

Personal Cloudlets: Implementing a User-Centric Data Store with Privacy Aware Access Control for Cloud-based Data Platforms

Nauman Birwadkar¹, Mrs. M.P. Deshmukh¹, Apoorva Arbooj¹, Suraj Todkari¹, Priya Agarwal¹

Department of Information Technology, JSPM's Rajarshi Shahu College of Engineering,
Savitribai Phule Pune University, Pune¹

Abstract: OPENi's Personal Cloudlets framework as a novel approach to enhancing users control and privacy over their data on a data driven, cloud-based platform. The OPENi concept and the research objectives that influenced the design and implementation of OPENi's Personal Cloudlet Framework. We outline OPENi's architecture and describe how through the use of REST based endpoints, object-based access control, OPENiTypes, and stateless JSON Web Token (JWT) it allows users share, reuse, and control access to their data across many mobile applications while maintaining cloud scalability. Furthermore how a number of the Personal Cloudlet Framework's features enhance a users privacy and control, including: the User Dashboard, the Privacy Preserving Data Aggregator, and the fine grained access control mechanism.

Keywords: Personal Cloudlet, OPENi, Cloud Security, User Centric.

I. INTRODUCTION

A central aspect of the contemporary digital economy is that of personal information. Many of the largest and most valuable Internet companies such as Google and Facebook are based on the business model of harvesting user data in order accurately facilitate the targeting of advertisements, and the influencing of behaviour. This business model is one which also applies to the market for smartphone and tablet applications. At present the privacy policies of these services generally operate on a take it or leave it basis; where users either reject the gathering of their personal information by not using a service or application, or they use the service under terms and conditions which they have little control over. With the inevitable move towards increased use of cloud computing these trends can be expected to continue and even to be exacerbated as more user data will be stored at remote locations.

The OPENi project provides a platform which will alter the dynamics of user control over their personal data. Its main components are the API framework and the Personal Cloudlet framework. The API framework allows for frictionless interoperability between cloud based services, and the Personal Cloudlet framework. The Personal Cloudlet is a virtual space that securely stores user data and gives users primary control over that data. OPENi allows for users to decide which aspects of their personal data they are prepared to share by giving them the option of fine grained authorization and access control. Therefore OPENi as a platform solves some of the problems raised, by empowering the user to take control of their digital identity and describes the user controlled privacy capabilities Of OPENi.

The design and implementation of the privacy preserving and user-centric features of the Personal Cloudlet

Framework is the primary focus of this project. The key objectives and other considerations that influenced their design. The Cloudlet Framework's privacy aware and user-centric features were implemented and how they were influenced by the objectives set out in the OPENi project.

II. GOALS AND OBJECTIVES

- 1) To build a web based security and authorization framework that will satisfy a context broker and the service provider's requirements that will enable the sharing of context information between applications in accordance with the users privacy settings and provide more control to user over their data.
- 2) To deliver an open source platform that will allow application consumers to create, manage and deploy their data in the cloud (Personal Cloudlet). Each Personal Cloudlet will have credentials that will be linked to its user's identity over the web in a similar way that a social profile does today.
- 3) To provide and promote a novel, user-centric application experience of cloud-based services not only across different devices and platforms but also inherently across different applications. The OPENi framework will enable users to share and distribute their data across their applications.
- 4) To ensure the OPENi platform provides support and maintains a low barrier to entry for application developers and service providers.

III. MOTIVATION OF THE PROJECT

- 1) Motivation of the project lies in the idea that it would be much sophisticated if the various application avail their

data requirements from one specific resource instead of gathering it from each of its source independently.

2) So here in this scenario the one specific resource (here, the admin) has the rights to decide the abstraction level of data for each of the application who uses the service.

3) It becomes more easy as the applications will just seek their data requirements from the admin. Also this assures the privacy and security of the data by the admin.

4) We accomplish these functionalities using the Personal Cloudlet over cloud.

IV. EXISTING SYSTEM PROBLEM

In existing format, the applications have to take the permission everytime it needs data, i.e., suppose there are various applications linked with facebook, whenever a user tries to use the application, the application asks for the access rights to facebook for that particular user. So, everytime the applications have to pass through the security layers for accessing the required data.

