Position Paper: OPENi – Future of a Consumer-centric Cloud-based Application Platform

Robert Kleinfeld and Lukasz Radziwonowicz
Fraunhofer FOKUS
Berlin, Germany
{robert.kleinfeld, lukasz.radziwonowicz}@fokus.fraunhofer.de

Leigh Griffin and Eric Robson
TSSG, Waterford Institute of Technology
Waterford, Ireland
{lgriffin,erobson}@tssg.org

Fenareti Lampathaki
National Technical University of Athens (NTUA)
Athens, Greece
flamp@epu.ntua.org

ABSTRACT

Today, over the Web there is a high availability of cloud-based services which, through their publicly available APIs, disrupt traditional industries and provide a vast number of features indicatively spanning: location-based services, photo sharing, video & music streaming, recommendations, data storage, data syncing, and social networking. This prolific array of services is fuelled by an insatiable market driven by modern European consumers constantly striving for innovative tools and services to support their personal and business needs.

The OPENi project will define and deliver an open-source Web platform that will enable mobile application consumers to store data and metadata from their mobile application usage in their own space in the cloud – the "Cloudlet". This information can be shared (by and under the control of the consumer) securely among their applications, services and across their connected devices. To realize this vision OPENi will provide the required components that will support both consumers in creating and managing their cloudlets but also Developers in creating applications that will be able to access and interact with them without disrupting service providers.

In this paper, the OPENi cloudlet concept is outlined and the architectural blueprints are described leading to a discussion about the lessons learnt and next steps.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures – *Domain-specific architectures*
H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based services*

General Terms

User Centric: An approach based on (end user) use cases, personas, their needs, goals, knowledge and experience. This includes analysis, design, development, testing and evaluation.

Service Enabler: Building block to enable the development of higher order services while not being an end user service itself.

HotTopiCS'13, April 20–21, 2013, Prague, Czech Republic.

Cloud Platform: The term cloud platform is used to refer to any computational or storage cloud. Cloud is used as a synonym.

Service Provider: Service providers include online service providers and traditional transaction based services, offer applications, tools and facilities to consumers with the goal of maximising their number of customers while reducing operational expenditure.

Consumer: The end user who consumes a service at a cost. The cost is dictated by the provider and may be tangible (money based) or intangible advertising / marketing driven.

Service: A service can be a standalone application or a tool to assist another service or application.

Keywords

Cloud, Personal Cloud, Open Standard, Security, Privacy, Mobile Services, API Framework

1. INTRODUCTION

In today's society, accessing cloud-based services, such as YouTube, Spotify, Facebook, TripAdvisor, is predominantly through applications on mobile devices. Through everyday usage, consumers of these services provide these services providers with an immense amount of content and personal data. However, due to the lack of consumer-controlled cloud storage infrastructures and the siloed nature of cloud-based services, consumers are deprived of any control over their content or data [1]. Furthermore, they are unable to make this data available to multiple applications across various services and devices – something which has led to a significantly impaired application usage experience and stagnation in application innovation.

OPENi aims to create a consumer-centric, Open Source mobile cloud applications solution that will be the catalyst for mobile application innovation advancement. The platform will incorporate an open framework that will be capable of interoperating with any cloud-based service, abstracting the integration challenges to a single open standard without losing any service features. It will be a single platform that will inherently promote innovation by offering application developers an advanced framework in which to design and build complex applications involving the combinations of independent cloud-based services. OPENi will be a significant catalysing factor, boosting European Union (EU) software industry – which already counts a considerable number of highly skilled Small Medium Enterprise (SME) software firms to take further advantage of the

thriving market of mobile applications, through a framework of open technologies that will be readily available and require almost no additional skill to take up.

The remainder of this paper is organized as follows. Section 2 details the background and related work in user-oriented cloud solutions. Section 3 introduces the OPENi cloudlet concept. Section 4 details key features of the platform. Section 5 provides an outline of the foreseen architecture. Finally, section 6 presents the next steps through pilot use cases which will be built on the OPENi platform capabilities.

