The Design and Implementation of a Cloud-Based Application Demonstrating the Use of Sticky Policies and Encrypti...

Kituyi Mayoka

THE DESIGN AND IMPLEMENTATION OF A CLOUD-BASED APPLICATION DEMONSTRATING THE USE OF STICKY POLICIES AND ENCRYPTION TO ENFORCE USERS' PRIVACY AND ACCESS CONSTRAINTS

AUTHORS

Peter M. Ogedebe
Bingham University
peter@binghamuni.edu.ng

A.H. Alaku
Bingham University
peter@binghamuni.edu.ng

Geoffrey Mayoka Kituyi
ICT University, USA
kituyi@ictuniversity.org

CONFERENCE THEME: ***Harnessing ICT in Education for Global Competitiveness***

VENUE: HILTON HOTEL, YAOUNDE

CONFERENCE PROCEEDINGS

Vol. 6. ISBN: 978- 9956- 27- 030- X

EDITOR
Mayoka Kituyi

CO-EDITORS
Adekunle Okunoye, Charles Masango, Cosmas Nwokeafor, Kehbuma Langmia, Victor Mbarika

ICT4AFRICA 2014 Conference Papers
Paper No. 81, pp. 316-334

Conference Paper

The Design and Implementation of a Cloud-Based Application Demonstrating the Use of Sticky Policies and Encryption to Enforce Users' Privacy and Access Constraints

Received May 31st 2014, accepted August 31st 2014

Abstract. *This research was motivated by the fact that traditional IT security approaches that focus on perimeter security by protecting network edges, firewalls and application endpoints appear to be deteriorating due to the introduction of the cloud concept whose architecture is more of bundled infrastructure and shared resources. Data are widely shared and data owners are losing control over the collection of personal information, their processing and usage as well as sharing with third parties. All these are done without regard to the choices and privacy preferences of the data owner.*

In this research, a data governance solution using sticky policies was designed and implemented using .NET 4.5 Framework with ASP.NET web application development technology. C# supported in the framework serves as the server-side programming language. This framework also integrates with the Microsoft SQL Server version 11.00.2100 which then serves as profile data and sticky policies repository hosted locally at the development stage. A user is able to create and manage profile information by supplying core data and specifying whether or not a piece of data can be shared with a third party or not for a given usage purpose and in what format. Third party applications are required to maintain verifiable credentials in the system and are permitted to request for user's profile information via a web service hosted in the cloud. The policy engine in the system is able to interpret user aggregated sticky policies to determine if data can be shared and how it can be shared as a mark of respect to the choices of the data owner.

KEYWORDS –Cloud based application, Sticky policies, Encryption, Internet security, Privacy.

INTRODUCTION

Over the years, dramatic changes have taken place in the architecture of computing, thereby triggering associated changes in measures of safeguarding organizational information technology (IT) resources including important business data as well as users' personal information. Traditionally, data security is provided by firewalls, endpoint and application user authentications as well as other network defence and control mechanisms. The data themselves are left naked and any compromise at either the network or endpoints and application layer get the data exposed and vulnerable.

Previously, data security focused more on intensifying security controls at the network level to End Points and Applications. However, this has now been extended to the data themselves in a sufficiently tight and heavily coupled fashion. According to a [1], report on security, organizations have and are predicted to continue to invest and focus greater attention on

data controls where relative efficiency achievable is expected to be on the rise as adoption of cloud computing and its associated trending revolution becomes very much inevitable.

The privacy of users' data is thought to be poorly managed in web 2.0 and arguably, the cloud concept may have possibly inherited this from the former. Storage and processing demand of users' personal information has also been on the rise as technological advancement has to a great extent resulted in high demand for such.

It is worth noting that cloud users are faced with security threats both from within and outside of the cloud [2]. So far, there are very much insufficient and barely convincing, concrete and technical guarantees that could assure cloud customers and users of their full control of their personal as well as business data as they fly around the cloud [3]. These data and files will often be shared and given out for one business transaction or the other. Ensuring sufficient privacy and access definitions of these data as they move from one point of usage to the other is increasingly becoming a challenge with unrelenting demand for achieving the same.

LITERATURE REVIEW

There are three (3) issues that are key and central to this study. They are:-

1. Security Challenges in the Cloud
2. Data Governance and Control
3. Sticky Policies and Identity Management

Firstly, Cloud computing is predicted to become a big deal and arguably the future of computing even though the question of security specifically with regard to data have continued to dominate discussions. Secondly, Data governance and data control are beginning to enjoy the increased attention and offer to address the many security concerns in computing. Lastly, the idea of sticky policies and other technical measures of empowering data owners to be able to stay in charge of their data have assumed a central role in data governance and control.

Security Challenges and Data Handling Issues in the Cloud

Cloud computing introduces the idea of having data and programs centrally stored in the 'Cloud', and made available to users and clients through lightweight channel [4]. It refers to both applications provisioned as services (SaaS) over the Internet and the hardware and software infrastructures across data centers (PaaS and IaaS) that support such delivery [5]. Even though there is still confusion about what exactly are the boundaries of the cloud, its recent dominance of discussions and general popularity may not be easily contested. It appears to be the irresistible and possibly the big thing in modern computing.

Cloud computing is designed to allow the power of information technology to be delivered to clients on demand [6], offering dynamically scalable computing resources provisioned as services over the internet [7].

Though it offers significant economic benefits by its promise of reducing capital and operational expenditure, [7] as well as enormous potential of delivering sufficient scalability, reliability and efficiency [8], there are genuine fears [9] with regards to fully adopting this irresistible offer of the much speculated utility computing. It has been observed that there are pressing concerns with the level of security provided by the cloud concept and addressing these concerns is very much important as promoting the paradigm itself. A number of investigative studies have been undertaken with an interest in understanding the peculiarity of the security challenges posed by this environment. These have resulted in a quite significant insight on the scale of cloud security challenges.

According to [4], the unique challenge of security in cloud computing is a result of

customer data being accessible to the cloud service provider and possibly its subcontractors where they could be deliberately or inadvertently disclosed and consequently misused.

In a deep dive article published in InfoWorld website, it is understood that, in as much as more benefit is derived from organizational investment in IT resources and services, the biggest challenge in cloud computing is that, a lot of control over such resources is incalculably giving away to unknown personnel. Whatever the kind of technology or services being adopted in the cloud, a great deal of control over data is ultimately handed over to a third party [11], [12].

Another problem is, Web 2.0 was thought to have poorly managed the issue of users' privacy in cases where internet applications wanted to publish users' private data that were linked, shared, aggregated, tagged, and copied even though mechanisms for controlling such publications were not precisely defined [13]. It could be argued that cloud computing possibly inherited from Web 2.0, the many problems of securities including handling of users' private data confidentiality as well as enforcing access and usage constraints.

The severity of the question of security in cloud computing has resulted in suggestions that the concept may be more suitable for less prioritized applications while critical business applications continue to maintain the use of organizational resources [14]. It is even worrisome in some cases where companies are likely not to even consider its adoption, but stick to its traditional data centre approach [7]. Cloud computing is faced with a very important question and a strategic one at that. The answer to this question may have a significant role to play in determining its future and by extension the future of computing.