V. METHODOLOGY USED

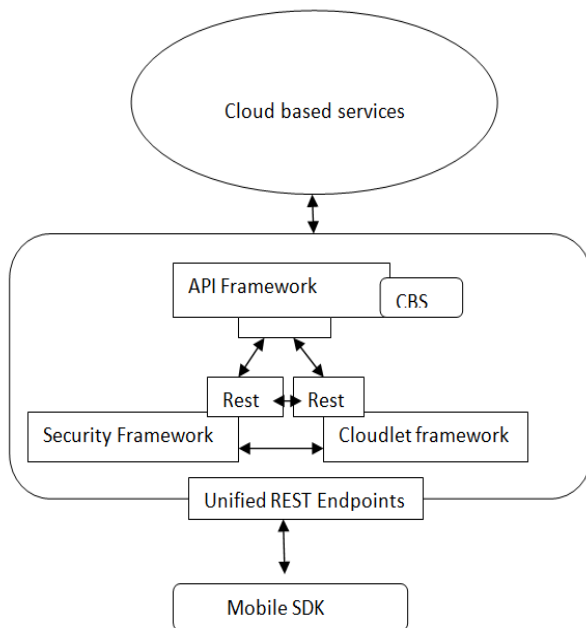


Fig1. System Architecture

The OPENi platform is composed of four distinct but interrelated frameworks, they are:

1) Security framework: This framework is responsible for security on the OPENi platform. The frameworks access control functionality; tightly coupled with the Cloudlet Framework; allows users to have more control over their personal data and their interactions with cloud-based services.

2) API framework: It is an open framework which operates with a number of cloud-based services, abstracting the challenges to a single open standard without affecting any service features.

3) Personal Cloudlet framework: The OPENi Personal Cloudlet framework provides users with one location to store and control their personal data.

4) Mobile Client Library: In order to provide access to the security, API, and Personal Cloudlet frameworks, OPENi architecture provides a number of mobile client libraries. One is a cross platform Javascript/HTML library for use in Apache Cordoba mobile web-apps and HTML5 and another is a native Android client library.

The combination of the API, Security, and Personal Cloudlet frameworks concept makes OPENi a very powerful and beneficial platform for consumers, application developers, and service providers.

VI. APPLICATIONS

It is used in cloud when the stored data on cloud is required by various applications and the privacy and security of this data is promised by the Personal Cloudlet framework. This data is accessed by the applications after seeking permission from the admin who decides the abstraction level for each application. We know the Personal Cloudlet framework and developed a privacy-preserving software for accessing data.

•Increased security

Providing access to the data depending on the permissions granted and sought by the applications makes it solely the responsibility of the admin to check any faults.

•One source

All the data is available at a single location. Suppose an insurance application wants the hospital and residential information of the citizens then it will avail it from the admin directly instead of gaining it from the medical and electricity department individually.

•Simplicity

Once the application seeks the rights from the admin then its not needed to pass through the security layers every time it needs the data.

VII. EXPERIMENTAL SETUP AND RESULT

Here we have tried to experiment with the abstraction of the security layers using the Personal Cloudlet framework. Suppose if a certain application shares it data with the framework then the same application and the system admin can access it, also the admin can share this data with other application according to the privacy and sharing rules and regulations. Due to this approach the third party applications need not pass through the security layers again and again every time instead just seek the permission rights initially. The OAuth and JWT ensures the applicability of the required functionalities.

VIII. CONCLUSION AND NEXT STEPS

The OPENi project implements fine grained access control to empower users with control over how applications can use their data. This is achieved through the use of REST based endpoints, object based access control, OPENi Types, and stateless JSON Web Tokens (JWT). The implementation allows users to share, reuse, and control

access to their data across many mobile applications while maintaining cloud scalability. The open source OPENi implementation is being leveraged by a number of commercialization projects for development of scalable cloud based mobile applications enriched with the privacy preserving access control mechanisms. Work currently under way with the OPENi project includes the validation of the implementation.