2. BACKGROUND AND RELATED WORK

The analyst in Forrester Research, Frank Gillet describes personal cloud becoming the third client software platform, following mobile devices and PC operating systems. He also predicts that the market is expected to grow from $500 million to $6 billion in direct revenue by 2016 [2]. However, asking for a definition of what defines a personal cloud will invariably result in a different response, most giving a definition of some method of data storage. Tian, Song & Huh [3] break the definition of the personal cloudlet into three categories as follows;

- Online storage; gives users a place to store information that is available to us wherever there is an Internet connection,
- Web-based Applications; software as a service,
- Webtop; recreating the highly personalized setting of our own desktop with an online version.

Research in this area has also loosely followed this split with supporting research into the required security aspects of them [4]. Cloud computing is also being investigated in the area of pervasive computing with a focus on ways to implement context and personalisation to the user. However, this simply targets the storage capacity the cloud can provide that is not feasible with all pervasive devices currently.

Furthermore OPENi can be considered in certain aspects as an application platform which addresses a runtime environment for applications. This runtime environment will have the same appearance to applications on all hardware platforms (already conducted by FP7 project webinos [5]) and thus abstract the underlying host system. The virtualization of application execution enables differentiated handling of application life-cycle and security aspects, as shown by legacy technologies like .NET and Java before [6]. The OPENi framework is tailored to applications with rich presentations, i.e. especially addresses graphical presentation, multimedia processing and high responsiveness to user actions. Different from comparable solutions like AdobeFlash/Flex Player, MS Silverlight or SUN Java FX runtime, the OPENi framework explicitly addresses cloud solutions. Cloud solutions can make use of/integrate all open data and services available on the open Internet. Also they can expose services and capabilities of the hosting devices as cloud services.

A key aspect is the presentation of application data and executable code in readable, non-obfuscated languages to enable convenient meta-processing (indexing and searching) as well as integration with applications of 3rd parties. For this reason OPENi will base on the open, standardized language and protocol stack of World Wide Web Consortium (W3C) and the latest webinos project research findings. Web hypertext application technology uses an application model tailored to the least common set of host device capabilities. OPENi will be deployed basically on mobile devices of different platforms, in order to ensure openness of the framework.

Summarising the above, it can be said that, OPENi will be the framework to create the third client software platform as suggested by Gillet [2]. Furthermore OPENi will have the ability to encompass the digital personal identity. Digital personal identity will cover health, finance, government etc. personal records owned and managed by the user in a central location the “Cloudlet”. OPENi intends the personal cloud topology to become the standard for mobile cloud computing creating a scalable, extensible framework to cater for developers of apps, social media add-ons and also enterprise level applications to add storage for functionality that they require. Finally, OPENi moves forward the state of the art by developing an Open Source cloud platform that can enable the instantiation of user spaces in the cloud (Cloudlets, the concept will be detailed in the next section), with capabilities such as storage, discoverability, addressability, access, security across applications and devices, which up to today can only be found in proprietary environments.

3. OPENi CLOUDLET CONCEPT

A central concept to OPENi is to reduce the fragmentation and duplication of citizens’ data. OPENi will provide application users with a single location to store and control their personal data. Therefore OPENi introduces the cloudlet concept. This personal cloudlet will enable consumers to manage what information is available to each application and for what purpose. It will be a single federative source for their personal data and content. The cloudlet will empower application consumers to remain in control of their data – it will be inherently secure and trustworthy and as an open technology, validated by the open source community, consumers will be assured their data is not being used without their consent. Such a cloudlet is supported by an APIs framework that ensures seamless communication with cloud-based services (ranging from popular social networks and photo / video sharing services to advertising services) and mobile apps that exploit its functionalities. The combination of the open APIs (to access cloud-based services' and cloudlets' functionality) and cloudlet concept creates a single platform of user data and service connectivity making OPENi a very powerful and beneficial platform for consumers and application developers [7]. Thus, OPENi envisions the concept of a cloudlet for use by both the consumer and application developer.

3.1 Consumer

A cloudlet is a virtual representation of an individual user’s personal space in the cloud, storing personal data and metadata about the user and applications they use. The customisable nature of the cloudlet allows the end user control what content is to be made available to other cloud-based services and providers. The user is primarily a mobile consumer interested in accessing and consuming content served within cloud computing held infrastructure.