With data being regarded as a key and arguably the most valuable and expensive resource in computing, ensuring its security may go a long way in restoring confidence in what it is standing for and what it is offering. A lot can be said to be depended on data and a lot is about data even in the emerging cloud environment. That is why this research regards it as a critical foundation.

Data Governance and Data Control

Data in computing has for a long time suffered exposure to a number of vulnerability and security threat irrespective of their environment. It appears though that there exist more specific security concerns that are unique to the cloud owing partly to the fact that resources are scalable and dynamically provisioned on demand [4].

Cloud computing has again brought about a greater emphasis on data and less emphasis on systems like it used to be in traditional computing. It is prompting a response to this trend by alerting stakeholders on the need to be aware of the growing concern and ensure best practices for governing and operating data and information in the cloud [15]. According to [1], data control is projected to attract increasing attention and investment by organizations. This is expected to result in proportional growth in the effectiveness of data security as compared to reliance on network controls, endpoint and application control mechanisms.

It is widely accepted that, data protection and governance need to be taken more seriously with closer attention being paid to data owners' privacy preference. Guidelines towards achieving this need to be clearly defined [14]. It is also argued that, such measures have to go beyond usual policy documents, service level agreement (SAL), and other non-technical measures that have fallen short of expectations, and lack clients' trust due to its inability to ensure the safety of data in the cloud [3], [4]. Security and privacy of users' confidential data is not just an option but a critical requirement in certain application domains [16].

Data governance refers to the approach adopted to holistically address the many aspects of data management, including information privacy and security as well as compliance [17]. [18]

puts it this way: 'Data Governance is the exercise of decision-making and authority for data related matters'. Such a system of decision rights and accountabilities executed as agreed-upon models should describe who takes what actions with what information, and when, under what circumstances and in which way. A number of flavours to data governance have been outlined in which any organization willing to undertake a data governance project will want to focus attention on one or more of it. They include:

- Policy, Standards, and Strategy
- Data Quality
- Privacy, Compliance, and Security
- Application Architecture and Integration
- Data Warehouse and Business Intelligence
- Management Support

Personal Identifiable Information refers to the users' personal data or information that is specific in nature and could aid a trace to an individual [19]. Such data may include: name, email address, phone number, and date of birth etc. With recent advances in technology such as face recognition, where an individual appearing in published photos could be identified [13], personal files like images may also be regarded as a PII.

The question still remains that, how can data control be effectively delivered in a typical cloud environment or distributed application bearing in mid users privacy agenda. How can data be made accessible to applications and data consumers for such genuine consumption demands and yet adhere to data owner constraints. This research aims to achieve a technical solution that can be managed by the data owners themselves in dictating access and usage rights and hence the search for tools that can support such delivery.

The Idea of Sticky Policies

Sticky policies in data handling and governance is the idea that privacy policies could be attached to data owners' data and become responsible for driving access control decisions and policy enforcement [20]. The idea was brought forward by [21], where it was thought that, privacy constraints should be made to flow with personal data [11].Elaborately, this concept means, when a user or an enterprise discloses her data, it should be possible for them to express their consents in the form of privacy policies. Such policies should then be attached to her data wherever the data moves, thereby driving authorization decisions [20].

There are a number of things that are central and essential to the idea of sticky policies in data governance. They are:

1. Sticking policies to data
2. Enforcing sticky policies
3. Transferring sticky policies throughout a distributed system or environment

Policies Enforcement

To make it clear on why more is to be done than just expressing or attaching SP to personal data, let us think through this scenario. What happens when an attacker gains access to our raw data in its storage media or intercepts it within the communication channel during transport? Of course the policies could just be disregarded and our naked and innocent data units will just be misused as desired by the new authority. What is important to us however is to get our data safe as much as possible at all levels of vulnerability and get its associated privacy policies enforceable for whatever direct access and usage. Putting our naked data in its plain form into the database or whatever storage media could result in far less an effective technical

guarantee for the confidentiality concern of our personal data in a typical cloud environment.

The essence of a technical mechanism for ensuring privacy protections is to support enforcement and possibly auditing obligations that have been outlined [11]. Due to the weak nature of some implementations of this concept where enforcement is not guaranteed, questions have been put forward regarding how sticky are these policies to their associated data units? We shall look at the various technologies and mechanisms such as cryptography, Trusted Agents TA, etc. that are available and could be used for supporting enforcement of sticky policies.

Cryptographic Support

Computing systems developers often do not wish to store or transmit plain text or readable data values within or across computing systems mostly due to the question of data security when accessed by the wrong party. This has resulted in such systems taking advantage of cryptographic techniques that allows them to encrypt these pieces of data units before storage or transmission across storage and application endpoints.

A Review of data governance solutions and sticky policies implementations shows the need for users' data and its attached privacy as well as access policies to be safeguarded and their security extended or intensified. It appears that, cryptographic techniques plays an invaluable role in realizing an effective technical guarantee and enforcement of users' privacy, access, and confidentiality constraints imposed over their own data. It is however not just about adopting encryption into an application but also considering how suitable this encryption is and what overhead this will introduce into the system. Another important consideration is its usability. Cryptography has been for long the backbone of security solutions delivery in computing. It is very powerful and can also on the contrary complicate a security solution if design decisions are not carefully reviewed.

Review of Key Data Governance Models

A number of research has been carried out and the concept of sticky and privacy policies on data have been used with varied and quite interesting models, giving us insight into the wide and possibly usefulness and applicability of this idea in terms of data handling and sharing. Here, we take a look at some key models and projects among others and explore briefly how they were used, which will avail us an understanding of these policies in practice as well as the renewed commitment to rebuilding users' trust and ensuring control in data usage and communication.

Sticky Policies and Access Control Engine (SPACE) Solution

This solution was motivated by the fact that there is a lack of trust in cloud infrastructures concerning the handling of data and its storage conditions. This concern is very well shared by this research and understands the need for the provision of a technical guarantee that should convince a cloud customer, assuring them of significant control over their own data. SPACE is based on sticky policies and offers data access and usage control functionalities around the cloud in a centralized fashion. Additionally, the user is provided with visual reporting capabilities that feed the user with access and up-to-date usage of his/her data across the cloud.

In a typical e-commerce scenario, SPACE technology mediates between the cloud server where users' data are securely held and consumers of these data pieces on the other hand like the retailers and other third parties who will normally initiate a request via defined APIs for customers' information in order to carry out sales transactions. User data is isolated in a safe zone in the server and access is only available through the SPACE that is in effect empowered to be

able to interpret the request and expressly enforce user access and usage constraints.

The EnCoRe (Ensuring Consent and Revocation) Project

With partners in the cadre of respected and highly placed academic bodies and industry giants as well as receiving funding from the United Kingdom Research Council, the EnCoRe project aimed at delivering efficient means of empowering users in controlling or managing issues relating to their personal information held by others, thereby going a long way in restoring confidence in digital economy as well as reviving users' assurance of compliant use of their data [10]. What makes the outcome of this project particularly appealing is the series of social research being carried out with the aim of understanding and putting into context, the real concerns of users in distributed computing. This forming part of the overall project may have well provided valuable inputs and significantly informed key decisions in the delivery of somehow a concise technical architecture for the management of identity and privacy in computing systems.

