Indian Institute of Management Calcutta

Working Paper Series

WPS No. 763
June 2015

Information Retrieval as a Service – IRaaS : A Concept Paper on Privacy Analysis

Asim K. Pal
Professor, Management Information Systems Group
Indian Institute of Management Calcutta
D. H. Road, Joka, P.O. Kolkata 700104, India
http://facultylive.iimcal.ac.in/workingpapers
asim@iimcal.ac.in

S. Bose
Professor, Head Dept. Computer Science and Engineering
Neotia Institute of Technology, Management and Science
D. H. Road, Jhinga, PO: Amira, South 24 Parganas, WB–743368, India
subratabose@yahoo.com

Information Retrieval as a Service - IRaaS: A Concept Paper on Privacy Analysis

A.K. Pal[1] and S. Bose[2]
[1]Indian Institute of Management Calcutta, Kolkata, India
[2]Neotia Institute of Technology Management and Science, Kolkata, India

Abstract

Privacy analysis has not always got proper attention in the literature often overtaken by security algorithms. This work attempts to fill in this gap. The strength of this work we believe is in the privacy analysis conducted in depth for a complex problem following an objective method. Providing information retrieval service from multiple heterogeneous autonomous data sources is a natural requirement. Queries may appear within an application framework or may be ad hoc. The service must meet the basic characteristics of cloud computing such as scalability and ubiquitous network access. We recognize this as *IR as a Service* (*IRaaS*), and propose different models - Open Access, Closed Access, Collaborative and Enterprise IRaaS. Such services should win the *trust* of all participants by enforcing required *security* as per *privacy* requirements. Explicit privacy statement going into depth on the need of sharing of information and computation which is agreed by each party helps secure computation through developing mutual trusts for the entire process of computation and communication. Privacy need of IRaaS is modelled by six *privacy issues*: *identity*, *schema*, *data*, *result*, *query* and *query distribution*, each issue analyzed on *privacy protections* involving information sharing among the parties. To limit flexibility we apply dominance relations between privacy issues across privacy protections and vice versa. We suggest a privacy algebra and apply it to IRaaS for consolidation. Finally a skeleton IR framework for IRaaS with an algorithm for secure query processing using the privacy model is proposed.

Keywords: cloud computing, data mediation, information security, database query processing, collaborative computing, multiple databases, privacy algebra

1 Introduction

The phenomenon of accessing multiple heterogeneous independent and autonomous data sources for the purpose of seeking information is rapidly gaining coinage because of proliferation of internet across all geographies and all societies. With the emergence of cloud computing along with the associated growth of mediation services in different areas of life as well as a large amount of database research work done for data integration and querying from multiple heterogeneous data sources through mediation [2-6, 9, 51], a lot of work done for security in databases [7,8,47] and privacy preserving query processing [10-16, 40, 44, 48, 50] and also the development of the concept of cloud database [17-20] and collaborative cloud computing [39, 42-44] the platform is perfectly set for *Information Retrieval as a Service* (IRaaS) as another highly prospective cloud based service. A few examples of information retrieval (IR) from heterogeneous data sources are: looking for houses along with the landscape / building plan and apartment interior design from real estate agents within a budget range and having certain demographic features; striking deals with online shops for purchasing dresses for a dance troupe going on a world tour; seeking suitable partners from different match making sites (with samples of cultural orientation of the candidates displayed – writing, painting, song, dance or drama); finding a useful degree from education providers; drawing up a suitable tour plan; searching for a location to start a factory; discovering fraudulent transactions across the globe connected with a group of suspected criminals, locating a terrorist on move, and looking for a criminal given voice samples or finger prints. One could be as esoteric as to find out the evolution of architectural types in modern India.

A good implementation of IRaaS would need a proper marriage of several wide ranging concepts and technologies including cloud computing, distributed computing, collaborative computing, database search, cryptography and information security, multimedia computing, agent based computations, web services, ontology and query semantics, cost and revenue sharing mechanism and pricing. For such a service to be successful the system need to win the trust of all the participants by ensuring that all the security mechanisms are in place. To optimize the security from the viewpoint of implementational feasibility within the constraint of cost and time, a practical system needs to be developed which will take into account the privacy concerns of all the participants. As noted above there is already a vast literature on distributed databases and querying on multiple databases even some on heterogeneous databases, security in databases, privacy preserving query execution and cloud databases and security in cloud, collaborative cloud computation and also some recent attempts in addressing

privacy concerns of the participants (see section 1.6 below). But there does not yet seem to be any concerted attempt to *holistically* look into the problem of privacy from the perspective of the users who are concerned with security as well as are interested to make the best possible use of their resources by giving maximum possible access and would want to impose some trust on the service provider and other parties. This means that we need to make privacy as far as possible an explicit function of the choice of users (both the information seeker and information providers and the intermediary) on various aspects of communication and computation depending on the requirement of the problem.

The current work has twin objectives. One is to integrate the database concepts, distributed computing, cloud computing (which is a centralization concept opposing distributed concept) into a day to day to information service at different levels of the society (from amateurish query making to sophisticated problem solving by making query on heterogeneous data spread across the globe). The other one is to emphasize somewhat neglected area of privacy analysis being often over shadowed by or confused with the security protocols, forgetting the need of expression of privacy by the collaborating (or, even competing) users at the ground level who do not know each other and thereby enhance trust in the systems people participate in. We demonstrate how a complex problem of privacy analysis can be simplified using simple mathematics.

This work attempts to dissect the complex web of privacy, security and trust and aims to focus on the process of privacy modelling by looking deep into the issues of privacy and protections from the concerns expressed by different parties for executing a query which seeks information from multiple heterogeneous sources with the help of a service provider who connects the data sources relevant for a query The privacy need of IRaaS is modelled by dissecting the privacy into six dimensions, called *privacy issues*, namely, *identity*, *schema*, *data*, *result*, *query* and *query distribution*, and each dimension being analyzed on a number of *privacy protections* – each protection refers to information sharing between two parties. Then we look for interdependencies linking the privacy issues. We have looked into *dominance relations* between any two privacy issues across privacy protections and vice versa which limit the privacy flexibility. We have developed a privacy algebra which simplifies the privacy analysis. We have investigated a few scenarios of the privacy requirement. But before we go into the privacy model we look into IraaS which was introduced in by Pal and Bose [1]. We look into the rationale of IRaaS, in the context of multiple-database search, cloud computing and cloud database. We develop taxonomy for IRaaS. Particularly, we make two broad categories, Open Access IRaaS (OA-IRaaS) – which accepts arbitrary query from arbitrary domains and Closed Access IRaaS (CA-IRaaS) – which allows seeking information within a given domain or application. We suggest collaborative IRaaS as a major area of development for both the open and closed models. Given the current

state of technology we believe that CA-IRaaS is more feasible and practical at this point of time. Our privacy model concerns this information service.

The paper is organized as follows. Section 1.1 to 1.3 discusses the importance, feasibility and rational behind IRaaS being a cloud based service, the role of mediator for IRaaS is discussed at section 1.4, the privacy need of IRaaS is discussed at section 1.5, section 1.6 offers the summary of literature survey in the related areas of work, while section 1.7 summarizes our contribution in this paper. The taxonomy of IRaaS is placed at Section 2. In section 3 we have made an in-depth analysis of different privacy issues of IRaaS. Interdependencies of the privacy issues and their simplification with the help of *privacy algebra* have been discussed in section 4. Section 5 offers the secure query processing algorithm and the processing framework of IRaaS. Section 6 concludes with suggestions for future work.

1.1 Why should IRaaS be a cloud based service?

IRaaS is seen as an application service of information retrieval which requires assimilating data available with other data owners. These data sources can be anywhere be it in the same cloud as the service provider, different cloud (collaborative), different databases who lends their data from their own premise but not in cloud. A variety of issues are of concern for such multi data source retrieval, namely, the ease of making a query, maximizing voluntary participation of relevant data sources, maintaining independence of operation of individual data sources, efficiency and accuracy of the results obtained through merging and mixing (sometimes very complex) of results obtained from separate sources, secure operations of the data sources, managing heterogeneity of data sources at different levels, privacy and trust issues vis-à-vis the information seeker, the mediator and the individual data sources (information providers). There could be other issues such as financial, e.g. cost or revenue sharing patterns among the parties which will be manifested through the pricing structure for the services rendered to the information seeker and profits shared between the mediator on one side and the data sources on other. An important task for the mediator would be to establish relevant data sources for a given application or a given query, and also to manage entry of new data sources or exit (if announced) of existing data sources. Further, it has to accommodate any change in the schema of a data source. Finally, scalability, elasticity of demand and performance of the IR service will be a crucial issue for the mediator [9] to remain successful in the business. The scalability issue is not just limited to the volume of data that is retrieved or moved in the process of computation needed for a given query or frequency of queries, but more importantly the performance issue could arise from heterogeneity factors. In the emerging scenario of growing popularity and highly scalable property of cloud computing it is only natural for the IR service provider to use a cloud based service. This is the very reason that we have used the term IRaaS for the implied IR service and envisage this as an example of SaaS. Dynamic

collaboration capability of the cloud could be another reason. But as such there is no pressing need for IRaaS to be cloud based if it is meant for a simple application. In our way of thinking IRaaS is supposed to be highly flexible from small to giant.

1.2 Why is IRaaS a SaaS?

Cloud computing offers different types of services over internet based on on-demand service requests of users on pay-as-you-use basis, prominent among them being IaaS, PaaS, SaaS and DaaS. It incorporates Software-as-a-Service (SaaS) providing common business applications online through web browsers to satisfy the computing needs of users, while the software and data are stored on the servers. Typically SaaS can be defined as software deployed as a hosted service and accessed over the Internet [30]. Conventional SaaS services can be generally put into two categories - business software which provides business solutions, such as ERP, SCM, CRM, etc. and consumer software which provides personal solutions, such as office applications [31]. In this paper we have offered a privacy model for information retrieval service to answer a user's query made to a service provider. This service provider has been positioned as a Cloud Service Provider (SP) who provides information retrieval service to its customer by integrating data from multiple heterogeneous data sources. This service is a variety of SaaS though it is not a SaaS in conventional way. In SaaS service, a provider hosts an application centrally and delivers access to multiple customers over the Internet on payment basis. While doing so the customer generally hosts the data in the cloud itself along with the application. A variety of security mechanisms are used to keep sensitive data safe in transmission and storage. In IRaaS however the SP acts as a coordinator, bridge between the customer and the data owners, it hosts and executes the application but the service being information retrieval SP has to use data of different data owners. These data owners may also be positioned as cloud holding data of data owners who wish to participate in IRaaS. We envisage that all the parties involved in this service will act according to the privacy preferences of each of them and thus calls for a comprehensive privacy scheme (model) encompassing all possible communications between any two parties and all possible privacy issues like identity, data, schema, query and query result.