3.2 Developer

A cloudlet is a facilitating tool enabling integrated APIs for the management of their users cloud storage and preferences. Developer cloudlets can access shared metadata in their users' cloudlet space, enabling pattern discovery across their application suite for analytics, quality of service monitoring and the identification of potential new user focused applications. The cloudlet possesses the following attributes and characteristics:

- A storage mechanism, on a user defined addressable space, for the storage of personal data and metadata of an individual user
- A mechanism for enabling the sharing of information between applications, services and devices by operating as a middle-man service
- Configurable by the user to ensure control over the access granted and the information that is available to other cloudlets and applications
- Interoperable with major service providers through API chains
- Capable of offering a historic perspective on users interactions through the cloudlet

In OPENi a server side cloudlet platform will exist to manage individual cloudlets owned by users. The cloudlet platform will act as a mechanism to enable an application to access a users' cloudlet and if necessary facilitate the user in the creation of a cloudlet. The platform can be deployed by both providers and large scale developers allowing them to manage a cloudlet infrastructure. This feature set will be realised through a User Interface (UI) and the hosted cloudlets are made available to external applications through discovery mechanisms and APIs. The UI will deal primarily with the management of the cloudlet and the configuration of access as required by the user. This access will be facilitated by best practices from the point of view of access control. The API layer will govern discoverability, access by applications and storage mechanisms for private data. This feature set combined will offer a lightweight, scalable and customisable access management to a user's cloudlet.

4. OPENi KEY FEATURES

Based on the introduced cloudlet concept this section will describe the key features of the OPENi platform. This includes not only software features but also rather strategies to maximize the impact of the OPENi platform.

4.1 Web APIs Framework

For the developer, OPENi will deliver a set of generic Web APIs that will abstract the functionality of existing (publicly available) cloud-based services APIs, hiding the implementation complexity of the backend [8]. OPENi will cater for providing a broad spectrum of functionality by defining generic APIs that combine cloud-based functionalities in a set of functionalities that are similar to Phone SDKs, and existing phone applications or functionalities. Such APIs indicatively cover social, payment, location based, data storage and syncing, media streaming and ad-serving cloud-based services which are selected on the basis of criteria like their added value to a developer, their usefulness to the consumer and their popularity (by the number of mash-ups utilizing it and the number of users it has attracted).. For example a *searchPhoto* method in the OPENi Media API will retrieve photos either from Flickr, Instagram and/or other providers according to the users' preferences. Through the OPENi API framework developers will be able to easily design and integrate existing cloud-based features into their applications, focusing primarily on what they require in order to enhance their applications user experience and not caring about how the implementation parameters are met. This approach will reduce fragmentation, development and maintenance effort and time to deliver through a purely Web compatible, platform-independent framework.

4.2 Service Enablers

One of the major benefits of the OPENi solution is that it becomes a central tenant of the relationship between a consumer and their services. By providing service enablers, the platform will allow the consumer, through the application developer, to unlock the potential of combining contextual information from one application with another.

OPENi defines service enablers as broker between service provider and application developer by including a set of generic Web APIs and complementary components like context awareness, inter-app communication, security and authorisation, data management, personalisation and multi-device UI rendering. From the perspective of an application developer a location based service for recommendations of places nearby can be enriched with a component for user authorisation and identification. So the final application supports several authorisation methods based on preferences of the end-user (e.g. authorisation via Facebook Connect, OpenID, Twitter, etc.) [9]. To sum up this OPENi service enabler offers the possibility to combine cloud-based service APIs with components for comprehensive Web and device features.

The service enablers will also benefit the service providers; they will be sufficiently flexible enough to protect their core business models providing them with potentially new revenue generating streams. Through the platform the service providers will be able to launch novel services that promote their business without losing control about how the service is used. For example, if a particular provider depends on an advertising business model, they can build the advertising details in to their Web service and reject the service if certain criteria are not met; ensuring that the service enablers, application developers and ultimately the application consumers adhere to their conditions. The enablers to be delivered will include: context brokering, security and authorisation, service failover and multi-device UI rendering.