SYSTEM IMPLEMENTATION

This section reports the development phase of the data governance solution proposed in the design. A system that aims to deliver the functionalities that allows users (Data Owner) to manage their profile information by attaching usage and access constraints to these pieces of data. Through cryptographic techniques, profile data and sticky policies are protected against unauthorized access. Profile data are also shared through web services with authorized parties and guarded by the policy decisions and enforcement engine component. This system can be subdivided into four distinct layers.

1. User Web Interface and Profile Manager
2. Encryption and Security Module
3. Policy Decisions and Enforcement Engine
4. Web Services: Data Access Application Programming Interface

This application was developed in the .NET 4.5 Framework with ASP.NET web application development technology. C# supported in the framework serves as the code-behind or server-side programming language. A database management system is also needed for this system data storage requirement. This framework also integrates with the Microsoft SQL Server version 11.00.2100 which then serves as profile data and sticky policies repository hosted locally at the development stage.

User Web Interface and Profile Manager

This section presents the implementation of a combination of the user-web interface component and the profile-data manager component of the system use case architecture as detailed in Figure 1. These components provide the user with visual interfaces and a host of support for profile creation and subsequently, functionalities needed for profile management. The overall system is simply a web application from the perspective of the user. It comprises pages for new user registration, profile creation, registered users' login, and profile update.

New User Registration Page (NewUser.aspx)

This web page serves as the entry point into the web interface for users who do not have valid login credentials. A screen shot of the page is in Figure 2. The new user begins registration by entering preferred Login ID and password. As the Login Id input field loses focus, an

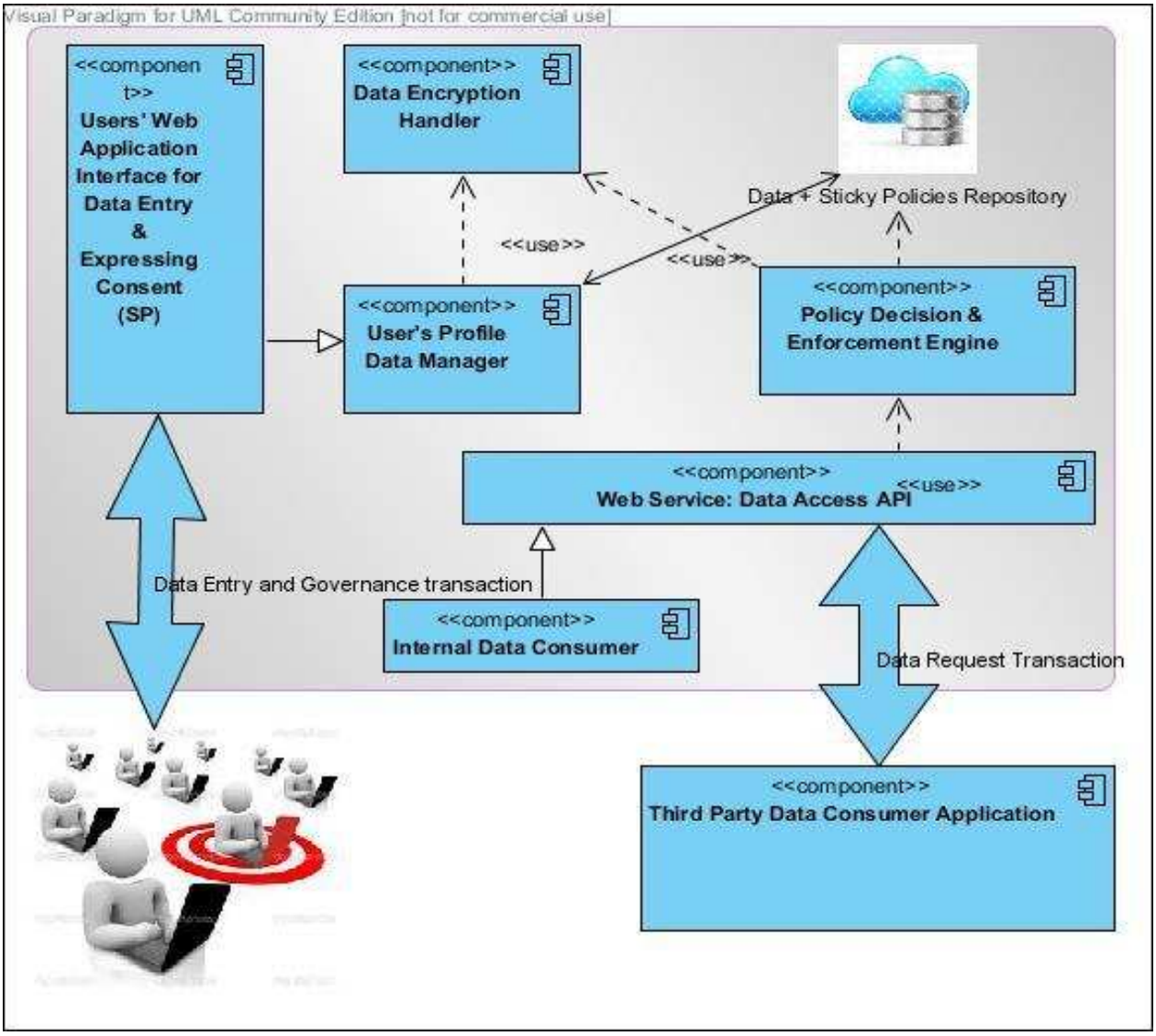

Figure 1: Use Case Architecture

asynchronouspostback is made that get the server to check if the chosen Login Id have been used by another user. If it has been used, the new user is prompted to choose another ID. As in most web applications, passwords are verified by comparison to ensure that the user understands and can remember the supplied security token. The user then selects from a list of available encryption algorithm. This is to be used for encrypting profile data for this user.

On submitting this registration form, the code-behind event handler which performs the function of the Profile Data Manager component as in the design use case, takes this form data, pass the selected encryption number to the Encryption Handler Module (Figure 3) which in response returns an encryption object containing the internal encryption key and initialization vector for the chosen encryption algorithm. The profile Data Manager then generates a unique identifier of data type 'Guid' and then proceeds with the database write into UserSecurityCredentials table. If the server successfully writes the new user's login and encryption details into the database, a session state object is created for the user to hold Login Id and the generated internal unique identifier. This session state will then be available for subsequent client-server interaction and the user request is redirected to NewProfile.aspx for

profile creation. On the other hand, if unsuccessful, error information of the write operation is sent back to the user on the same page.