1.3 Collaborative Cloud

Collaborative computing enables users to work together on documents and projects, usually in real time using network communication systems. A good example of collaborative applications designed for Internet use are Microsoft's NetShow and NetMeeting. Collaborative document creation [41] in Google Docs is a simple example of collaborative computing in cloud where several people can work on different parts of the same document at the

same time. Proprietary nature of existing cloud service providers restricts consumers to use multiple cloud services simultaneously for the same problem. Collaborative cloud computing for software services enables customers to have better access to software, computing facilities, and data and also create more business opportunities [31, 42]. For example, a snapshot of customer's data from various data sources would help a user access to information which would have otherwise been difficult for him to assimilate. This could be as open as train or flight information or as restricted as financial records or crime records, etc. With democratization and collaborative cloud computing information can be obtained dynamically as per the arrangement and need of the business. Yoon et al. [42] presents a mathematical model for dynamic collaboration of cloud service providers for auction market to offer collaborative services to its customers. Formation of the collaborators is initiated by one of the providers who act as a primary CP to form a virtual organization with other collaborators for providing a set of services to its customers. Karnouskos et al. [43] proposed a SOA based service architecture for industrial automation. The proposed architecture will offer a collection of services providing common functionalities, interact with each other and form a cloud of services which need to be collaborative. Query executions in a collaborative cloud [39] in which different parties need to release information and cooperate with others require protection of sensitive information. The data source participating in such systems could be completely independent, federated or a centrally planned distributed database system. Query processing in such a scenario should support selective sharing of information by different data owners (similar to restrictive view, authorizations and access restriction mechanisms in relational databases) as per their access authorization to different players. The problem thus requires a solution that helps capture different data protection needs of the cooperating parties. S. Vimercati et al. [44] presents an approach for the specification and enforcement of authorizations regulating data release among data owners collaborating in a distributed computation, to ensure that query processing discloses only data whose release has been explicitly authorized. The authors also present an algorithm that determines whether a given query plan can be safely executed and if so produces a safe execution strategy. Answering queries with access restrictions has been studied extensively in the literature [45].

1.4 Mediator for IR service

Let us now focus on the job of mediation performed by an IRaaS provider. We have to assume that the service provider is adequately knowledgeable about the data sources required and resourceful and trustworthy to connect them. It is very much possible that the service provider looks for appropriate data sources by using his or her contacts, by searching through the net, or inviting for participation (possibly through a bidding process), etc. Data sources would join the provider depending on their interests, their knowledge

about the provider and also based on the amount of trust they have on the provider and finally establish a business deal with the provider on revenue sharing and pricing schemes, etc. Ultimately, a list of data sources (information providers) becomes part of a given IR service. But this list will occasionally change, depending on entry of new sources or exit of old sources. Having established the data sources the service provider collects meta information about the exposable data of each data source. The data could be heterogeneous in a number of ways, the content of data (text, audio, video), formatting of individual data elements, and data structure (e.g. flat file, relational database). The meta information of a data source contain data about the attribute details such as their names and data types. The mediator will construct a global schema by combining the individual schema, a set of mapping rules and rules for semantic integration to reconcile the similarities and differences [9, 51]. This will be used for creating a uniform interface for inputting a query by the information seeker. The global schema in the data integration may be an ontology, which can act as a mediator to reconcile the heterogeneity between different data sources [21, 51, 52]. For each data source the service provider creates a wrapper which basically acts as an interface between the *mediator engine* (often a relational database manager) and the data sources. The customer query is posed to the mediator which acts as a central system with interfaces to the autonomous wrapped data sources for the information retrieval service.

1.5 Rationale behind privacy in IRaaS

IRaaS is a cloud based service, though technically it is not mandatory that the service provider has to be hosted in cloud only. However for such a system to work the application has to be web based and its natural positioning is in cloud when it is seen as a gigantic resourceful service. Thus its security concern remains be it simple web based or cloud based. As such there is no need to differentiate a web based service from a cloud based service as far as privacy is concerned. The privacy concerns of the participating parties remain the same irrespective of clouds hosting the customer (querrier), the data sources or even the service provider. The primary source of concerns emanates from the intentions of the respective parties, which has nothing to do with cloud. The security implementation of IRaaS may have to deal with this issue more explicitly; particularly the clouds hosting these participants can not be guaranteed to be independent. From the users' perspective, security concern is a major barrier for the adoption of cloud computing. According to a survey from IDCI in 2009, 74% IT managers and CIOs believed that the primary challenge that hinders them from using cloud services is its security issues. In another survey carried out by Gartner in 2009, more than 70% CTOs believed that the primary reason not to use cloud services is data security and privacy concerns. Over and above this openness and multi-tenancy of cloud adds more to the security concern among its clients [19, 32]. In IRaaS the querier (customer) and the data owners like to

protect their identity and respective assets from each other including SP and thus call for privacy concern among them. Privacy and security of cloud computing is a legal requirement [33, 34]. Systems for electronic health record (EHR) in the US are constrained by federal regulatory legislation and oversight law which basically focuses on security and privacy. Therefore, EHR built with the cloud computing model will need compliance of such regulations [33]. In his work, Acquisti [35, 36] makes economic analysis of privacy, discusses issues related to protection or non protection of user's identity vis-à-vis its economic impact. Although technology for privacy preservation is not a barrier but economic consideration of privacy impacts its marketability. Different parties might have conflicting interests and views about the amount or items of information to disclose during a certain transaction, Also an individual might face trade-offs between her need to reveal and to conceal different types of personal information [35]. Trade-off lies in the domain of economics. This comment justifies the need for availability of different privacy types to the players during execution of an application over internet in cloud to protect privacy of data, query, identity of the parties, result of query etc. in IRaaS. Economic studies [37] have shown that at times un-protecting privacy makes sense and benefit a buyer in an online purchase when information about customers' tastes and purchase history is available and can be shared among sellers. To check the integrity of outsourced data users can even use a third party auditor (TPA) to perform privacy preserving public auditing of the outsourced data [38].

To summarize, security is of paramount importance in cloud because it needs to be trusted by its client for everything they do in it. Same concern even applies to other web based services such as email or social networking sites, etc. If the service provider is not cloud based security concerns remains probably more in the users' minds because IRaaS provider requires to connect and work with third party data sources. Even for a standalone service provider which just gives this service for a client, the client may have more control on him but still needs to manage the data sources. Moreover to handle heterogeneity of data our system proposes use of mediator which should be able handle all types of query but privacy on the top of mediation will put challenges which will vary from case to case. However *fully homomorphic encryption* which can compute any arbitrary function [49], once put into real life use can possibly be the breakthrough for full-fledged implementation of IRaaS as far as query complexity is concerned.

Here we are often interested in anonymous communication, in which user's IP address and any other personally identifiable information are concealed from the server hosting the website visited by the user and some system encrypts the traffic between the user and the service. There are many ways of accomplishing anonymous web browsing. There are proxies that are usable, as well as programs such as TOR (The Onion Router), which sends information through a net of routers to hide the destination of information. Tor is a widely used anonymous communication system. Most deployed

systems for anonymous communication have a centralized or semi-centralized architecture, including Anonymizer, AN.ON, Tor, Freedom, Onion Routing, and I2P.

1.6 Related work

Providing IR service from the data owned by different independent and autonomous data sources demands integration of heterogeneous data lying in multiple servers. A number of approaches have been proposed in the literature, mediator based approach being the most prominent among them [51]. In a mediator system user queries made on a single schema is reformulated into queries on the local schema of the respective data sources containing the actual data. This arrangement perfectly suits IRaaS, with the required flexibility of known static data sources or unknown dynamic sources operated on one or many collaborative cloud computing infrastructure. The global schema provides a reconciled, integrated, and virtual view of the underlying sources. Two most important approaches [9] to establish mapping of global schema to local schemas are *global as vie*w (GAV) [6] and *local as vi*ew (LAV) [24]. In the global as view every component data in the global schema is associated with a view over the source local schema. Therefore querying strategies are simple, evolution of the component databases are not easily supported. The local approach however permits global schema to be defined independently from the sources, and the relationships between them are established by defining every source as a view over the global schema but query processing can be complex. GAV is preferred when the sources being integrated is known and stable, whereas LAV is considered suitable for large-scale ad-hoc integration. Based on architecture, there are two kinds of integration systems - central data integration systems [3-6, 22, 51] and *peer-to-peer* (P2P) data integration systems [23]. Mediator based system falls under central data integration system. A central data integration system has a global schema and thus provides the customer with a uniform interface to access information stored in the data sources. In a P2P data integration system, there is no global point of control on the data sources (peers). Instead, any peer can accept customer queries for the information distributed in the entire system [21]. The work of Sheth [3] is a classic approach towards building a single global schema for data integration encompassing the differences among the local database schemas. The Pegasus system proposed by Ahmed et al. [22] offers a SQL-like language, HOSQL, a unifying data definition and data manipulation language to map local schema of individual databases to the global schema (global as view approach). Another approach known as federated approach [25, 51] relies on multiple import schemas and customized integration at each multi database level enforced by the system. The work of Lakshmanan *et al.* [5] extends traditional SQL syntax to a language schemaSQL to support querying data and metadata in a

heterogeneous multi database system. MD-SQL [26] is a similar work allowing querying data and metadata in a multi database system. The Distributed Interoperable Object Model (DIOM) [4, 6] offers a query mediation framework through an adaptive approach to interoperability instead of an integrated global schema. The DIOM project [6] offers a framework for integration of relational data sources with a centrally performed compilation process [9]. Its main features include information access through a network of *application-specific mediators* which is also aimed for IRaaS implementation. Semantics is an important component for data integration which has led to the inception of ontology-based approach. The pioneering work of Doerr et al. [52] focused on semantic integration and use of ontology for mixing heterogeneous schema across multiple sites. Their efforts have provided a new dimension for information integration.