4.3 Cloud Platform

For the consumers, OPENi will deliver a cloud platform that will allow them to create, deploy and manage their personal space in the cloud (the before mentioned cloudlet). It will enable their applications to access, store and update user data and content (i.e. profile, location, social data) according to their preferences. For example a user accesses a news reading application that internally uses an OPENi *statusUpdate* API to post the articles they are reading to their preferred social network. The user will be able to opt that this information is also stored in their cloudlet and then made available to any other applications they wish (i.e. a bookstore applications that could use it to give him recommendations for new book titles based on the articles he was reading). OPENi will create all the necessary cloud components to allow the users to create, deploy and manage their cloudlets, providing qualities such as data storage, discoverability-addressability-access by applications and a user controlled privacy and security framework.

4.4 Open Web and Cloud Technologies

OPENi will develop a solution that reduces the technical barriers for wide-spread adoption by developing the platform using Web technology skills readily available to the vast majority of software developers. Web technologies and standards, such as HTML5, CSS and JavaScript present a significantly higher degree of platform-independence, portability and interoperability across different operating systems and devices in relation to all other

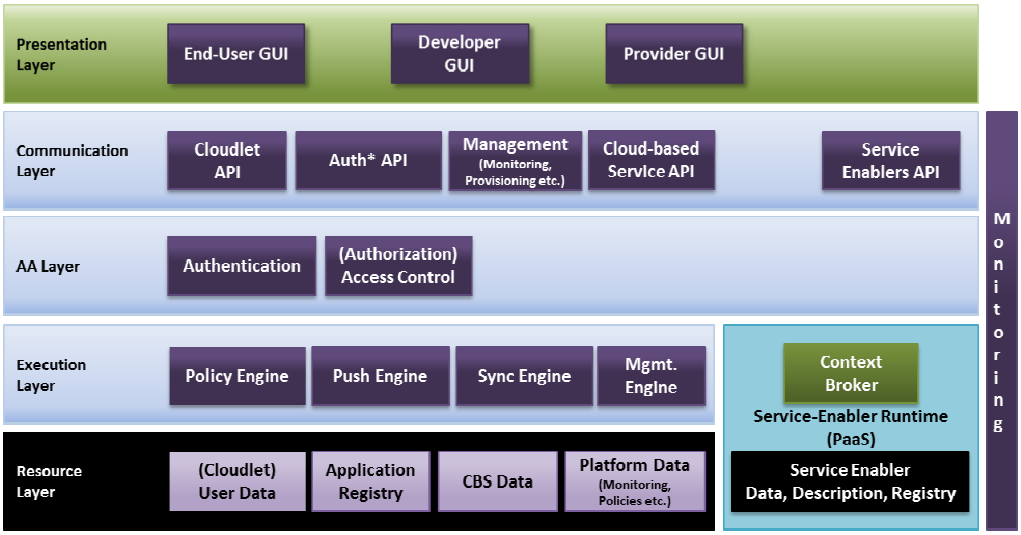

Figure 1: OPENi Platform Layers

"native code" technologies (i.e. Objective-C, AdobeFlash, Java, etc.). In addition they have a significantly higher adoption footprint in relation to any other native platforms and already have established thriving communities. Furthermore, many development frameworks (PhoneGap and Titanium) and standards based efforts (W3C DAP WG specification) adopt Web technologies as the development platform for mobile applications [10].

Cloud technologies have reached today a level of maturity to be used in creating novel consumer-centric cloud solutions. Open Source cloud stacks (Eucalyptus [11] and OpenStack [12]) provide the necessary capabilities to design, implement and deploy end-to-end cloud solutions and the availability of online storage is all the more becoming affordable. Therefore it is possible today to design and deploy a solution that can enable users to manage their cloud assets (data and services).

4.5 Open Source Model

In order for OPENi to maximise its impact, the technology needs to have the potential to lead to new market opportunities, within a viable time period (time-to-market). To ensure this for each potential adopter OPENi will adopt and extend the Open Source model across all its deliveries. This will come with an equally open governance schema – i.e. Apache, MIT licenses that will enable potential adopters to evolve the project deliveries to potential products and business offering, endowing yet with sufficient time and leverage the project partners in order to have sufficient exploitation of the project results.

Additionally through free access, Open Source licensing not only reduces the barrier to adoption, but it also fosters a community of interested parties in the project. This community, though their use, interaction and extension of the platform will ensure the platform remains secure, reliable, extendable and flexible.