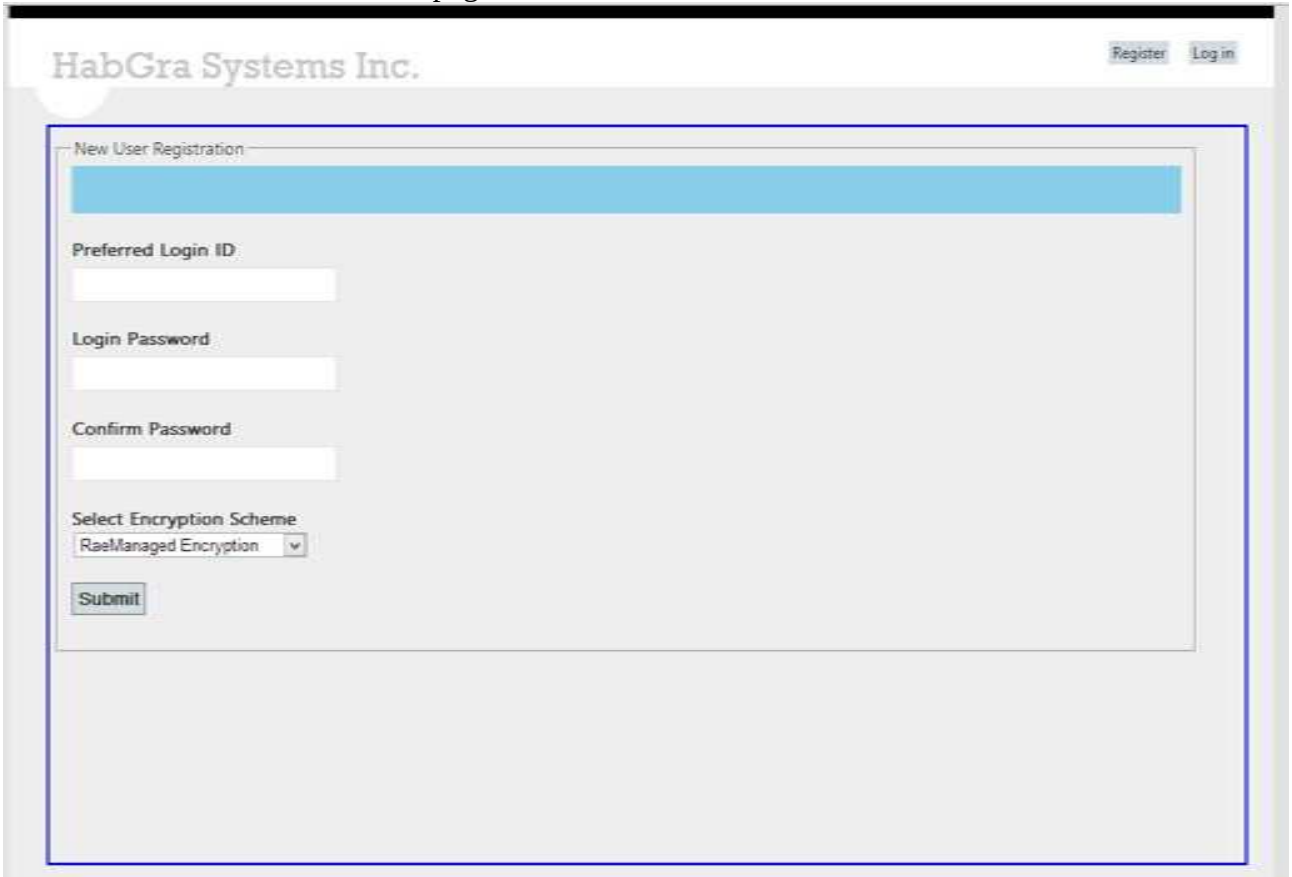

Figure 2: New User Registration Page

New Profile Set-Up (NewProfile.aspx)

This is the second phase of a new user registration. To allow users to both supply their profile core information and also define access and usage constraints in a fine-grained and easy to use manner, this web page provides a visual web interface that can be used. As can be seen in the screen shot presented in Figure 3, the profile page is subdivided into four sections with each supporting data entry and policies definition of an instance of the data unit model. For each instance, the user can specify whether or not it can be shared and if it should be shared on an anonymous basis. Check boxes are used to support this Boolean input as required by the database table's field for these instances. In Figure 5, enumerated collections are used to store the different aggregations of possibly data sharing formats as per data unit instance. These collections are: NameFormat, ContactAllowed, DateBirthFormat, ad AddressInfoAllowed. On page load, these entries are read into their respective Share <DataUnitInstance>As list boxes for users to choose from. A collection of trusted third parties and data usage context are also held in a similar enumerated collection to be loaded into third party and access context for all instances of the data unit model.The code snippet below shows how this was achieved.

```
if (!IsPostBack)
        {
//Name Information
```

```
foreach (string item inProjectData.NameFormat)
            {
ShareNameAs.Items.Add(item);
            }

foreach (string item inProjectData.AccessContext)
            {
NameContextListBox.Items.Add(item);
DateBirthContextListBox.Items.Add(item);
ContactContextListBox.Items.Add(item);
AddressContextListBox.Items.Add(item);
            }

foreach (string item inProjectData.Companies)
            {
NameThirdPartyListBox.Items.Add(item);
DateBirthThirdPartyListBox.Items.Add(item);
ContactThirdPartyListBox.Items.Add(item);
AddressThirdPartyListBox.Items.Add(item);
            }
//Date of Birth
foreach (string item inProjectData.DateFormat)
            {
ShareDateBirthAs.Items.Add(item);
            }
//Contact Information
foreach (string item inProjectData.ContactAllowed)
            {
ShareContactAs.Items.Add(item);
            }
//Address Information
foreach (string item inProjectData.AddressAllowed)
            {
ShareAddressAs.Items.Add(item);
            }
        }
```

One significant operation that takes place on the server every time a segment of the form in the new profile page is submitted besides the data encryption activities is the wrapping of the selected third party and allowed access context into a valid XML element to be written to the table representing the given data unit instance.

The web interface is built to provide a friendly and easy to use web interface. In a live system, more attention will have to be paid to making the web pages even more friendly and descriptive so as to appear appealing instead of risking having the system boring and difficult to use. This is however not a core requirement of this project and hence less effort spent on delivering the said objective.

After completing this form, a user profile would have been properly set up and the user can return back to the system to log in and even modify profile information or generate a new encryption key and initialization vector.