Privacy is a serious concern in IRaaS and thus privacy preserving techniques for query processing is of significance. Privacy preserving techniques have been applied to a number of different areas like information retrieval [27], data anonymization [28] and data mining [29]. The specific task of privacy preserving query processing over distributed databases has been studied extensively in the past. Chow et al. [10] proposed a computation model comprising of two semi honest parties other than the customer and the databases. The model supports data privacy and result privacy but does not consider query privacy. Scalability of query computation over large databases was their focus area. The work of Emekci *et al.* [11] proposed a model of query computation by third parties in a hash-based P2P system. Their model considers data privacy but not query privacy. Agrawal *et al.* [7] developed protocols for intersection, intersection size and equijoin database operations for two databases using commutative encryption and hashing. Aggarwal et al. [8] proposed a two-party storage model to enable secure database query service for outsourced data on a single database in a distributed architecture. The work of Hildenbrand et al. [16] proposes an encryption scheme named POP to keep encrypted data in cloud and process a range of SQL queries on the encrypted data. It can be implemented on the top of a relational database system or database as a service like SQL Azure or Amazon RDS. Homomorphic encryption is a common technique for preserving data privacy and query privacy in privacy preserving query processing [13, 40, 48, 50]. Though the theoretical breakthrough in *fully homomorphic encryption* by Craig Gentry [49] allows computing arbitrary function on encrypted data it is yet to be seen practically implementable. Haibo Hu et al. [13] proposed a framework of three parties - the data owner, the querying customer, and the cloud service provider with the objective of data and query privacies in a single database system. Shiyuan et al. [40] protects privacy of user's query data and plaintext part of the query where user is querying a public data store. They work on range queries and join queries. In [48] the authors analyze the use of fully homomorphic encryption for solving complex selection, range, join or aggregation query on encrypted data. Yubin et al. [50] provides a solution to preserve privacy of both data

owners and query users using classical homomorphic encryption on database NoSQL which is less structural than relational.

Privacy has been extensively studied in the literature. Privacy has no fixed definition; it is generally associated with the collection, use, disclosure, storage, and destruction of *personally identifiable information* (PII). Identification of private information depends on the specific application scenario and the law, and is the primary task of privacy protection [32]. A cloud service allows users to interact with (possibly) unknown parties to access services. Such interactions may reveal to a malicious observer (or to the server itself) private information about the user, who may not like to disclose her identity to gain access to the service of interest. This problem requires the adoption of appropriate techniques supporting the *anonymous interaction* of users with remote servers [39]. In his work, Acquisti [35] distinguishes between individual's *offline* and *online* identities. Online privacy is related to individual's privacy in online transactions such as in e-commerce. Offline identity represents individual's actual identity such as SSN, name, address etc. Both online and offline identity privacy are of importance to us. User's on-line identity can be hidden if she does not create profiles on the server nor has she cookies enabled, and she does not disclose her IP addresses to the server (e.g. via the use of a proxy) [35, 40]. Enforcement of privacy policies to restrict disclosure of PII in PaaS application has been studied by Yu et al. [46]. In the use case mobile application end-user preferences regarding handling of the PII are captured through a privacy policy language for privacy specification and then enforced in the PaaS application. In cloud environment massive data is outsourced to a number of resources for storing and processing. These data could be dynamic and potentially sensitive. In another direction of work, design of an optimal outsourcing arrangement given a set of dynamic data with potential confidentiality and privacy constraints, a pool of resources, and an estimated query workload (static or dynamic) has been studied by Li et al. [47]. The arrangement consists of proper encryption, fragmentation (horizontal or vertical), and synopsis outsourcing that minimizes the cost associated with data shipping and processing for the given workload.

Anonymity is a need in open media like internet. However, costs of adoption of anonymous systems might be high [35]. Exchange of messages between two parties even when the content of the messages is kept secret may compromise their privacy. In IRaaS if an observer knows that a user (police or detective department) is querying SP for the query service and SP in turn is gathering the result from some data owner which is a financial institution, the observer can infer that the user is querying about some financial information. In this case, it is necessary to protect the relationship between a user and the queries that she sends. Anonymous communication protocols for providing anonymity to mobile users have been proposed in the literature, specific solutions specific to cloud systems are also being investigated [39].

1.7 Contribution

This is a novel attempt to combine the powers of a) cloud computing concept, particularly its SaaS, for scalability and capability to handle complexity, b) distributed computing as a concept for the distributed processing of a complicated information retrieval task, c) data mediation task, d) privacy modelling coupled with security and trust issues to achieve a ubiquitous *Information Retrieval as a Service* for multiple independent and autonomous heterogeneous data sources. We have tried to establish the logic of IRaaS as a cloud based service. A taxonomy has been proposed to suggest two broad categories, Closed Access IRaaS (CA-IRaaS) and Open Access IRaaS (OA-IRaaS). The former one is targeted to a set of applications or an application area where the client data sources are pre-fixed, while the latter one is much more open in its depth and coverage. The taxonomy also delves into collaborative IR services, besides looking into the hierarchy of application areas as the focus of the IR services. Then the work discusses how privacy, security and trust play together a vital role for IR services, mainly for CA-IRaaS. For IRaaS the privacy issues have been discussed at great length, e.g. how different privacy issues are interlinked. A privacy algebra has been suggested to process different privacy issues and privacy protections to enable one to come up with a comprehensive privacy view which is negotiated and agreed across all parties (data sources, the querrier – customer and the service provider). This algebra has been demonstrated on IRaaS. A secure IR framework along with a sketch of the security protocol for IRaaS (including query processing) has been provided. The strength of this work we believe is in the privacy analysis conducted in depth for as complex a problem as IRaaS following an objective method suggested in the work itself. Privacy analysis has not always got proper attention in the literature often overshadowed by security algorithms. This work attempts to fill in this gap.

This paper is based upon Pal et al. [1]. But the current work has added strength in wider coverage as well as higher depth. The background and motivation for the work has been strengthened by making fresh literature search and expanding the scope of the literature review. The IRaaS taxonomy has been sharpened. The privacy analysis has been widened as well as heightened to a significant extent. A new section has been added on Privacy Algebra. The IR security framework and sketch of the security protocol has been revised.

2 IRaaS – the Taxonomy

Let's now try to focus on who the possible customers are for these kinds of IR services. The orientation and objective of the customers could vary a lot. On one extreme there could be one-off customers who are interested in arbitrary queries, like the ones we make to a keyword based search engine,

e.g. show me the architectural types of Calcutta during the British rule of India, or show me the most memorable tragic scenes from Charley Chaplin films. These can basically be referred as *Open Access IR Services* (OA−IRaaS), where the mediator has to retrieve data from dynamically selected data sources. A good (general purpose) OA−IRaaS provider hence needs to be very powerful, something like a combination of Google search engine, YouTube and a few database search engines. We may be a decade or so away from that to happen in a way similar to the kind of services one now gets from the Google search engine. One can think of discipline oriented OA services, e.g. arts−OA−IRaaS, history−OA−IRaaS or crime−OA−IRaaS. The arts−OA−IRaaS provider would be interested and knowledgeable in different forms of arts. Based on a query it will have to collect information from arts related data sources on the spot. The efficiency and practicality of such an open access system at the current stage of development of technology, even if for a single discipline, seems farfetched because such a provider cannot be very resourceful going by the logic of business. One can also think of more specialized OA IR services, like film−arts−OA−IRaaS or paintings−arts-OA−IRaaS. Yet these will not be easily available on a realistic scale in the current scenario. On the other hand there will be more realistic *Closed Access IR Services* (CA−IRaaS) which are centred around specific applications or business interests. For examples, there can be services meant for commercial banks in India (bank−CA−IRaaS), for police or criminal investigation departments in different states in India (this will help catching a criminal who commits a crime in one state and runs away to another state) – Police−crime−CA−IRaaS and CID−crime−CA−IRaaS, for travel agents or tour organizers, and for managing land use for different purposes (this helps in the development of land map for industries and other projects in a land scarce country like India). The data sources would be restricted according to the discipline, market or geography concerned for the specific CA−IR provider and these will be registered with the latter. A service provider in this category would have to be properly knowledgeable in the given area of specialization. Figure 1 illustrates the taxonomy of general IR service.

But, sometimes depending on the information sought the service provider may have to cross boundaries of disciplines or geographies, which might imply one of the two things – either service providers of different disciplines or geographies collaborate (Collaborative CA−IRaaS, in short Coll−CA−IRaaS), or one treats this as a Mixed−IRaaS by exposing a subject devoted IRaaS beyond its own data sources to other dynamically selected open data sources as needed by the query. Coll−CA−IRaaS could again be of two types: CA−Coll−CA−IRaaS and OA−Coll−CA−IRaaS, depending on whether there is any pre−arranged set up of CA−IRaaS providers or not. Thus in case of a CA−Coll−CA−IRaaS a set of CA−IRaaS providers will have a contract for serving customers jointly, if needed. Say Police−crime−CA−IRaaS and CID−crime−CA−IRaaS (for criminal investigation department) may collaborate with pre−arrangement to make it Police+CID−CA−Coll−crime−CA−IRaaS, a joint IR provider. Or, they may

collaborate occassionally and have a loose collaboration system Police+CID−OA−Coll−crime−CA−IRaaS. Note, a Police+CID IR service will exist along with Police IR and CID IR services. Similarly, there can be USA+Canda−bank−CA−Coll−CA−IRaaS. Figure 2 illustrates the taxonomy of collaborative cloud.

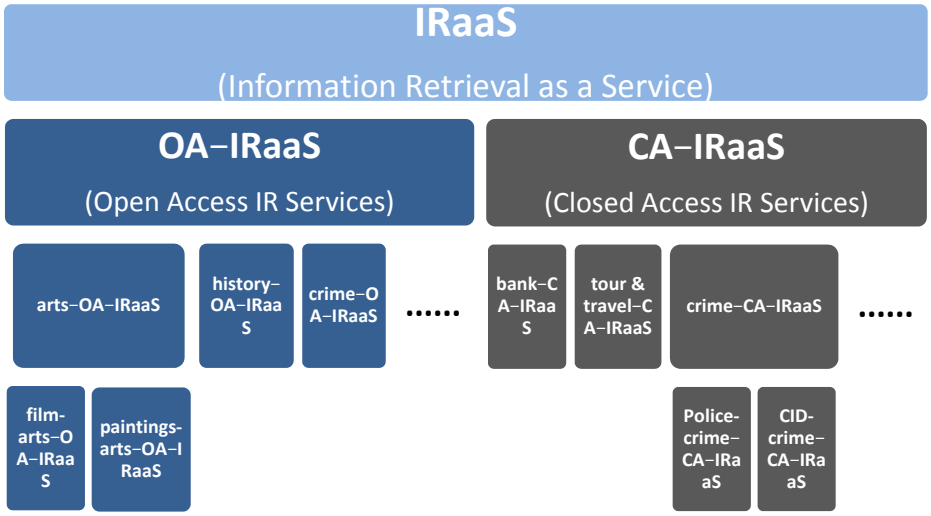

Fig 1. Hierarchy of Information Retrieval as a Service [IRaaS]