REFERENCES

- [1] "OPENi - Open-Source, Web-Based, Framework for Integrating Applications with Cloud-based Services and Personal Cloudlets." <http://www.openi-ict.eu/>, accessed: 2015-01-15.
- [2] "OPENi open source project," <https://github.com/OPENi-ict/>, accessed: 2015-01-15.
- [3] A. Iosifet al., "A Community-based, Graph API Framework to Integrate and Orchestrate Cloud-Based Services," in Proceedings of AICCSA, IEEE Computer Society, 2014, awaiting publication.
- [4] M. Petychakis et al., "Enterprise Collaboration Framework for Managing, Advancing and Unifying the Functionality of Multiple Cloud-Based Services with the Help of a Graph API." Collaborative Systems for Smart Networked Environments. Springer Berlin Heidelberg, pp. 153–160, 2014.
- [5] K. Doyle and D. McCarthy, "OPENi White Paper: An End Users Perspective: Digital Identity Putting the Genie Back in the Bottle," <http://www.openi-ict.eu/wpcontent/uploads/2014/07/openi-whitepaper.pdf>, Sept 2014, accessed: 2015-01-15.
- [6] D. McCarthy et al., "OPENi Deliverable D3.5: OPENi Cloudlet Framework Design Document," http://www.openi-ict.eu/wpcontent/uploads/2014/10/OPENi_D3.5.pdf, Sept 2014, accessed: 2015-01-15.
- [7] R. Kleinfeldt et al., "OPENi Deliverable D3.6: OPENi Security and Privacy Specification," http://www.openi-ict.eu/wpcontent/uploads/2014/10/OPENi_D3.6.pdf, Sept 2014, accessed: 2015-01-15.
- [8] "Apache Cordova," <http://cordova.apache.org/>, accessed: 2015-01-15.
- [9] R. Illera, S. Ortega, and M. Petychakis, "OPENi Deliverable D2.3: Security and Privacy Considerations for Cloudbased Services and Cloudlets," http://www.openi-ict.eu/wpcontent/uploads/2013/11/OPENi_D2.3.pdf, Jan 2013, accessed: 2015-01-15.
- [10] "PrimeLife - Bringing sustainable privacy and identity management to future networks and services," <http://primelife.ercim.eu/>, accessed: 2015-01-15.
- [11] C. A. Ardagna, L. Bussard, S. D. C. Di, G. Neven, S. Paraboschi, E. Pedrini, S. Preiss, D. Raggett, P. Samarati, S. Trabelsi, and M. Verdicchio, "Primelife policy language," <http://www.w3.org/2009/policyws/papers/Trabelisi.pdf>, 2009, accessed: 2015-02-25.
- [12] "A4Cloud, The Cloud Accountability Project," <http://www.a4cloud.eu/>, accessed: 2015-02-25.
- [13] "TCLLOUDS: Trustworthy Clouds Privacy and Resilience for Internetscale Critical Infrastructure," <http://www.tclouds-project.eu/>, accessed: 2015-01-15.
- [14] "TRESSCA TRustworthy Embedded systems for Secure Cloud Computing Applications," <http://www.trescca.eu/>, accessed: 2015-01-15.
- [15] "CUMULUS: Certification infrastructure for Multi-Layer Cloud Services," <http://www.cumulus-project.eu/>, accessed: 2015-01-15.
- [16] "PICOS - Privacy and Identity Management for Community Services," <http://www.picos-project.eu/>, accessed: 2015-01-15.
- [17] "SWIFT - Secure Widespread Identities for Federated Telecommunications," <http://www.ist-swift.org/>, accessed: 2015-01-15.
- [18] "Apache Cocoon," <http://cocoon.apache.org/>, accessed: 2015-01-15.
- [19] "Apache Shiro," <http://shiro.apache.org/>, accessed: 2015-01-15.
- [20] "Spring Security," <http://static.springsource.org/spring-security/site/>, accessed: 2015-01-15.
- [21] "ZeroMQ," <http://zeromq.org/>, accessed: 2015-01-15.
- [22] "Couchbase Server," <http://www.couchbase.com/>, accessed: 2015-01-15.
- [23] "Mongrel2," <http://mongrel2.org/>, accessed: 2015-01-15.
- [24] "Docker: Build, Ship and Run Any App, Anywhere," <https://www.docker.com/>, accessed: 2015-01-15.
- [25] "JSON Web Token (JWT)," <http://jwt.io/>, accessed: 2015-01-15.