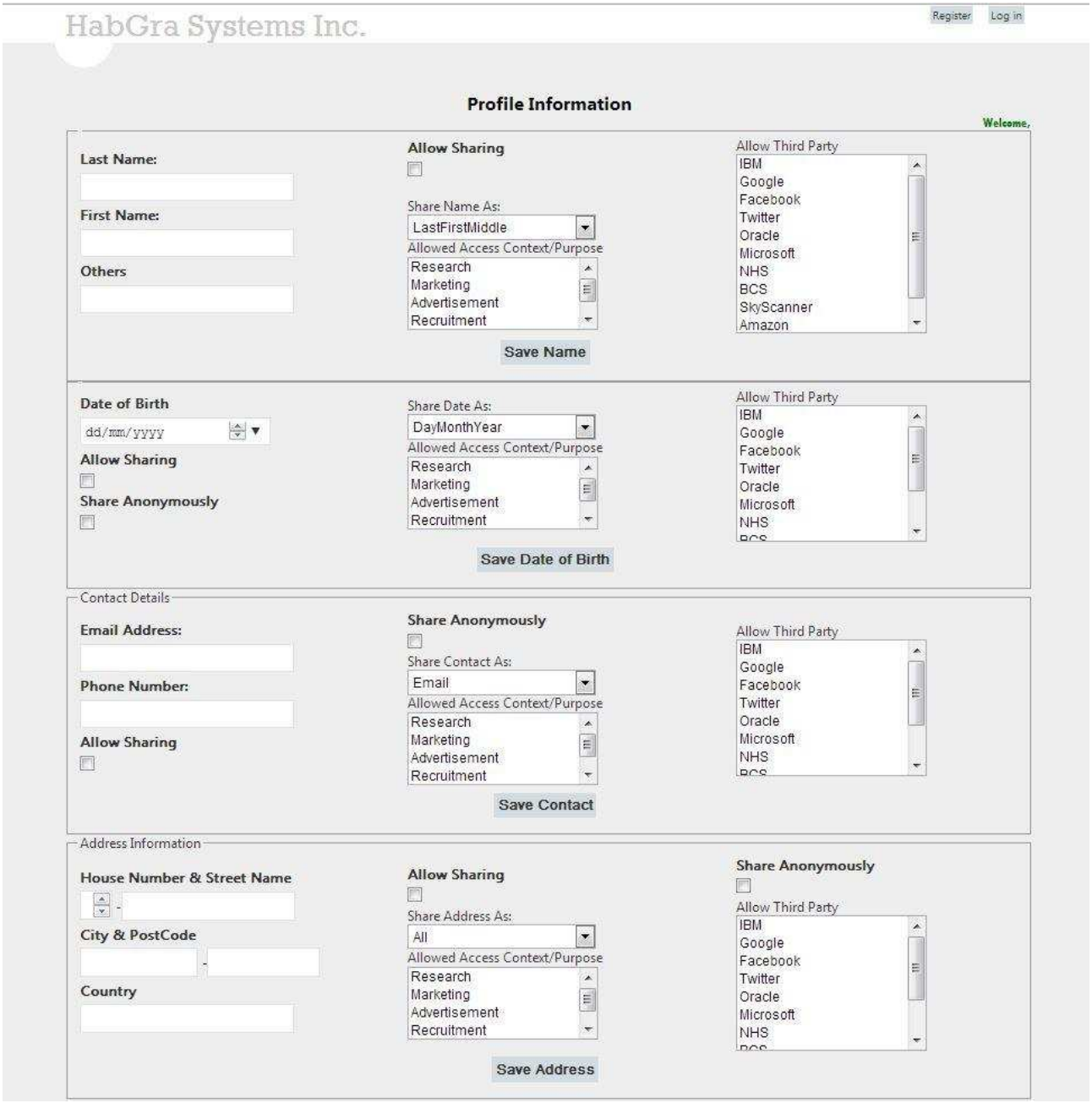

Figure 3:New Profile Date, Access and Usage Policy Set-Up

The Encryption Module

The design of this component suggests the use of a collection of symmetric key encryption algorithms from which a user can make a choice of which to use for the handling of his/her profile data encryption and decryption calls. Figure 4 shows the class diagram of classes in this component. The 'DataEncryptionInterface' defines the basic methods required of any class that is implementing an encryption algorithm and that will be included in the module's collection for a data owner to choose from. This component has implemented two encryption algorithms. They are:

1. AesManaged Encryption
2. RC2 Encryption

Both are symmetric algorithms, part of the .NET library collection and available in the System.Security.Cryptography namespace.

The choice of what encryption algorithm to be included in this collection was not rigorous but was centered on having an easy-to-manage symmetric algorithms for the purpose of demonstration rather than investigating performance and other criterion. This is to manage time and be able to deliver the core functionalities of the overall system. It will be interesting taking on both these considerations as it will bring about greater efficiency and performance enhancement.

The 'EncryptionHandler' as shown in the class diagram (Figure 5) is the coordinating class for this module. User Web Interface, PDEE, and the Data Access Web services component communicate with this module through this class to carry out cryptographic activities required of their processes. It takes an integer as argument which maps to an encryption algorithm in its collection and the handling class creates and returns an instance of the specified encryption algorithm class. The calling component is able to retrieve from the data repository, the internal encryption key and its accompanying initialization vector to be used in encrypting or decrypting the target user's data.

The code snippet below shows how the constructor of the 'EncryptionHandler' class accepts the integer and determines which of the encryption algorithm's instance in its collection to instantiate and return to the calling component.

```
publicEncryptionHandler(int ID)
        {
encryptionId = ID;
switch (encryptionId)
            {
case 1:
cryptoService = newAesManagedEncryption();
break;
case 2:
cryptoService = newRC2CryptoServiceProvider_SP();
break;
default:
break;
            }
GenerateNewKeys();
        }
```