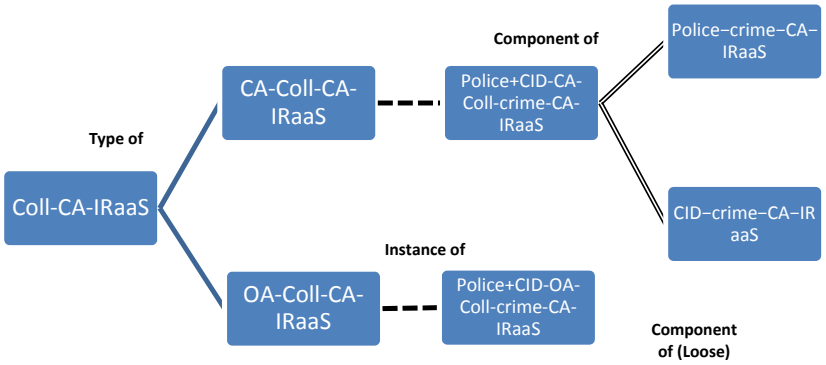

Fig 2. Collaborative Closed and Open Access IR Services

There can be many issues of privacy here, such as which data items can be freely shared across two systems and which can be shared anonymously or which cannot be shared at all. This can easily be confused with *composable mediators* (mediating the mediators) [9], because here we are talking of service level collaboration, which is basically an organization level collaboration where the latter is about software level collaboration. It is though of course true that service level collaboration will need software level collaboration at some level, but there will be many other extraneous issues

there, e.g. cost sharing or privacy issues. There is still another possibility of application of the idea of CA−IRaaS, which is meant for enterprise applications, enterprise−CA−IRaaS, one instance is for one enterprise, e.g. WM−enterprise−CA−IRaaS for Wallmart, or more narrowly, Mexico−WM-enterprise−CA−IRaaS. It is possible that WM−enterprise−CA−IRaaS is same as USA+EU+Mexico−WM−enterprise−CA−IRaaS, assuming that WM is spread across theses zones. Figure 3 illustrates the taxonomy of an enterprise IR service. A corporate can benefit a lot from such an IR service meant for its own organization, processes, employees, customers, etc by integrating information across the enterprise from heterogeneous data sources. Actually one can think of redesigning their existing ERP systems in view of these kinds of new enterprise based services. And, from the business point of view Enterprise IR services appear to be highly effective for corporate, particularly the big ones. And these can based on the company's private cloud. Further, collaborative cloud computing could be put to good use for developing collaborative IS, both OA and CA types.

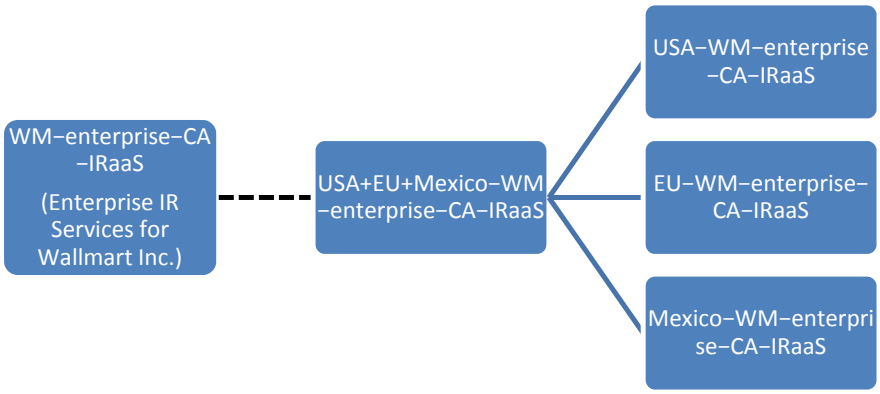

Fig 3. Enterprise Closed Access IRaaS for Wallmart

3 IRaaS – the Privacy Issues

In CA-IRaaS model a set of autonomous data owners (data sources) having independent operations allow the *mediator* (the mediating agent on behalf of the service provider who offers the IR service to its information consumers or customers) to access their data based on the query made. In OA-IRaaS model which is more general or flexible a customer makes an arbitrary query for which the IR service provider searches for potential data sources that are relevant to the query and solves the query with the help of volunteering data sources. For the purpose of privacy and security concerns we treat the mediator as an *untrusted third party* (*utp*). We also assume here a *semi honest* or *honest but curious* model for the privacy preserving computation, in the sense that all participating parties would follow the protocol without any deviation but they are free to use any intermediate result or data that pass through them during the execution of the security protocol [7, 8, 13]. The

task of the mediator here is only to help resolve the query made by the information seeker by analyzing the query, splitting the query as applicable for individual data sources, collecting data (often in encrypted form) from individual data sources and compiling the final result in one or more iterations (this also may be in encrypted form) from the intermediate result parts sent by the data sources. The assumption of utp for the mediator will imply minimum possible trust to be imposed upon the service provider by both the querrier and the data sources, which will in turn allow the querrier to seek information freely and the data sources to respond to those freely provided of course other privacy concerns discussed below are taken care of.

We shall restrict ourselves in the current work to the CA-IRaaS, wherein a set of N Data Sources (data owners – for simplicity we do not make any distinction between data source and data owner unless it is called for), $DS_1, \ldots, DS_N$ are registered with the *Service Provider* (*SP*). They give out their Schema (data definition as in a database) for the exposable part of their data to SP. We have assumed for simplicity that the data belonging to a data owner is in the form of a relational database [53]. The data sources can also inform in general about their various privacy concerns, such as whether they would like to remain anonymous to the querrier (referred as Customer here) or to other data sources during query execution, whether their data or schema is public, i.e. whether these can be shared with other data sources or with the Customer. But it is always possible for a data source to tighten or relax the privacy constraint for a specific query. Given a query Q, the SP will invite potential data sources for resolving the query. Note that there may be one or more subsets of N data sources who can respond to this query. The SP takes this decision based on various factors, including cost, time estimates and contracts with the data sources. And, if one or more data sources have some problems at that point of time or are not interested to participate in the query, SP tries for another subset and so on. Assume that finally $DS_1, \ldots, DS_n (n <= N)$ have agreed to participate in the query processing. Note, the query Q formulated by the customer can be in clear to the SP, if so then the query components Q_k will also be clear to $DS_k, k = 1, \ldots, n$. But if Q or part of Q is opaque, then Q_k can be either clear or opaque. The parts of the query that are not visible are referred as '*sensitive*'.

In the information retrieval system other than the complexity of integrating diverse heterogeneous data in multiple data sources querying environment privacy is a serious issue both for the information consumers and providers. The privacy concerns are unauthorized disclosure of various information: for the customer the query Q and result, and for the data sources their data or result parts. For the service provider the issue is of winning the trust of both the information seeker and the information providers. More he can be trusted easier will it be for everyone else. Here the customer is an unfamiliar entity for the data sources, possibly even for the SP himself. Similarly, DSs are unfamiliar to the customer, sometimes unknown even to fellow data sources. Further, the strategic or business interests of any individual data source could conflict with those of other data sources. But,

sometimes the issue of efficiency and cost may overshadow the issue of security. Depending on the complexity of a query it may be beneficial to share schema, data, result parts or even query parts. Thus to understand the complexity of the privacy problem in full, one has to find out all possible points and channels of leak.

We are interested in developing a privacy model by taking into considerations the privacy concerns of all the players, the customer (querrier), data sources and the service provider. We find that there are six dimensions of privacy which presumably cover all the aspects involved: *Identity* (*I*), *Schema* (*S*), *Data* (*D*), *Result* (*R*), *Query* (*Q*) and *Query distribution* (*Qd*). These privacy issues are described below:

a) *Identity Privacy* mainly involves the knowledge required to communicate with a party, e.g. the ip-address which is a personally identifiable information (PII) automatically captured by another computer when any communication is made over the Internet. This usually occurs before there is any opportunity to review a privacy policy. The amount of information available about users from their IP addresses varies depending on how they are connected to the Internet and other information that may be available. This is often referred as *online* identity. We would also like to include off-line identity (like name of the person or organization, address, and other details). This would be useful from the point of view of trust that one can associate with the party. Here, the issue would be whether one would like to remain anonymous or would like to avoid access from another party.

b) *Schema Privacy* refers to the protection of individual schema for the data sources. Since the SP has to prepare a query plan for query execution similar in line with distributed database system and thus break down the query into query parts for the data sources, knowing the schema by the SP is necessary. For the SP the schema would mean the global schema that SP constructs out of the schema of all the *N* data sources. This global schema is necessary for verifying the query given by the customer. If the query cannot be satisfied with the global schema then this query cannot be answered by this IRaaS. In that sense the customer should have either direct knowledge of the schema or some indirect knowledge such as application specific IR services to facilitate the query making process. The SP can also give a query interface to the customer to accept a query. Once the query has been accepted, one may consider a subschema which is adequate for the query as the schema of the SP.

c) *Data Privacy* refers to the protection of data belonging to a data source from other data sources, customer as well as the service provider. But for solving a query sometimes data might have to be shared with others, then appropriate transformation like encryption needs to be applied before sending it to another party, unless data sharing has been permitted by the data source. Further, note that data sharing would often need access to the respective schema. But if data is public for any data source access can be unrestricted. Still precaution has to be taken to prevent data from being manipulated.

Either way maintenance of appropriate data privacy would encourage more data sources participate in the IR service.

d) *Query Privacy* refers to the protection of the customer query from the SP and DSs. The customer is particularly interested to protect the sensitive parts of a query, e.g. I am searching for details of certain activities of a criminal within a given period of time, but I would not like to disclose the identity of the criminal or the period of interest either to DSs or to SP. This would mean protecting query parts from respective DSs.

e) *Result Privacy* refers to the protection of the result from the SP as well as the DSs. It is possible that the customer may not mind the SP knowing the result, but does not want DSs to know their parts of the result; or it could be just the converse. In some cases the customer may not also like to reveal the final query result as well as any intermediate computation to any third party.

f) *Query distribution Privacy* refers to the protection of knowledge of query distribution of SP from the DSs and the customer. Note, the SP has distributed the query to a set of DSs based on their availability and suitability. But protecting this information could be very crucial to the success of the query execution without violating any privacy. This protection works at two levels. At the first level there are DSs who are not involved in the query execution and hence may not have any idea about the ongoing query, particularly those whose data has no relevance for the query, either for the absence of relevant attributes or absence of relevant data (tuples or rows in a relational database). However the SP has to have the required knowledge regarding the range of data values of a DS in case it has to take a decision about the DS's participation. At the second level the DSs who are involved would not be informed about each other's participation. This information also may be kept from the customer.

