloud SDK-s today for popular programming languages are becoming hard to manage and somehow

redundant with the growing needs of programmers for more flexible and cooperative environments.

Having new business models adopted and new development services added to the cloud infrastructures shall provide increased growth of interest in exploiting cloud applications at enterprise level.

The new IDEaaS service model suggest that the development environment could be offered as other services currently are within the cloud. This leads to new business model opportunities where coding can be demanded and outsourced at any time, making it easier for project managers for the first

time to convert actual coding hours into development costs. All this done without the need of third party tools.

Major cloud providers are becoming aware of the developer needs and already supporting online development tools, although most of them are still relying on old-fashioned downloadable SDK-s.

The major benefit of the IDEaaS, like any other service, it can provide new features based on cloud end user demands as well as generate solid revenues for every provider adopting the business models relying on it.

The present solution proposed in this paper GAE Launcher provides all the features already offered from the Desktop SDK application but integrating further new services which have already been offered online for a long time (Git repositories, PM tools etc.). Offline provisioning and online synchronization solution as well as integration with other existing tools, by means of plugins or add-ons, should become part of the core functionalities of the service, so that developers would still feel comfortable continuing their work even when online connection is lost.

Future trends will prove the effectiveness and allow more cloud providers to adopt the proposed platform and models, where machine learning and artificial intelligence will ease the cloud application development process.

## 7. ACKNOWLEDGMENTS

Our thanks to the DOMINUSOFT developers who have contributed towards development of the cloud service.

## 8. REFERENCES

- [1] A. van Deursen, A. Mesbah, B. Cornelissen, A. Zaidman, M. Pinzger, and A. Guzzi, 2010, Adinda: a knowledgeable, browser-based IDE; In ICSE, pp.203-206.
- [2] A. Zeller, 2007, The future of programming environments: Integration, synergy, and assistance
- [3] Amazon Web Services: [https://en.wikipedia.org/wiki/Amazon\\_Web\\_Services](https://en.wikipedia.org/wiki/Amazon_Web_Services)
- [4] BESPIN. Bepin: <https://mozillalabs.com/skywriter/2010/10/01/bespin-0-9a2-released-skywriter-update/>
- [5] CA. Codeanywhere <https://codeanywhere.net/>
- [6] CLOUD9. Cloud9: <https://aws.amazon.com/cloud9/>
- [7] CLOUDFORGE. CloudForge <http://cloudforge.com>
- [8] CODRERUN. CodeRun: <http://www.coderun.com/ide/>
- [9] CODEPLEX. CodePlex <https://www.codeplex.com/>
- [10] COMPILR. Compilr <http://compilr.com>
- [11] Condevy: <https://codenvy.com/>
- [12] CREATELY. Creately <http://creately.com>
- [13] Django Support: <https://developers.google.com/api-client-library/python/guide/django>
- [14] E. Kuada, K. Adanu and H. Olesen, "Cloud computing and information technology resource cost management for SMEs," Eurocon 2013, Zagreb, 2013, pp. 258-266.
- [15] Fylaktopoulos G, Goumas G, Skolarikis M, Sotiropoulos A, Maglogiannis I. An overview of platforms for cloud based development. SpringerPlus. 2016;5:38.
- [16] GAE Launcher as PaaS Application : <http://gaelauncher.appspot.com/>
- [17] GIT. Git <https://git-scm.com>
- [18] GITLAB. GitLab <https://about.gitlab.com>
- [19] Google Cloud Python API : <https://cloud.google.com/resource-manager/reference/rest/v1/projects/create>
- [20] Google Cloud Tools: <https://cloud.google.com/docs/overview/developer-and-admin-tools>
- [21] Jam J, Brannock E, Dekhane S (2010) JavaWIDE: innovation in an online IDE: tutorial presentation. J Comput Sci Coll 25(5):102–104
- [22] JAVA WIDE: <http://www.javawide.org>
- [23] J. Hausladen, B. Pohn and M. Horauer, 2014, A cloud-based integrated development environment for embedded systems, IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA), Senigallia, pp. 1-5.
- [24] JSFIDDLE. jsFiddle <https://jsfiddle.net>
- [25] L. Wu, G. Liang, S. Kui and Q. Wang, 2011, CEclipse: An Online IDE for Programming in the Cloud, 2011 IEEE World Congress on Services, Washington, DC, pp. 45-52.
- [26] Microsoft Azure : [https://en.wikipedia.org/wiki/Microsoft\\_Azure](https://en.wikipedia.org/wiki/Microsoft_Azure)
- [27] OAuth2 Python API: <https://developers.google.com/api-client-library/python/auth/web-app>
- [28] R. Frost, 2007, Jazz and the eclipse way of collaboration, Ieee Software, pp. 114-117
- [29] REMICS (2010) REuse and Migration of legacy applications to Interoperable Cloud Services. <http://www.remics.eu>
- [30] Schmidt R (2012) Scalable business process enactment in cloud environments. Enterprise, business-process and information systems modeling. Springer, Berlin Heidelberg, pp 1–15
- [31] STACKDRIVER. StackDriver: <https://cloud.google.com/debugger/>
- [32] STATISTA. Statista: <https://www.statista.com/statistics/250520/forecast-of-amazon-web-services-revenue/>
- [33] Voorsluys W, Broberg J, Buyya R (2011) Introduction to cloud computing. In: Buyya R, Broberg
- [34] J, Goscinski A (eds) Cloud computing: principles and paradigms. Wiley Press, New York
- [35] X. Jia, "Google Cloud Computing Platform Technology Architecture and the Impact of Its Cost," 2010 Second World Congress on Software Engineering, Wuhan, 2010, pp. 17-20.