5. OPENi ARCHITECTURE

OPENi comprises two basic platforms, the API framework and the cloudlet platform. The following section presents the basic architectural components in each one.

The purpose of the cloudlet platform technology is to be readily usable by any cloud provider in order to establish his new user cloudlet business offering. OPENi will ensure such a minimum time-to-market for the delivered technology. Therefore OPENi will define and implement from scratch only the components that provide novel functionality and added value with respect to the cloudlet concept. Furthermore OPENi will reuse existing components of open cloud stacks with established and widely accepted functionality.

As cloudlets contain personal user data, security and privacy is an important issue. However, third party applications or service providers, such as the advertisement service, need to access information in a cloudlet. OPENi should grant that cloudlet owner the ability to decide which information can be retrieved by third parties and which not. This will be solved via access policies or similar access control mechanisms. Cloudlet owners should also be able to define different access policies for different kinds of applications or services accessing the cloudlet information.

Figure 1 introduces the architecture of the OPENi platform. The architecture is divided in five layers: Resource, Execution, Authentication and Authorisation (AA), Communication and Presentation. Each of the layers is monitored to identify malicious intents and provision applications. The service enabler is of particular importance. While respecting the AA layer, service enablers access the resource and execution layer directly. In the following subsections role and features of each layer will be described in detail.

5.1 Resource Layer

The resource layer stores the different forms of data that is used by the OPENi platform. The resources are accessed by the service enabler runtime, the execution layer components and via the public APIs. In addition, the resource layer also encompasses the virtual machine resources, which are used to run the execution layer. They are omitted from Figure 1 due to space restrictions.

Cloudlet (User Data): The Cloudlet contains personal user data. This information must be stored securely, privately and the access management should solely rely in the hands of the user. Services can add, remove or alter the cloudlet data based on user

defined permissions. If a service alters the cloudlet, the changes are transparent to the user and other permitted services at any time.

Application Registry: Applications need to be registered with the OPENi platform. After the registration OPENi providers are able to manage applications e.g. block them or define quotas for them.

Platform Data: The platform stores data as part of the configuration and management process. This includes policies and monitoring data.

Cloud-based service (CBS) Data: The data is stored by the platform as part of the communication process with cloud-based services. For example the Cloudlet Social API is connected to the cloud-based service "Facebook". OPENi will be registered as a Facebook application to access the data. Such a registered application needs to store and manage tokens and *IDs* to access Facebook.

5.2 Execution Layer

The execution layer engines are executed on the cloud computing resources of the OPENi platform. The engines are accessed via the public APIs by presentation layer components. The execution layer accesses the resource layer components.

Policy Engine: The Policy Engine enforces the registered policies. It includes the policy enforcement and decision point. The policies are registered by users as part of their data access control or the providers as part of the service provisioning and management.

Push Engine: The push engine notifies users and applications about changes to the user's resources by other applications or cloud-based services.

Sync Engine: The engine synchronizes the data and references between the OPENi platform and cloud-based service resources.

Management Engine: The engine provides provisioning and monitoring applications and cloud-based services. It analyses access logs to determine malicious applications and manages the applications via provisioning quotas.

5.3 Authentication and Authorisation (AA) Layer

The AA layer governs access from the communication layer to the lower layers. Applications and users need to be authenticated and authorized for such access to be granted.

Authorization (Access Control): Access to the underlying resources is restricted by the access control layer. Users are able to restrict the access to their cloudlets. Services are restricted from accessing resources if users have not granted the appropriate permissions. The OPENi provider uses the access layer to manage the permissions of administrative accounts and services. The access control is an abstraction layer to the policy engine.

Authentication: Authentication is the process of validating user credentials. The OPENi platform provides a single sign-on login mechanism. The component checks the user credentials and if the user exists and its credentials are valid, access is granted. Applications gain access to the OPENi platform and use token-based credentials to access personal data on behalf of a user.

5.4 Communication Layer

The communication layer exposes the functionality of the lower layers via public APIs to application developers. Users and providers are able to access the communication layer via the presentation layers web GUIs.