Let us look at the individual privacy issues at greater details. First consider the Identity privacy. In one sense this is the most important of all privacies. Two reasons can be cited. First it concerns each and every party. Second, it is the gateway for making further accesses to a party. In our model problem we have N+2 parties – SP (service provider), C (customer or querrier) and $DS_1, \ldots, DS_N$ (N Data sources). So (N+2)(N+1) one way communications (or more specifically access rights) are possible among them which results in a maximum of $2^{(N+2)(N+1)}$ privacy types against identity privacy. So our objective is to find out feasible communications between the parties. Here, SP knows the identity of C and vice versa. Similarly, SP knows the identity of DSs and vice-versa. So what remains of interest are the possible communications between C and any $DS_k, k = 1, \ldots, N$ and between DS_j and DS_k, $j = 1 \ldots N$ and $k = 1 \ldots N$. Total number of such communications is $N(N+1)$, or to be more precise $n(n+1)$ where n data sources out of N participate in the processing. Assuming that all the DSs choose similar privacy parameters (symmetric case) the following table is constructed for identity privacy.

Privacy Type	Privacy Protection (Identity)		
	C from DS	DS from C	DS from other DS
0 (Public)	No	No	No
1	No	No	Yes
2	No	Yes	No
3	No	Yes	Yes
4	Yes	No	No
5	Yes	No	Yes
6	Yes	Yes	No
7 (Private)	Yes	Yes	Yes

Table 1: Identity Privacy of Customer and Data Sources (Symmetric Case)

The protection columns indicate whose identity is protected (hidden) from whom. Thus a protection has only two possible values - Yes or No. For the non-symmetric case where each data source decides independently the identity privacy issue is more elaborate. This is expressed as follows:

Privacy Protection (Identity)								
C from DS_1	. . .	C from DS_N	DS_1 from C	. . .	DS_N from C	DS_1 from DS_2	. . .	DS_N from DS_{N-1}
*	*	*	*	*	*	*		*

Table 2: Identity Privacy of Customer and Data Sources (Non-Symmetric Case)
[wildcard * indicates either "Yes" or "No"]

The table shows the existence of $(N+1)N$ privacy protections, which implies $2^{N(N+1)}$ privacy types. In other scenarios, for example, there may be some public data sources which do not mind incoming communications for any given query. Further note that the privacy statement may change from query to query.

Let us next consider Schema privacy. Here the main concerns are protecting individual schemas S_k of the data sources DS_k $k = 1 \ldots N$ being protected from other DSs as well from the customer. This is because the SP already has the knowledge of schemas. Thus we have the following possible privacies:

Privacy Type	Privacy Protection (Schema)	
	Schema Sk protected from	
	C	DSj
0 – 3	*	*

Table 3: Schema Privacy of Data Source DSk

There can be other issues like the disclosure of the global schema to the customer or even to the data sources. Though the knowledge of global schema may help the customer in expressing its query, its disclosure to the

data sources does not reveal much information to any individual DS regarding other DSs, unless the number of DSs is not too small.

Privacy Type	Privacy Protection (Schema)	
	Global Schema - SP Protected From	
	C	Any of the DSs
0 – 3	*	*

Table 4: Schema Privacy for SP (Symmetric for Data Sources)

Now let us look at Data privacy. Data are the valuable assets of the data sources. The data privacy of DS_k relates to the set of attributes A_k which are present in Q_k (the query part corresponding to DS_k). The concern is if any attribute from this set needs to be protected. The attributes may have to be protected from other data sources, the customer as well as the service provider. Here we have not distinguished between data and its metadata (other than schema) like index, range values, etc.

Privacy Type	Privacy Protection (Data)		
	Attribute Set Ak protected from		
	C	SP	DSj
0 – 7	*	*	*

Table 5: Data Attribute Privacy of DS_k

Result privacy for a data source refers to protecting its computed (intermediate) result part R_k from C, SP and other data sources. Note, it is possible that R_k is computed in several iterations taking possibly help from other DSs and SP, and even C (used as a conduit). For the SP it means protecting the entire result or part result computed by SP in the process from DSs, the latter might also have to be protected from C. Important thing to note here is that the result part R_k apparently does not disclose anything directly about data of DS_k or about the query. Similarly, the final result might not be of direct interest to a data source. But there can be hints about these from part results or the full result. The data sources may not want this partial leakage of their data. The customer may not like the leakage of either the part result or the full result. The interesting issue is that sometimes even the service provider would not like to share the part result with the customer as it might reveal some extra knowledge about the result, data or schema which may not be desired. Thus it seems this privacy requirement is more likely to change on query basis. Further note that SP is unlikely to distinguish between data sources for the protections. The following two tables describe the privacy issue.

Privacy Type	Privacy Protection (Result)		
	Result Part Rk Protected From		
	C	SP	DSj
0 – 7	*	*	*

Table 6: Result Privacy of Data Source DSk

Privacy Type	Privacy Protection (Result)	
	Final Result - SP Protected From	
	C (for intermediate result)	Any of the DSs (for final result)
0 – 3	*	*

Table 7: Result Privacy for SP (Symmetric for Data Sources)

Query privacy refers to the protection of the full query Q (originated from C and passed onto SP with possible encryption of sensitive parts) and $Q_k, k = 1, \ldots, n$ (n = number of DSs finally selected by SP for query solving) belonging to DS_k. Q_k may or may not have a sensitive part. C would like to protect Q from SP in the sense that it would not like the sensitive parts of Q are disclosed to SP through implication of any information passing through it. C also might like to protect Q_k's (sensitive parts) being protected from DS_k. Here the symmetry between the data sources is very much expected both from SP and C's points of view.

Privacy Type	Privacy Protection (Query)
	Qk Protected From DS_k
0	No
1	Yes

Table 8: Query Privacy of Data Source DSk

Privacy Protection (Query)		
Privacy Type	C Protected From	
	SP	Any of the DSs
0 – 3	*	*

Table 9: Query Privacy for SP (Symmetric for Data Sources)

Finally we come to what is known as Query distribution privacy. This is a totally different kind of privacy. The issue is while a query is being executed through a collaborative computation undertaken by a set of data sources, the SP and also possibly the customer a simple knowledge can greatly influence how the security is implemented as well as how the efficiency is going to be achieved. Usually, there will be a tension between these two factors, though security takes a priority in most situations. This knowledge is regarding the choice of the set of $DS_1, \ldots, DS_n$ made by the SP and mutually agreed by all (the customer usually wouldn't be involved in this process.). We are talking about the disclosure of identities of the chosen DSs – we call this *Query Distribution* knowledge. This disclosure can be made to the DSs and / or to C. The most restrictive one would be when it is not disclosed to either of them, we call that *Closed Query Distribution* (*Closed Qd*). This seems to be the most acceptable privacy as it makes preservation of privacy much simpler. The other options are named *Data Source Open Query Distribution* (*DS-Open Qd*), *Customer Open Query Distribution* (C-Open

Qd) and *Open Query Distribution* (*Open Qd*) depending on whether the knowledge is made open to the DSs (all of them – it doesn't make sense to distinguish one from another), C or both. Openness helps in query efficiency but makes it harder to ensure privacy. If data sources are public, the open schemes would be more useful. The privacies are put down in the following table.

Privacy Protection (Query Distribution)		Privacy Type
SP Protected From		
C	Any of the DSs	
Yes	Yes	Closed Qd
Yes	No	C-Open Qd
No	Yes	DS-Open Qd
No	No	Open Qd

Table 10: Query Distribution (Qd) Privacy for SP (Symmetric for Data Sources)

4 Interdependence of the Privacy Issues

It is well understood that the privacy modelling of a task as complex as querying heterogeneous data over multiple unknown data sources for allowing maximum possible privacy flexibility taking care of security concerns as well as efficiency concerns along with winning the trust of all parties involved is not a simple problem. This suggests that we need to standardize the process of privacy modelling to address all the issues concerned in an organized and objective manner. Towards this objective one needs to first identify the issues for privacy which are as far as possible mutually independent and exhaustive. The number of privacy issues to be selected depends on the granularity of privacy analysis. More granular is the privacy analysis more detailed will be the privacy statement and easier it will be to embed the privacy statement in the query execution protocol. For our problem we have identified six *privacy issues* (see above) which we consider adequate to take care of the problem. If we wanted to make a distinction between online identity and offline identity, and wanted to have control on the index of the data as well as the main data, we would have two more privacy issues. Then the next task would be to investigate these privacy issues acting upon interactions between any two parties involved in the process (this is called *privacy protections*). Here we have n+2 parties, n data sources, customer and the service provider. So the number of privacy protections is $(n+2)(n+1)$ as the protection is one directional and the maximum number of possible *privacy types* that IRaaS may need to support is $2^{m(n+2)(n+1)}$, the maximum number of degrees of freedom (dof) is $m.(n+2)(n+1)$ where m is the number of privacy issues at the root (top). For example, for our proposed CA-IRaaS symmetric model we have m = 6 and n = 10, so this number is $2^{6.12.11}=2^{792}$, and dof is 792. To make this manageable we have to look into *interdependencies* among these issues for any two

parties to interact. For this we have developed algebra based on *join* and *dominance* relation between privacy issues and protections. Finally we look into some feasible scenarios.

The basic idea behind this algebra is the simple fact that a privacy issue need not be completely independent of other issues. For example, if identity of a data source is protected from the customer then the customer cannot access the data source for its schema or data. Thus the identity protection automatically gives protections to other type of privacies. We envisage that identity privacy dominates schema privacy and data privacy for protection of DS from C. Again if we examine separately we find schema privacy dominates data privacy, query privacy is dominated by identity privacy and so on. This calls for a deep look at the privacy issues against protections and their dominance relations and join. The privacy algebra is built on this idea.

4.1 Privacy Algebra

We develop a simple algebra for constructing *composite* privacy issues and protections from elementary privacy issues and protections. This helps in developing a consolidated model for privacy for a complex multi-party computation.

Definitions:

A *privacy issue* (*entity*) is a specific privacy concern expressed and agreed by all the parties in a multi-party computation. It is represented as a matrix, each column represents a *privacy protection* and row represents a *privacy type*. Let *P* be a privacy type used in the following discussion.

A *privacy protection* refers to the protection of one party, say *a,* from another party, say *b*, i.e. *a* protected from *b*, or conversely, *a* open to *b* w.r.t. the underlying privacy issue and hence it has only two possible values "Yes" (*y*) or "No" (*n*). The set of privacy protections in *P* is denoted by *protection*(*P*).