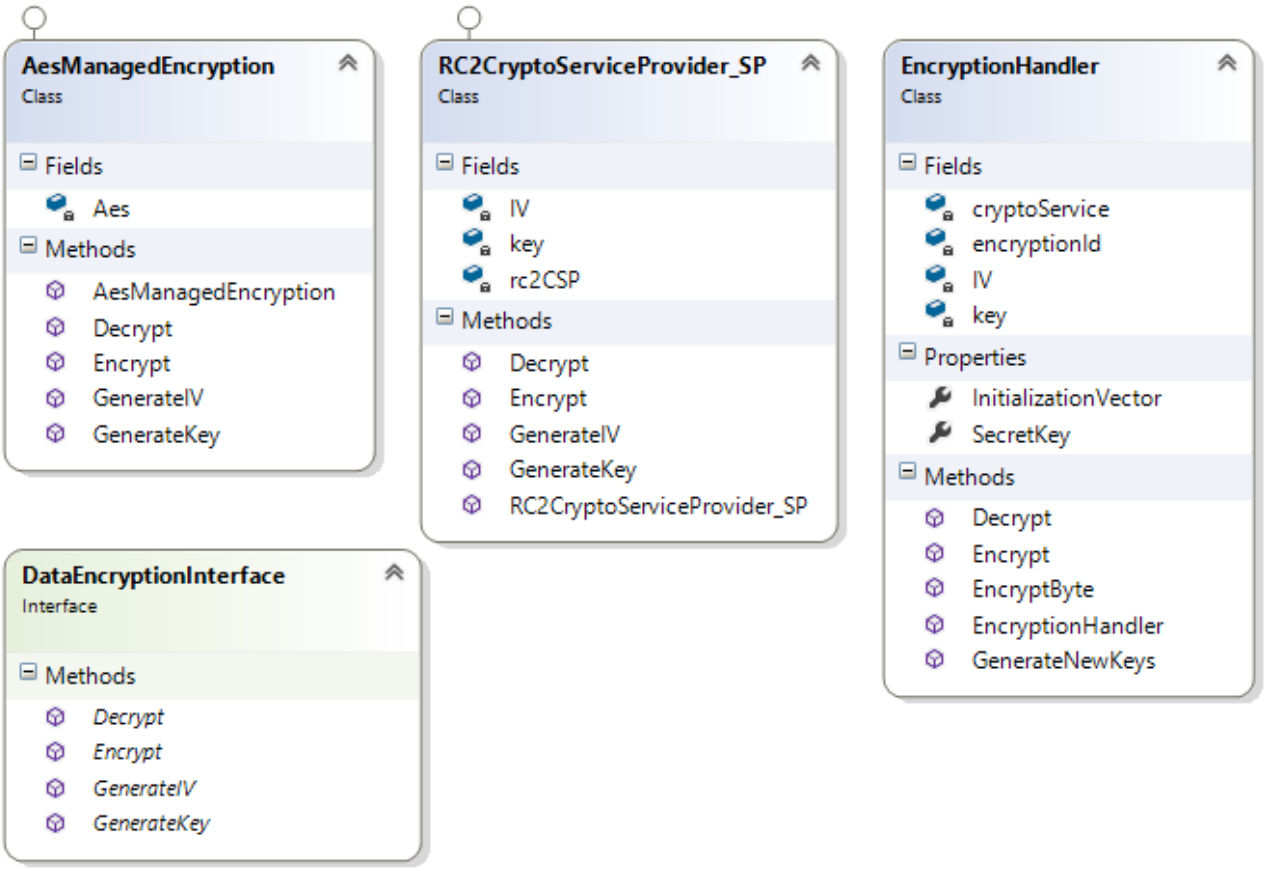

Figure 4: Encryption Module Class Diagrams

Policy Decision and Enforcement Engine (PDEE)

This component is designed to act as the intelligent powerhouse of the system. It mediates between the Data and Sticky Policies Repository of this system, and the web service component which allows third party application to share users' profile information. It exposes a static public method for each instance of a data unit model in the system and another method for requesting an aggregation of all the data unit instance making up the user's complete profile.

The class diagram of this component is presented in Figure 4.6 showing the signature of the methods in it. A request for data comes into the system via the web service component. All the four methods handling request for instances of the data unit model are structured and works in a very similar fashion except for the difference in the items making up their respective instances and the tables used for storing their data and sticky policies.

Figure 5 is a common flowchart showing the flow of data request transaction and policy decisions and enforcement processes in the methods of the PDEE component.

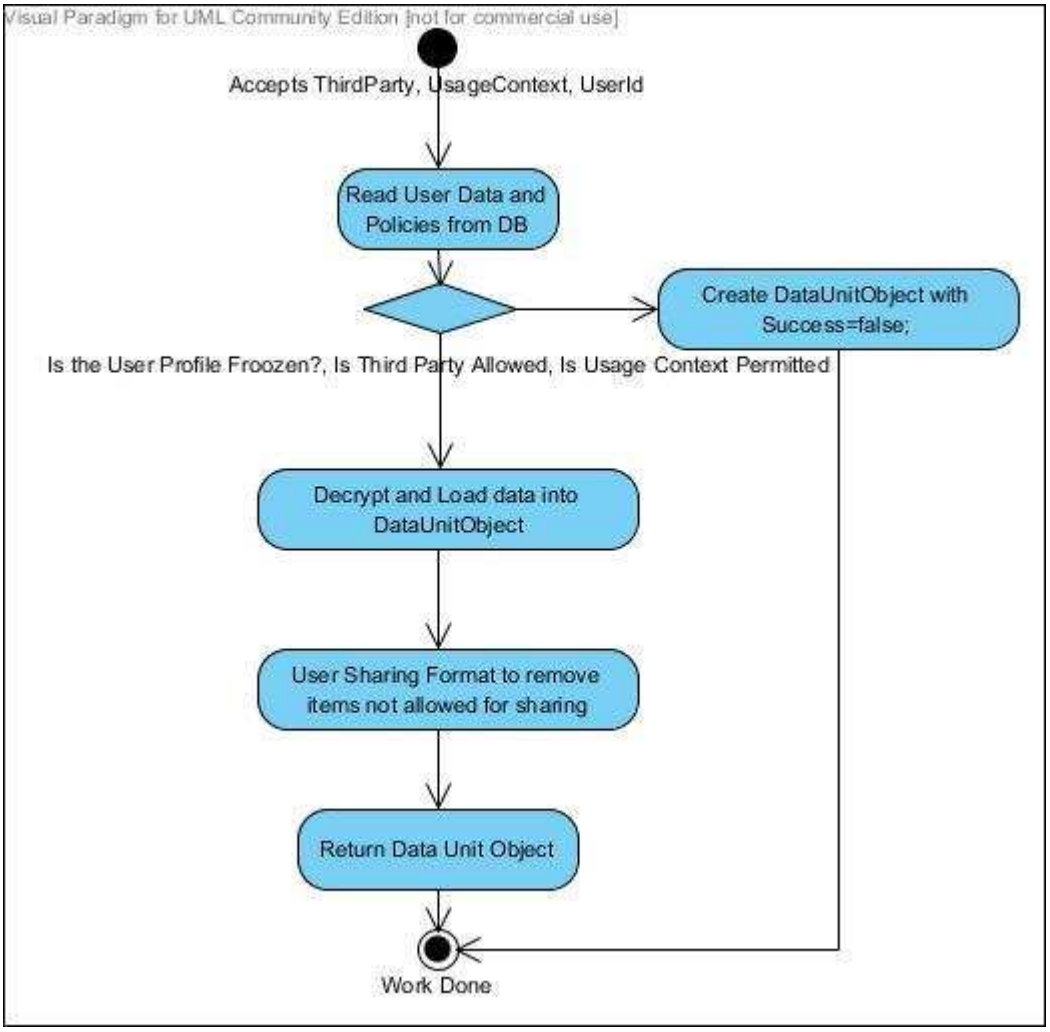

Figure 5: Flowchart showing Policy Decision and Enforcement Processes

The PDEE takes a Third party credentials, intended usage context and the identity of the user for which profile data are required. It then reads the user data and its sticky policies to begin data request processing. Key policy questions are asked and decisions made. Typically, the engine checks if the user profile has been frozen which means access to the entire profile have been momentarily suspended. It will further check to verify that the requesting third party is on the list of permitted data consumer and that usage context is in agreement with allowed context. In each of these points, a false Boolean response will result in a failed response to the data requester.

If third party and usage context are not violating user preferences, data can then be decrypted and the requester data unit instance object is built and loaded with user data. Users are required to specify how much of a data unit instance they will be happy sharing with third parties. The preferred sharing format is used to determine which part of the instance data piece must be removed. The code snippet below shows how it is handled in the case of the address data unit instance.

```
switch (Format) //Remove what is not allowed
        {
case"HouseStreetCity":
AddressObject.PostCode=AddressObject.Country="XXXXXXXX";
break;
case"StreetCity":
AddressObject.House = -1;
AddressObject.PostCode = AddressObject.Country = "XXXXXXXX";
break;
case"CityCountry":
AddressObject.House = -1;
AddressObject.PostCode = AddressObject.Street = "XXXXXXXX";
break;
case"PostCodeCountry":
AddressObject.House = -1;
AddressObject.City = AddressObject.Street = "XXXXXXXX";
break;
default:
//Handle Default case
break;
            }
```

An object instance of one of: DateBirthDataUnit, or AddressDataUnit, or NameDataUnit, or ContactDataUnit that all inherit from the abstract class: DataUnitInstance is returned to the requester. Class diagram of these classes is also presented in Figure 7.