Cloudlet API: The Cloudlet API provides access to all the user data stored within the cloudlet. This includes personal data generated by service enablers, applications or cloud-based services. The API consists of various sub-APIs, which are envisaged to provide the following functionality:

1. Discovery API to allow an application to discover a particular user cloudlet
2. Query API to allow an application to query a cloudlet about its stored data and the schemata
3. StoreData API to allow an application to store data and content into a cloudlet
4. GetData API to allow an application to retrieve stored data from a cloudlet.

The APIs abstract and unify the different cloud-based service classes. These include Payments and Goods, Ad Networks and Analytics, Articles and News, Data Storage and Synchronization, Social Networks, Location Based Services, Photo Sharing, Video Streaming and Music Streaming. Each API (i.e. Media API) defines a number of Interfaces that provide several generic methods (i.e. *searchPhoto* and *getPhoto* methods). To use an API an application makes a call to one of the defined methods i.e. *OPENi.Media.searchPhoto*. Internally, each API implements generic method calls to specific cloud-based service methods in order to deliver the necessary functionality or content (i.e. the generic *searchPhoto* implements calls to search methods of the Flickr, Intsagram, Picasa APIs, etc.). References to the data or content itself can be stored in the cloudlet. Only the content which has been added (e.g. imported) by the user into the cloudlet can be accessed by the applications.

Authentication and Authorization API: The APIs are used to facilitate the authentication, authorization and permission management of users, applications and providers.

Service Enabler API: The service enabler API allows applications to access the service enabler runtime and discover service enablers, their APIs and their schemata.

Management APIs: The API provides access to the management engine. It is used to provision and monitor applications, cloud-based services and service enablers.

5.5 Presentation Layer

The presentation layer provides web-based Graphical User Interfaces (GUIs) to users, developers and providers. The GUIs access the resources and engines via the communication layer. All GUIs are able to access and view logs and statistics.

End-User GUI: The GUI allows end-users to view and manage their data in the cloudlet. Users can also view their connected applications and their required permissions. Users can alter all his data or the permissions of the connected applications at any time.

Developer GUI: Developers are able to register and unregister applications with the platform. They are able to alter the needed application permissions. The platform provides developer services e.g. automated error reports.

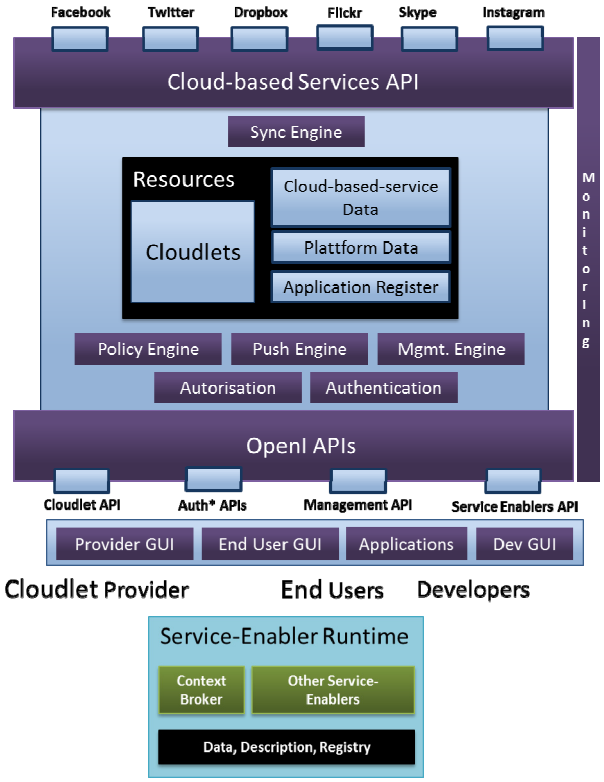

Figure 2: OPENi Architecture

Provider GUI: The OPENi provider can manage the platform, including the infrastructure and applications as needed. The platform management allows adding, removing and configuring virtual resources. Malicious applications can be removed and denied access.

5.6 Service Enabler Engine

The service enabler engine is a Platform as a Service (PaaS) runtime. It contains a resource layer and an execution layer. The engine can be used to create new service enablers that consume, create and alter data within the cloudlet. They may provide aggregates, filters or projections and expose them to other services.

Context Broker: The context broker is an example for a service enabler. It enables the sharing of context information between applications and may perform additional computations on them.