A *privacy type* refers to a particular combination of protections available in a privacy issue. Sometimes privacy types are labelled for easy reference (e.g. Qd privacy – Table 10). The set of types in P is denoted by *type*(*P*).

P1 is a *type-subset* of *P* if *type*(*P1*) is a subset of *type*(*P*) [use subset notation]. Similarly, *P2* is a *protection-subset* of *P* if *protection*(*P2*) is a subset of *protection*(*P*).

Conditioned Privacy Issue P(*c*) is obtained by applying certain selection condition *c* onto the parent privacy issue *P*, or *P*(*Q*) by imposing another privacy issue *Q* upon it. Note, *P*(*c*) or *P*(*Q*) could be a type-subset, protection-subset or both of P.

A privacy issue having *m* privacy protections has a maximum of 2^m privacy types. A *non-trivial* privacy issue will have less than 2^m privacy types. A trivial issue would have 0 or all 2^m protections.

Examples: Refer to Table 1 displaying the Identity privacy for the Customer and Data Sources for IRaaS (the symmetric case). It has three protections, i) C protected from DS, ii) DS protected from C, and iii) DS protected from other DSs. This privacy has eight types. All or some of the types could be labelled for convenience, e.g. the first type has been called Open or Public – where each party is accessible to other, the last one Closed or Private – where none is accessible to another. Since we haven't put any condition on the issue, there are all 2^3=8 types. From Table 2 one can see that all possible communications are being allowed between any two parties – C and DSs. Thus there are $N(N+1)$ protections and $2^{N(N+1)}$ privacy types. But note that Identity privacy issue for C and DS (Symmetric Case) is both a type-subset and protection-subset of Identity privacy issue for C and DS (Non-symmetric Case). The condition 'symmetry among the DSs' applied on the latter will reduce it to the former, in other words, the former is a conditioned issue w.r.t. the latter.

Consider the scenario of Closed Query Distribution (Table 10), where the customer and the data sources are unknown to each other, the only privacy type allowed for Identity privacy is (Yes, Yes, Yes) – the Closed / Private type, whereas the Open Query Distribution allows only (No, No, No) – the Open / Public type. The following table depicts a restricted identity privacy which allows identity sharing only as guided by Query distribution policy and nothing else. Formally, Identity privacy has been conditioned by Query distribution policy.

Privacy Type	Privacy Protection (Identity)			Corresponding Qd Type
	C from DS	DS from C	DS from other DS	
0 (public)	No	No	No	Open Qd
1	No	No	Yes	C-Open Qd
6	Yes	Yes	No	DS-Open Qd
7 (private)	Yes	Yes	Yes	Closed Qd

Table 11: Identity Privacy conditioned by Query distribution Privacy

Let us consider three privacy issues *X*, *Y* and *Z* for the following discussions.

Definitions:

*Join of privacy issues***:** Let *X* have *p* privacy protections and *Y* have *q*, of which *r* are common. Then *X.Y* represents a new privacy issue obtained by joining *X* and *Y*, the join is performed in the same way database relations are joined. Hence, the privacy issue *X.Y* will have *p*+*q*-*r* protections. If *X* and *Y* are two independent privacy issues, then *X.Y* will have *p*+*q* protections, as they do not have any common protection. Note, $A.A = A$, $A.B = B.A$ and

(*A.B*).*C* = *A*.(*B.C*) . The join operation is thus *idempotency preserving*, *commutative* and *associative*.

*Dominance between Privacy Protections***:** Let *x* and *y* be two privacy protections belonging to a privacy issue P. We say that the protection *x* dominates protection *y* over the privacy issue *P* (denoted, *x* > *y over P*) iff when *y* is not protected (i.e. *y* is open, or privacy protection of *y* = "No") then *x* cannot be protected (i.e. *x* must be open, or privacy protection of *x* = "Yes" can not hold). In other words protection pair (*x*, *y*) cannot assume (Yes, No).

*Dominance between Privacy Issues***:** Let *x* and *y* be two protections belonging to the privacy issues *X* and *Y* respectively. We say that *x* > *y* iff *x* > *y* holds in the privacy issue *X.Y*. We refer to this as dominance between two privacy issues, i.e. *X* dominates *Y*, denoted *X* > *Y* over protections x and y respectively. Moreover, if *x* and *y* refer to the same protection, then we say *X* > *Y* over *x*, or *X* > *Y* w.r.t. *x*. By applying dominance relations between privacy issues or privacy protections one essentially conditions the joint privacy issue or protections, i.e. obtains conditioned privacy issues.

Transitivity of Dominance Relations: The protection dominance is a transitive relation, i.e. *x* > *y* and *y* > *z* then *x* > *z*, where *x*, *y* and *z* protections may or may not belong to the same privacy issue. Similarly, privacy issue dominance also is transitive, i.e. *X* > *Y* and *Y* > *Z* then *X* > *Z*, where *X*, *Y* and *Z* are privacy issues over a common privacy protection or over different protections as discussed above.

Let us represent the privacy type constants "Yes" by 1 and "No" by 0. Let a and b be two privacy issues, such that a > b, then the set of valid privacy types for (a, b) is {(1, 1), (0, 1), (0, 0)}, i.e. (1, 0) is not a valid privacy type. In other words for a = "Yes" only value that is allowed for b is "Yes". We can also represent the privacy types by binary strings, in this case of length 2. Then the valid types are represented by three binary strings, {11, 01, 00} = {11, 0*} = {1^2, 0*}. (Here, the superscript indicates repetition of the bit and * is a wildcard (either 0 or 1). Therefore, $0 *^2 1^3$ represent four strings, namely, 000111, 001111, 010111 and 011111.

Let $b_1, \dots, b_k$ k>= 2, be k privacy issues w.r.t. to a common privacy protection. Alternatively, let $b_1, b_2, \dots, b_k$ be the k privacy protections over the domain of one or more privacy issues (which have been joined). For the joint protection domain $(b_1, b_2, \dots, b_k)$ the total number of possible privacy types is 2^k. As discussed above, we can represent these types by binary strings of length k, i.e. k-bit strings. Treating each bit string a number, the privacy types ranges from 0 to $2^k - 1$.

Claim: Let $b_1 > \cdots > b_k$ hold for the joint protection domain $(b_1, \dots, b_k)$. Then, there are only $k + 1$ valid privacy types $\{1^k, 0^1 1^{k-1}, \dots, 0^{k-1} 1^1, 0^k\}$, equivalently $\{2^k - 1, 2^{k-1} - 1, \dots, 2^0 - 1\}$ out of a total possible 2^k types.

Proof: We prove this by induction. Note the claim holds for k=2. The valid types are {11, 01, 00}. Let the claim hold for k = j. To show that it holds for

k = j+1. The case k = j indicates that $b_1 > \cdots > b_j$. Assume $b_j > b_{j+1}$. So, in the combined privacy $(b_1, \ldots, b_j, b_{j+1})$ obtained by joining $(b_1, \ldots, b_j)$ and (b_j, b_{j+1}), we have $b_1 > \cdots > b_j > b_{j+1}$. The result can be verified from the following table:

Privacy: $(b_1 > \cdots > b_j)$		Privacy: $(b_j > b_{j+1})$		Privacy: $(b_1 > \cdots > b_{j+1})$	
Numeric Value	j-bit Binary String	2-bit Binary String	Numeric Value	(j+1)-bit Binary String	Numeric Value
$2^j - 1$	111…11	11	2^2-1	111…11	$2^{j+1} - 1$
$2^{j-1} - 1$	011…11			011…11	$2^j - 1$
…	…	…	…	…	…
$2^1 - 1$	000…01			000…11	$2^2 - 1$
$2^0 - 1$	000…00	01	2^1-1	000…01	$2^1 - 1$
		00	2^0-1	000…00	$2^0 - 1$

Clearly, the combined privacy has j+2 privacy types $\{2^{j+1} - 1, 2^j - 1, \ldots, 2^1 - 1, 2^0 - 1\}$. This completes the proof.

4.2 Privacy Algebra applied to IRaaS

In this section we demonstrate how privacy algebra can be used to simplify and consolidate the privacy issues for IRaaS. First we notice that successive applications of join of (elementary) privacy issues are not affected by the sequence in which the operands are selected for the join operations. For example, A.(B.C) = A.(C.B) = (A.C).B = (C.A).B = C.(A.B) = C.(B.A) = (C.B).A = (B.C).A = B.(C.A) = B.(A.C) = (B.A).C = (A.B).C. We are thus left with detecting interdependencies in the form of dominance relations between privacy issues over privacy protections or vice versa.

To have a peek at the interdependence issue in action it will be useful to look at a situation where a lot of flexibilities are available. Let us therefore focus on the most complex case of privacy in IRaaS (complexity is by the number of feasible privacy types) where we allow all possible communications between any two parties out of *n*+2 parties, C, SP, $DS_1, \ldots, DS_n$, i.e. no symmetry among DSs are assumed. We use *k* and *j* (*j* ≠*k*) for indexing DSs. We consider all the five privacy issues: *I* (identity), *S* (schema), *D* (data), *Q* (query) and *R* (result). The query distribution (Qd) issue is fixed at (No, No), i.e. *Open-Qd* issue. So there are a total of $2^{5(n+2)(n+1)}$ possible privacy types (There are (n+2)(n+1) one way accesses for each of 5 privacy issues.). Let us consider the communication between DS_k and DS_j, i.e. the privacy protection "DS_k protected from DS_j" across different privacy issues. Note, for this protection, the following dominance holds: $I > S > D$, $I > R$ and $I > S > Q$. This is because without access to identity one cannot have access to schema. Similarly without learning the respective schema learning data would not be possible, but for learning the result part the knowledge of schema need not be essential.

However, the query part would require the knowledge of the respective schema.

Coming to the privacy types, for I > S we have 3 valid privacy types as seen in Table 12a, a crisp form in Table 12b. Similarly for S > D we have 3 valid privacy types as seen in Table 12c, by joining I > S with S > D we have 4 valid privacy types for I > S > D as seen in Table 12d.