In the case of a request for the entire profile of a user, a DataUnitInstance typed list is created and calls made for the individual data unit instance which are together held in the list since the all inherited from this common abstract class. The code snippet below shows this method and how it handles this operation.

```
publicstaticList<DataUnitInstance>GetProfileData(GuidUserId, CompaniesThirdParty, AccessContext Context)
      {
List<DataUnitInstance>ProfileData = newList<DataUnitInstance>();
ProfileData.Add(GetNameInforcement(UserId, ThirdParty, Context));
ProfileData.Add(GetContacts(UserId, ThirdParty, Context));
ProfileData.Add(GetDateOfBirth(UserId, ThirdParty, Context));
ProfileData.Add(GetAddressDetails(UserId, ThirdParty, Context));
returnProfileData;
            }
```

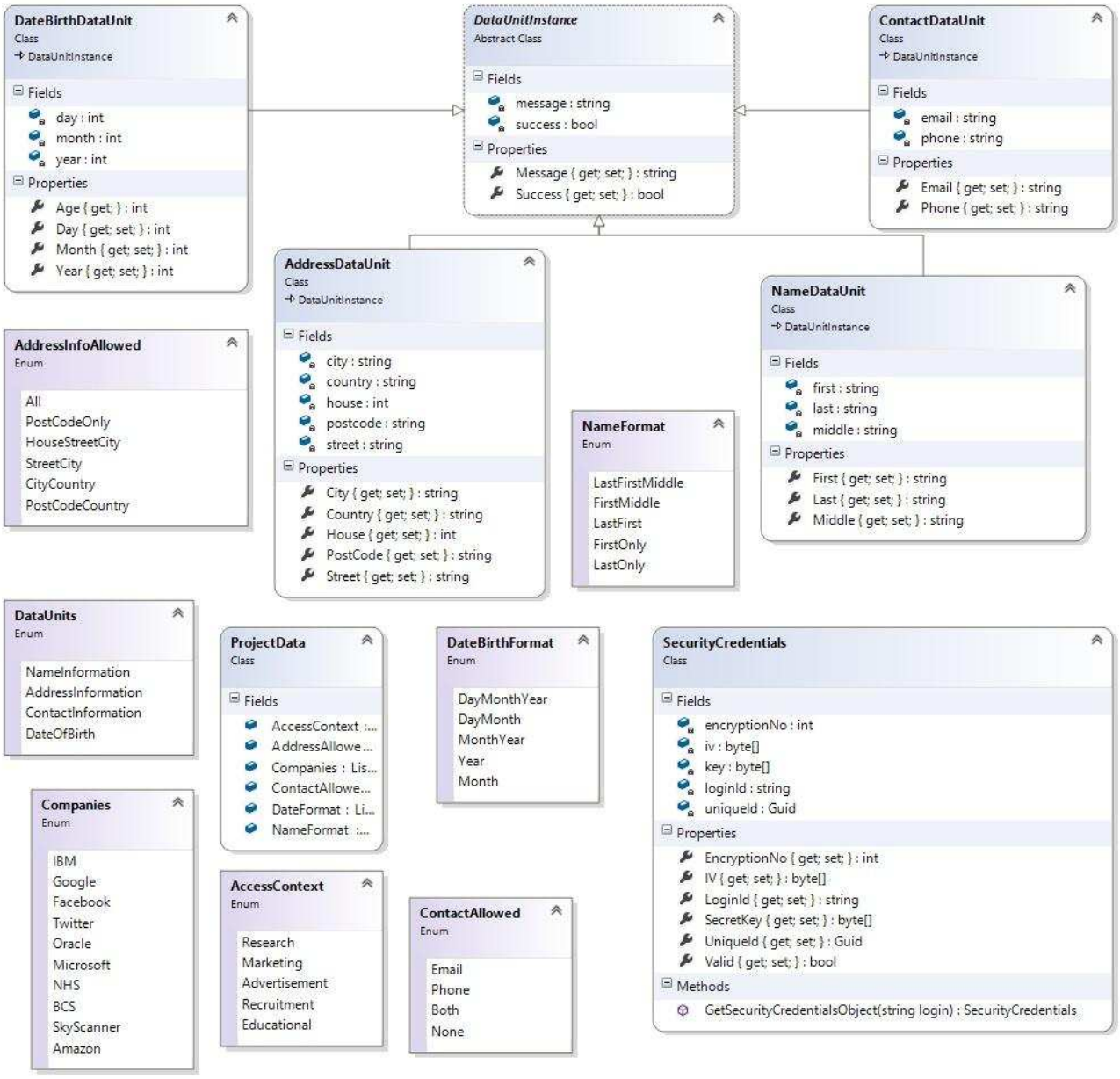

Figure 6: Class Diagrams of Application Utility Modules

Web-service and Data Access APIs

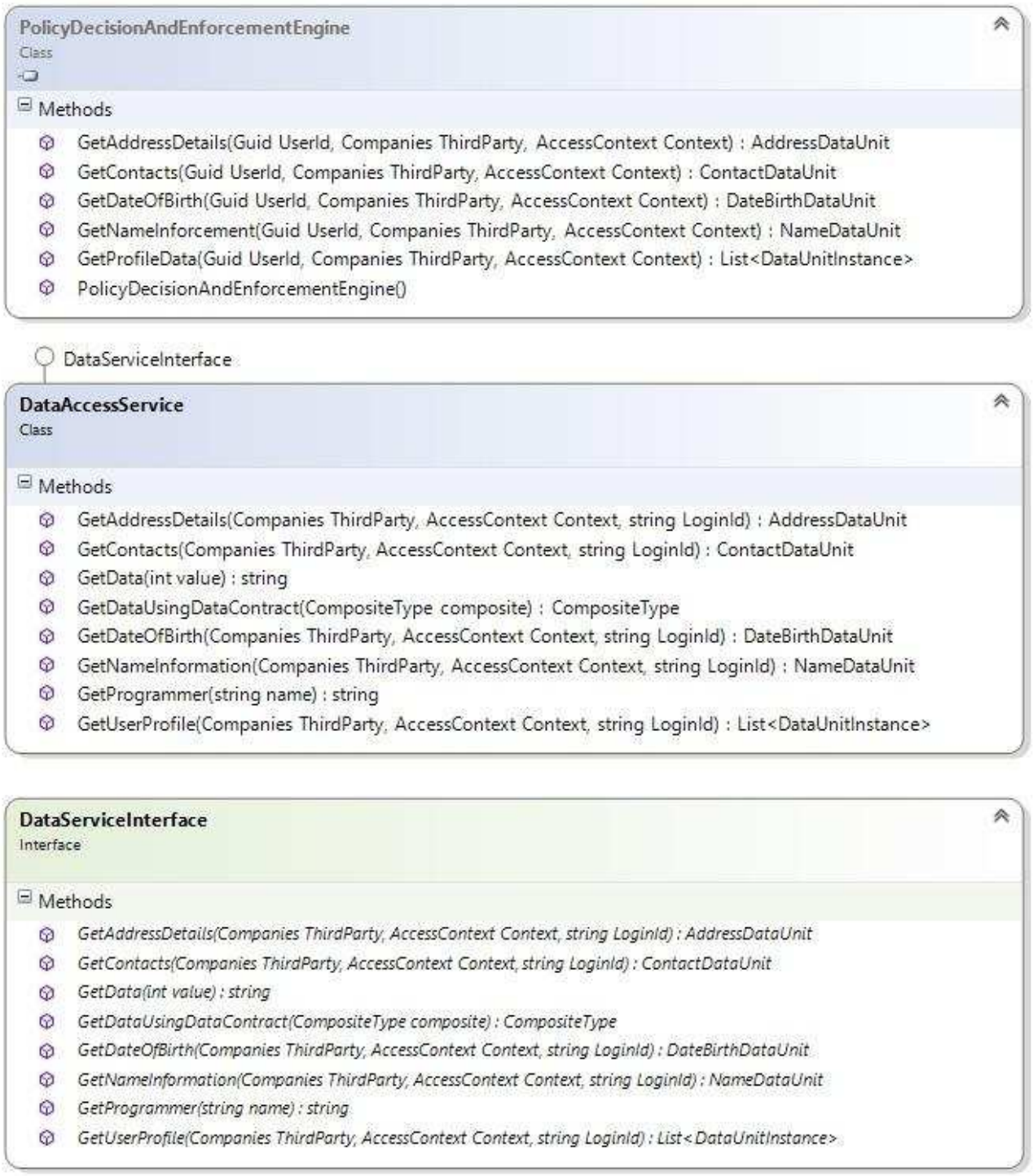

Figure 7: Class Diagrams showing Policy Decisions & Enforcement Engine and Data Web Services Modules

This component is a web service and is designed to serve as a gateway for third party application to connect to and access shared data in this system. Web services allow software applications to communicate and are designed to support interoperable machine-to-machine interaction over a network setup. The class diagram in Figure 7 shows its interface and the

service class exposing operations to third parties. This service uses SOAP technology to convey messages over an HTTP connection. Trusted Third parties are able to request for a data unit instance of a user data or the entire profile data by providing the user login identity and specifying the requesting third party and the intended usage context or purpose. This component then calls for the user login identity to be translated into the system internal unique id. The encryption object for the required user is then made available to the policy engine for the processing of the data request. The same data unit object returned to this service by the policies engine is returned to the calling third party.

CONCLUSION

This study was able to highlight the many data security vulnerabilities in the cloud environment as well as identify data control and the use of sticky policies as top of the measures for protecting data resources in the shared cloud infrastructure. A data governance solution for the management of users' profile was also designed and implemented, thereby demonstrating the use of these privacy policies to deliver the ultimate data control power to the data owner.

REFERENCES

1. Gartner Inc (2011). Gartner identifies the Top 10 strategic technologies for 2011. Accessed on 27th May, 2014 from http://www.gartner.com/it/page.jsp
2. Komu, M., Sethi, M., Mallavarapu, R., Oirola, H., and Khan, R. Secure Networking for Virtual Machines in the Cloud. Accessed on 29th May, 2014 from www.cloudsoftwareprogram.org