Service Enabler Data: Is non-user specific data which is generated by service enablers. This includes configuration data and service enabler states.

Service Enabler Registry Data: The service discovery component provides a way to register, unregister or query for service enablers and their descriptions.

Service Enabler Description: In order to use the service enablers, their APIs, restrictions and licenses need to be documented. The API documentation includes the communication protocols to access the service and receive responses.

Figure 2 provides an architectural overview. The cloud-based services are connected to the OPENi platform via the internal cloud-based service API. The cloud-based service resources are kept in sync with the OPENi platform via the sync engine. The resources are accessed by applications and web GUIs via the public OPENi APIs and by the internal engines of the execution layer. The service enabler runtime provides a PaaS cloud computing infrastructure and storage capabilities to deploy new services enablers to the OPENi platform.

6. CONCLUSION AND NEXT STEPS

This section details the innovative use cases that OPENi will implement as a showcase of the platform. Each use case is designed to pilot various technical aspects and features of the platform while ensuring they are relevant to the end users of the OPENi platform. In particular: i) the "MyLife" use case pilots a sample scenario that directly impacts on the consumer; ii) while the "personal advertising" use case pilots how the traditional revenue stream of online service providers are protected; iii) the final use case for a "personal shopping" assistant validates OPENi's applicability to the traditional bricks-and-mortar retail industry.

The first pilot will focus on time-line view of the various events and transactions that occur on a daily basis. The pilot will expand on the recent Facebook Timeline concept and build a solution that will consolidate this view to include any (time indexed) information from any cloud-based service. It will include additional intelligence to categorise and cluster similar events, making the user presentation more intuitive to the consumer's needs.

The second pilot will focus on personalized ad services via multiple channels in an opt-in and anonymised basis. The pilot can make use of the enhanced security and controlled data access of the user's data in order to anonymously and non-invasively provide context based personalized advertising (pull model) and marketing (push model) where the users can be encouraged to opt in to a marketing program via incentives like offers, coupons etc. Access to personalized and behavioural data across applications can also provide an additional value to in-app advertising creating more relevant and targeted ads without violating the privacy of the end user.

The third pilot will focus on heavily personalised in-store shopping experiences. This pilot will validate that OPENi can enable the delivery of personalised shopping experiences on a mobile device through demonstrations of working prototype apps in a virtual in-store environment. The pilot clearly describes how information from many different social networks, app and services can be combined to power a service that could not exist without that information.

7. REFERENCES

[1] G. von Laszewski, J. Diaz, F. Wang, and G. Fox, "Comparison of multiple cloud frameworks," in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, pp. 734-741, june 2012.

[2] The Personal Cloud: Transforming Personal Computing, Mobile, And Web Markets Air Date: Tuesday, August 02, 2011.

[3] Michael Brock and Andrzej Goscinski, Toward a Framework for Cloud Security, Algorithms and Architectures for Parallel Processing, Volume 6082/2010, pp. 254-263.

[4] F. Wu, \Presence Technology with its Security and Privacy Implications," in Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on, pp. 1 {6, june 2007.

[5] ICT Collaborative Project webinos (Secure WebOS Application Delivery Environment), grant agreement n. FP7-257103. http://webinos.org.

[6] J. Gosling, B. Joy, G. L. Steele, and G. Bracha, The Java Language Specification. Upper Saddle River, NJ: Addison-Wesley, 3 ed., 2005.

[7] Blanchette, J. (2008) The Little Manual of API Design. Jasmin Trolltech, a Nokia company. Retreived on November 13th, 2012 from: http://chaos.troll.no/~shausman/api-design/api-design.pdf

[8] Vermorel, J. (2010) A few tips for Web API design. Retrieved on November 13th, 2012 from: http://vermorel.com/journal/2010/12/22/a-few-tips-for-web-api-design.html

[9] Q. K. A. Mirza, “Restful Implementation of Authorization Mechanisms," in International Conference on Technology and Business Management, 2011.

[10] Appcelerator / IDC Q4 2012 Mobile Developer Report. Voice of the Next-Generation Mobile Developer. Retrieved on November 13th, 2012 from http://www.appcelerator.com/thinkmobile/surveys

[11] http://open.eucalyptus.com/

[12] http://openstack.org/