I	S	Type
No	No	00
No	Yes	01
Yes	Yes	11

Table 12a: I > S (#type = 3)

I	S	Type
No	No	00
*	Yes	*1

Table 12b:Crisp form of Table 12a

S	D	Type
No	No	00
*	Yes	11

Table 12c: S > D (#type = 3)

I	S	D	Type
No	No	No	000
No	No	Yes	001
*	Yes	Yes	*11

Table 12d: I > S > D (#type = 4)

For joining two tables T_A and T_B, each row of T_A has to be crossed with each row of T_B. However, only matching types will remain. Note, "*" matches anything, but outputs the symbol which matched it, i.e. "*" matching with "Yes","No", "*" will produce "Yes" ,"No", "*". Table 12d further simplifies to:

I	S	D	Type
No	No	*	00*
*	Yes	Yes	*11

Table 12e: Simplified version of Table 12d (#type = 4)

By applying the *Claim* about the sequence of dominant relations, we get the same types: $\{2^3-1, 2^2-1, 2^1-1, 2^0-1\}_{value} = \{111, 011, 001, 000\}_{bit-string} = \{*11, 00*\}$. Similarly, for I > S > Q, we have:

I	S	Q	Type
No	No	*	00*
*	Yes	Yes	*11

Table 12f: I > S > Q (#type = 4)

By joining Table 12e with Table 12f one gets for (I, S, Q, D),

I	S	D	Q	Type
No	No	*	*	00**
*	Yes	Yes	Yes	*111

Table 12g: Join of (I>S>D) with (I>S>Q) (#type = 6)

Now for I > R we have:

I	R	Type
No	No	00
*	Yes	*1

Table 12h: I > R (#type = 3)

Thus for the protection "DS_k from DS_j" considering the dominance relations among I, S, Q, D and R privacies, we have the following valid privacy types:

I	S	D	Q	R	Type
No	No	*	*	*	00***
No	Yes	Yes	Yes	No	01110
*	Yes	Yes	Yes	Yes	*1111

Table 12i: Joining of Table 12g with Table 12h (#type = 11)

Thus by applying the dominance relations alone we have been able to reduce the number of privacy options between two data sources from 2^5 = 64 to 11. Similarly, considering the protection DS_k from C, we have relations I > S > D and I > R.

I	S	D	R	Type
No	No	*	*	00**
No	Yes	Yes	*	011*
Yes	Yes	Yes	Yes	1111

Table 13: I > S > D and I > R (#type = 7)

Protection C from DS_k involves only identity privacy, and hence has two types: "*". Protection DS_k from SP involves data and result privacies which are independent of each other, hence has all four types (*,*) valid. DS's or C can not expect any additional information from SP, hence no further privacies are necessary. In any case C and DS_k both can communicate with SP, and vice versa.

The above privacies are independent. Hence, the total number of feasible privacy type is $11^{n(n-1)}.7^n 2^n.2^n$.This is considering all possible independent decisions by each data source. The situation becomes much simpler if we consider the symmetric case where each data source follows an identical policy towards the customer and SP, and the customer and SP also follow identical policy for all the data sources. Number of feasible types would reduce to 11.7.2.2=308. For any practical problem many other conditions will be imposed by the participants to make the situation more manageable.

We can conclude this analysis by observing that even if the privacy issue of a complex multi-party computation appears to be no less complex, this analysis can be simplified a lot by identifying privacy issues, privacy protections, dominance relations interlinking these privacy issues and protections and finally applying privacy algebra to come out with consolidated privacy statement. Of course this algebra needs to be enriched to a canonicalization process. The understanding of privacy modelling

problem can be strengthened further by applying this analysis process to different problems such as on line auctions, combinatorial or reverse auctions and on line shopping. This in turn will improve these services as well. As next generation systems will be highly collaborative and will have to share information, interoperability via open communication and standardized data exchange is needed [43]. Such system will need planned privacy model. One such example is collaborative cloud based industrial automation [43].

5 Secure IR Framework

The main task of the information retrieval mediation is to coordinate the communication and distribution of information consumer's query among the mediator, the information consumer and the data sources [6]. Mediator is a software component at middleware layer with the services for information retrieval. The proposed framework of CA-IRaaS is '*central mediator/wrapper*' architecture [9] along with the security mechanism built at the information consumer's end and at the data sources' side. The mediator which sits in between the customer and data sources is basically positioned in SP who provides the necessary interface to the customer for querying. The central mediator contains a universal mediator schema that presents a view of the integrated data to the customers through the application. The mediator architecture is depicted Figure 4.

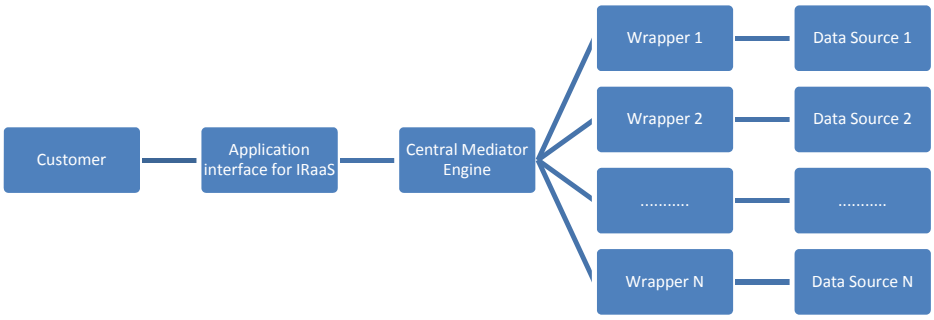

Fig 4: Central Mediator Architecture

The application interface and the central mediator engine are hosted in the Cloud. The mediator engine is interfaced to a number of data sources through wrappers. The central mediator contains a global schema made out of the individual schema of the data sources. Through its application interface IRaaS presents a transparent view of the integrated data to the customer [9]. For each data source there is a wrapper. The wrappers contain code to map the global schema to local schema applicable to individual data

source. Customer's query passes through query optimization before mapping by the central mediator and generates query components for each data source. Now, the privacy statement *PS* arrived at through joint negotiation of all the parties involved has to be embedded properly in the algorithm (without privacy considerations). Each action gets modified accordingly.

The system architecture of IRaaS *without privacy mechanism* is summarized in the following steps:

1. Customer sends a query using the IRaaS application interface to the Service Provider
2. The Service Provider accepts the query, determines the set of appropriate data sources to answer the query and hands over the query to the mediator engine
3. Using the global schema the mediator optimizes the query and generates sub query (query components) for individual data sources
4. For data source its wrapper translates the sub query into a query expression that it is executable locally n
5. Each data source executes the sub query and sends the result to the mediator engine through the wrapper
6. At the mediator engine the final result is obtained after joining, selecting or merging as appropriate (if required iterating the process by going back to Step 5) and passes on to the Service Provider
7. The Service Provider returns the answer to the Customer

The system architecture of IRaaS *with privacy mechanism* is summarized in the following steps:

Pre-processing step: (this step is query-independent and performed initially by the data sources). Each data source submits its preferences for the privacy issues and protections to the Service Provider. [The privacy preferences are accepted for only those cases which allow user choice (*)]

Query processing steps:

1. Customer sends a query along with her preferences for the privacy issues and protections using the IRaaS application interface to the Service Provider. The sensitive components (constants for example) of the query are kept hidden in this communication depending on the privacy requirement of the Customer from the Service Provider and the unknown Data Sources. However the query text remains legible to carry on with the processing.
2. The Service Provider accepts the query, determines the set of appropriate data sources to answer the query. At this stage the Service Provider matches the privacy preferences of the selected data sources related to customer and if required may negotiate with either party to finalize the set. Once determined the Service Provide hands over the query to the mediator engine and other information like set of data sources etc. as appropriate. The knowledge of the set of appropriate data sources who

answer the query (query distribution) is passed on either to the Customer and/or the Data Sources as per the privacy setting.

3. Using the global schema the mediator optimizes the query and generates sub query (query components) for individual data sources.
4. For each data source its wrapper translates the sub query into a query expression that it is executable locally. Depending on the privacy requirement of the query the data source either gets the hidden components directly from the customer or through the Service Provider.
5. After obtaining the sensitive query components each data source executes the sub query and sends the result to the mediator engine through the wrapper. Depending on the specific privacy choice of the customer the query execution may have to be executed differently like PIR where the sensitive components are not even seen at the data source level [15, 27], the query is executed at the data source with its consent but without it knowing what is being executed. (Steps 4 and 5 may have to be iterated more than once depending on the complexity of the query.)
6. At the mediator engine the final result is obtained after joining, selecting or merging as appropriate and passes on to the Service Provider. This step involves number of computations and communications among the parties which again depend on the privacy settings and the complexity of the query.
7. The Service Provider returns the answer to the Customer. Depending on the privacy need result may have to be hidden from the Service Provider but the Customer should be able to unhide it.

The schematic diagram of the query processing framework is seen in the figure below:

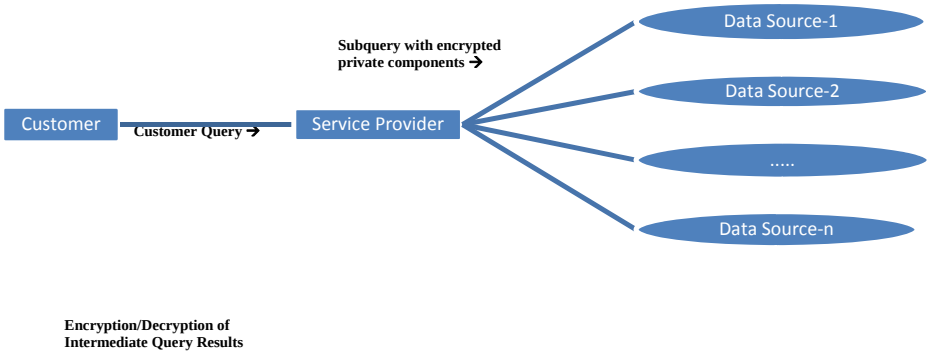

Fig 5. Secure IR Framework

6 Conclusion and Further Scopes of Work

Privacy analysis has not always got proper attention in the literature often overridden by security algorithms. This work attempts to fill in this gap. The

strength of this work we believe is in the privacy analysis conducted in depth for a problem as complex as IRaaS following an objective method developed in the work itself. We have discussed here the problem of information retrieval services offered by a service provider to its customers and proposed an ubiquitous IR service like IRaaS. We have proposed two basic models – open access and closed access. While the closed access information retrieval service has a pre-determined domain based query infrastructure through data integration from heterogeneous data sources, open access poses the problem of any arbitrary query being made to the service provider. We have suggested collaborative IR services in the form of collaborative IRaaS for both the closed and open models. There is a huge scope of work in this direction. Enterprise based IRaaS is another area to look into deeply. The service provider plays the role of a mediator in the information retrieval service. We have performed a detailed analysis of privacy and a secure framework sketch for the closed access service which can be used for privacy preserving information retrieval system. We have proposed a privacy algebra which has been demonstrated on IRaaS to show that the former greatly simplifies the process of privacy modelling. We are in the process of strengthening this privacy algebra which we believe will be helpful for canonicalization of the complex task of privacy modelling for any reasonably complex multi-party computation task. One important issue remains unresolved how to determine the granularity of the privacy issues, as the number of privacy types explodes exponentially. But keeping this number too small may create a lot of semantic distance between the users' thinking and the implementational feasibility. A hierarchical approach may be beneficial in this respect. Another thing this work is completely silent is regarding query complexity. Our experience shows that handling even apparently simple queries in a privacy preserving manner is not at all a trivial task. Similarly, the impact of heterogeneity of data on privacy is yet to be studied. Another problem is how to support different privacy requirements (usually this variety will be quite huge) within an IR service, and this is the crux of privacy modelling requirement. This we believe will be a real challenge for some time to come. Further scopes of work would be in the area of working out costs and revenue sharing and pricing mechanism for providing different types of IR services and attracting more clients both data sources and customers, developing a robust mediation framework maintaining the provision of privacy, security and trust and also developing a user interface model for inputting query in arbitrary areas.