3. Trabelsi, S., and Sendor, J., (2012) "Sticky Policies for Data Control in the Cloud." Tenth Annual Conference on Privacy, Security and Trust.
4. Ryan, M.D., (2013), Cloud Computing Security: The Scientific Challenge and a Survey of Solutions. J. Syst. Software (2013), http://dx.doi.org/10.1016/j.jss.2012.12.025
5. Armbrust, B., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., et al. (2010). A View of Cloud Computing, *Communications of the ACM, Vol 53*, 9.
6. Descher, M. Masser, P. ,Feilhauer, T. Tjoa, AM. &Huemer, D. (2009), Retaining Data Control to the Client in Infrastructure Clouds, IEEE, DOI 10.1109/ARES.2009.78
7. Jensen, M., Schwenk, J., Gruschka, N., &Iacono, L. Lo. (2009). On Technical Security Issues in Cloud Computing. *2009 IEEE International Conference on Cloud Computing*, (2009), 109–116. Do: 10.1109/CLOUD.2009.60
8. Dogo, E.E., Salami, A., and Salman, S.I.(2013). Feasibility Analysis of Critical Factors Affecting Cloud Computing in Nigeria. International Journal of Cloud Computing and Services Science. 2013; 2(4): p.276-287.
9. Ume, A. (2012). The Fear and "Phobia" of the Cloud and Cloud Computing. Journal of Educational and Social Research. 2012; 2(8); p.147-154.
10. EnCoRe Project, (2012), Ensuring Consent and Revocation-Technical Architecture, Accessed 12/5/2013 from www.encore-project.info
11. Marco, M.C., Siani, P., and Kounga, G. "Enhancing Accountability in the Cloud via Sticky Policies," Secure and Trust Computing, Data Management and Applications, vol. 187, Springer, 2011, pp. 146-155.
12. BCS (2010). Embrace the Cloud. Accessed on 15/5/2014 from

www.bcs.org/contant/conWebDoc/34794
13. Trabelsi, S., Njeh, A., Bussard, L. and G. Neven, (2010), PPL Engine: A Symmetric Architecture for Privacy Policy Handling, W3C Workshop on Privacy and data usage control.
14. Ramgovind, S., Eloff, M.M., and Smith, E. (2010) "The Management of Security in Cloud Computing" IEEE.
15. Cloud Security Alliance, www.cloudsecurityalliance.org Last Retrieved: 19/08/2013
16. Shucheng, Y., Cong, W., Kui, R. and Wenjing L. (2010). Achieving Secure, Scalable, and Fine-grained data Access Control in Cloud Computing. Accessed on 28/5/2014 from www.leonsoftsolutions.com/ieeepapers
17. Microsoft Corp (2014). Trustworthy Computing/The Microsoft Approach to Compliance in the Cloud. Accessed on 10/5/2014 from download.microsoft.com/download.
18. Thomas, G. (2011-11-1). Goals and Principles for data governance. The Data Governance Institute (web site). Accessed on 10/2/2014 from http://www.datagovernance.com
19. Siani, P., and Mont, M. C., (2011). "Sticky Policies: An Approach for Managing Privacy Across Multiple Parties" IEEE Computer Society, Springer-Verlag Berlin Heidelberg.
20. Qiang, J, (2008), On Using Encryption Techniques to Enhance Sticky Policies Enforcement
21. Karjoth, G., Schunter, M., Waidner, M.,: Privacy-enabled services for Enterprises. In: Proc. 13th International Workshop on Database and Expert Systems Applications, Sept. 2-6, 2002 pp.483-487 (2002).