References

[1] A.K. Pal, S. Bose, "Information Retrieval as a Service for Multiple Heterogeneous Data-Privacy Model", in B.H.V. Topping, P. Iványi, (Editors), "Proceedings of the Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering",

Civil-Comp Press, Stirlingshire, UK, Paper 31, 2013. doi:10.4203/ccp.101.31

[2] R. Lawrence, "How to Query Multiple Databases and Generate Reports", http://www.unityjdbc.com/doc/multiple/multiplequery.php

[3] A.P. Sheth, "Semantic Issues in Multidatabase Systems - Preface by the Special Issue Editor." SIGMOD record 20.4 (1991): 5-9.

[4] L. Liu, C. Pu, Y. Lee, "An adaptive approach to query mediation across heterogeneous information sources." *Cooperative Information Systems, 1996. Proceedings., First IFCIS International Conference on*. IEEE, 1996.

[5] L. Lakshmanan, F. Sadri, I.N. Subramanian, "SchemaSQL-a language for interoperability in relational multi-database systems." *VLDB*. Vol. 96. 1996.

[6] L. Ling, C. Pu, "An adaptive object-oriented approach to integration and access of heterogeneous information sources." *Distributed and Parallel Databases* 5.2 (1997): 167-205.

[7] R. Agrawal, A. Evfimievski, R. Srikant, "Information sharing across private databases." *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 2003.

[8] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, Y. Xu, "Two can keep a secret: A distributed architecture for secure database services." *CIDR 2005* (2005).

[9] Risch, Tore. "Mediators for Querying Heterogeneous Data." (2004).

[10] S.S.M. Chow, J.H. Lee, L. Subramanian, "Two-Party Computation Model for Privacy-Preserving Queries over Distributed Databases." *NDSS*. 2009.

[11] F. Emekci, D. Agrawal, A.E. Abbadi, A. Gulbeden, "Privacy preserving query processing using third parties." *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*. IEEE, 2006.

[12] B. Hore, S. Mehrotra, H. Hacigumus, "Managing and querying encrypted data." *Handbook of Database Security*. Springer US, 2008. 163-190.

[13] H. Hu, J. Xu, C. Ren, B. Choi, "Processing private queries over untrusted data cloud through privacy homomorphism." *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE, 2011.

[14] F. Olumofin, I. Goldberg, "Privacy-preserving queries over relational databases." *Privacy enhancing technologies*. Springer Berlin Heidelberg, 2010.

[15] J. Reardon, J. Pound, I. Goldberg, "Relational-complete private information retrieval." *University of Waterloo, Tech. Rep. CACR* 34 (2007): 2007.

[16] S. Hildenbrand, D. Kossmann, T. Sanamrad, C. Binnig, F. Faerber, J. Woehler, "Query Processing on Encrypted Data in the Cloud" by ETH, Department of Computer Science, 2011.
[17] R. Kolavenu, R. Arasanal, "A Survey on Enterprise Databases in Cloud Computing", 2012, https://wiki.engr.illinois.edu/download/attachments/200481897/A+Survey+on+Enterprise+Database+Systems+on+Clouds.pdf
[18] M. Hogan, "How databases can meet the demands of cloud computing."*Sun Cloud Computing* (2008).
[19] S. Das, S. Nishimura, D. Agrawal, A.El. Abbadi, "*Live database migration for elasticity in a multitenant database for cloud platforms*. Technical Report 2010-09, CS, UCSB, 2010.
[20] C. Curino, E. Jones, Y. Zhang, E. Wu, S. Madden, "Relational cloud: The case for a database service." *New England Database Summit* (2010).
[21] I.F. Cruz, H. Xiao. "The role of ontologies in data integration."*Engineering intelligent systems for electrical engineering and communications*13.4 (2005): 245.
[22] R. Ahmed, P. De Smedt, W. Du, W. Kent, M. Ketabchi, W. Litwin, A. Rafii, M-Chien Shan, "The Pegasus heterogeneous multidatabase system."*Computer* 24.12 (1991): 19-27
[23] W. S. Ng, B. C. Ooi, K. L. Tan, A. Zhou, "PeerDB: A P2P-based system for distributed data sharing." *Data Engineering, 2003. Proceedings. 19th International Conference on*. IEEE, 2003.
[24] A. Y. Levy, A. Rajaraman, J. J. Ordille, "Querying heterogeneous information sources using source descriptions." (1996).
[25] A. Sheth, J. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases." *ACM Computing Surveys (CSUR)* 22.3 (1990): 183-236.
[26] C. M. Rood, D. Van Gucht, F. I. Wyss, "MD-SQL: A language for meta-data queries over relational databases." *Indiana Univ. CS Dept. TR528*(1999).
[27] B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, "Private information retrieval." *Journal of the ACM (JACM)*45.6 (1998): 965-981.
[28] L. Sweeney, "k-anonymity: A model for protecting privacy." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002): 557-570.
[29] R. Agrawal, R. Srikant, "Privacy-preserving data mining." *ACM Sigmod Record* 29.2 (2000): 439-450.
[30] G. Carraro, F. Chong, "Software as a service (SaaS): An enterprise perspective." *MSDN Solution Architecture Center* (2006).
[31] H. Katzan, "Cloud software service: concepts, technology, economics". *Service Science* 1.4 (2009): 256-269.
[32] D. Chen, H. Zhao, "Data Security and Privacy Protection Issues in Cloud Computing", *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*. Vol. 1. IEEE, 2012.

[33] E. J. Schweitzer. "Reconciliation of the cloud computing model with US federal electronic health record regulations", *Journal of the American Medical Informatics Association* 19.2 (2012): 161-165.

[34] L. A. Martucci, A. Zuccato, B. Smeets, S. M. Habib, T. Johansson, N. Shahmehri "Privacy, security and trust in cloud computing: The perspective of the telecommunication industry." *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on*. IEEE, 2012.

[35] A. Acquisti, "Privacy and security of personal information."*Economics of Information Security*. Springer US, 2004. 179-186.

[36] A. Acquisti, "Protecting privacy with economics: Economic incentives for preventive technologies in ubiquitous computing environments." *Proceedings of Workshop on Socially-informed Design of Privacy enhancing Solutions, 4th International Conference on Ubiquitous Computing (UBICOMP 02)*. 2002.

[37] A. Acquisti, H. R. Varian, "Conditioning prices on purchase history." *Marketing Science* 24.3 (2005): 367-381.

[38] C. Wang, Q. Wang, K. Ren, W. Lou, "Privacy-preserving public auditing for secure cloud storage." (2013): 1-1.

[39] De Capitani di Vimercati, S. Foresti, P. Samarati, (2012). "Managing and accessing data in the cloud: Privacy risks and approaches."*Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on*. IEEE, 2012.

[40] W. Shiyuan, D. Agrawal, A. E. Abbadi, "Towards practical private processing of database queries over public data with homomorphic encryption.", *Technical Report* 2011-06, *Department of Computer Science, University of California at Santa Barbara*, 2011

[41] S. Marston, Z. Li, S. Bandyopadhyay, A. Ghalsasi, "Cloud computing -The business perspective." *Decision Support Systems* 51.1 (2011): 176-189.

[42] C. Yoon, M. M. Hassan, H. Lee, W. Ryu, E.N. Huh, "Dynamic collaborative cloud service platform: opportunities and challenges." *ETRI journal* 32.4 (2010): 634-637.

[43] S. Karnouskos, A. W. Colombo, T. Bangemann, K. Manninen, R. Camp, M. Tilly, P. Stluka, F. Jammes, J. Delsing , J. Eliasson. "A SOA-based architecture for empowering future collaborative cloud-based industrial automation." *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2012.

[44] D. Vimercati, S. Foresti, S. Jajodia, "Controlled information sharing in collaborative distributed query processing." *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*. IEEE, 2008.

[45] M. Benedikt, P. Bourhis, C. Ley, "Querying schemas with access restrictions." *Proceedings of the VLDB Endowment* 5.7 (2012): 634-645.

[46] P. Yu, J, Sendor, G Serme, A. S. de Oliveira, "Automating privacy enforcement in cloud platforms." *Data Privacy Management and*

Autonomous Spontaneous Security. Springer Berlin Heidelberg, 2013. 160-173.

[47] X. Li, S. Goryczka, V. Sunderam, "Adaptive, secure, and scalable distributed data outsourcing: a vision paper." *Proceedings of the 2011 workshop on Dynamic distributed data-intensive applications, programming abstractions, and systems*. ACM, 2011.

[48] W. Shiyuan, D. Agrawal, A. E. Abbadi, "Is homomorphic encryption the holy grail for database queries on encrypted data", *Technical report, Department of Computer Science*, UCSB, 2012.

[49] C. Gentry, "A fully homomorphic encryption scheme.", *Diss. Stanford University*, 2009.

[50] G. Yubin, Z. Liankuan, L. Fengrena, L. Ximing. "A Solution for Privacy-Preserving Data Manipulation and Query on NoSQL Database." *Journal of Computers* 8.6 (2013): 1427-1432.

[51] T. Aditya, S. Chakravarthy, Y. Huang. "Information Integration Across Heterogeneous Sources: Where Do We Stand and How to Proceed?." *COMAD*. 2008.

[52] M. Doerr, J. Hunter, C. Lagoze, "Towards a core ontology for information integration." *Journal of Digital information* 4.1 (2006).

[53] G.Goetz, U. M. Fayyad, S. Chaudhuri. "On the Efficient Gathering of Sufficient Statistics for Classification from Large SQL Databases."*KDD*. 1998.