Distributed Accounting in Scope of Privacy Preserving

Marcus Hilbrich and René Jäkel

Center for Information Services and High Performance Computing (ZIH),
Technische Universität Dresden
marcus.hilbrich@tu-dresden.de

Abstract. Accounting is an essential part of distributed computing infrastructures, regardless whether these are more service-driven like Clouds or more computing oriented like traditional Grid Computing environments. Those infrastructures have evolved over more than the last decade and additional. beside the further development towards service-oriented architectures, the business aspect of especially Cloud Computing solutions becomes more and more relevant. In this paper we focus on user-centric aspects like privacy preserving methods to hide the users behaviour and to collect only necessary information for billing, under the assumption that an accounting system has to be integrated in the computing infrastructure and that a central interface is still desirable for billing and financial clearing.

1 Introduction

Nowadays, it is a quite common to pay just for metered services (pay-per-use) which are consumed for a specific task but having potentially a large resource share on hand. The payment is usually done by spending money, but could also realized by giving services in return or the promise that the work is relevant for a scientific community or a wider society. Meanwhile, different payment options in terms of pricing models have been developed, from simple flat rates (pay once and take what you need) to pay per request depending on the answer of the request and the number of requests.

As long as services of a single resource provider are used the payment has to be done straightforward. The provider logs what users or customers are using and tells them what they have to pay. If services have to be combined from different providers, or even service brokers, the accounting and billing issues are getting more complicated. It is obvious, that in such cases the billing of individual operators in a long service chain is not very comfortable, since in the common case direct contract between resource operators and users is needed.

Usually in an accounting and billing service in a distributed environment such as Grid, all accounting data are collected and transferred from all service providers to a central place. Based on this data pile bills are written and statistic information are created. The danger in this approach is that the operator of the accounting and billing service has a lot of sensitive information by hand, and it has to be guaranteed that privacy preserving issues for the users hold (usually done by contracts ore agreements).

J.J. Park et al. (Eds.): GPC 2013, LNCS 7861, pp. 11–22, 2013.

Our approach of an accounting concept is based on the prevention of such a central component to store all accounting information in a centralised manner. The accounting data are kept in the domain of the service operators and accumulate only coarsely granular data. In this way we realise billing and accounting without a need on transparent users.

2 Related Architectures

Accounting and billing is already done by various systems. The field of research we are working on are mainly covered by Grids and Clouds. Grid services are usually offered to a Virtual Organisation (VO) which allows its users to access services in its domain. Often the VO cares also of scheduling, billing and user support. The accounting systems tend to use centralised accounting databases, such as LUTS [1] or DGAS [2]. Accounting data are read e.g. from the batch system and are transferred to a central database. Afterwards the rights to access parts of the data are assigned to the users of the accounting system. SGAS [3] stores accounting data on VO servers to improve scalability. Common for all these systems is that accounting data are moved from the resource provider, which are the creator of the data, to a central component.

A quite new and emerging field of interest are federated Clouds, for which more advanced accounting concepts are needed in terms of privacy preserving. This kind of cooperating Cloud is not yet a way of Cloud usage, beside direct services or infrastructure utilization. In most cases, there are isolated Cloud provider [4], which can led to the widely discussed vendor-lock problem [5]. The manifold drawbacks (e.g. proprietary data formats which hinder exporting data and unexpected price changes or closing down of essential services) are already known [6] and different concepts were developed to overcome this limitations.

One initiative towards an Open Cloud is developed by the *Open Cloud Manifesto*'[1], which is a loose group of companies and projects to communicate demands and solutions for an Open Cloud.

In recent years more and more Open Source Cloud middlewares evolved, e.g OpenNebula [7] or Eucalyptus [8], to name only some prominent projects. Those enterprise solutions also support interfaces to established commercial cloud service providers, such as the EC2 interface introduced by Amazon. On the other hand they also follow the recent Open Cloud Computing Interface (OCCI)[2], which represents a RESTful protocol and development API. The development of this interface is driven by community users and has some history in distributed computing and particular in grid computing. By introducing a flexible interface the interoperability between different cloud providers can be increased. Therefore, the migration of applications or services from one provider to a different one becomes relatively easy, which is a huge step avoiding the vendor locked-in problem on the way towards a common Cloud Computing standard.

[1] Details via the Open Cloud Manifesto: `http://www.opencloudmanifesto.org`

[2] Listed projects and details can be found on their website: `http://occi-wg.org/`

A Hybrid Cloud [9] combines different cloud resources (in most cases a local or private Cloud and a public Cloud). This allows to schedule the users requests (e.g. to run a job or to access a service) based on the job description and a set of rules (constraints where to run the job, available budget for external resources etc.) on one of the Cloud resources. The decision which cloud is used can be delegated to a Cloud broker [10]. In this case a user (e.g. a company or a scientific community) asks a so called broker, which is the best Cloud provider to run a specific job at a given time. Therefore, the broker gets the actual service description of different Cloud providers and ranks them according to the needs of the users. The user has to sign at least two contracts. The first with the broker and the second with the Cloud provider. If more then one Cloud provider is needed to complete a task (e.g. one for storing data and one for computing) additional contracts have to be closed. Additional conditions to drive such an architecture is to use compatible APIs to access the different Cloud providers and to offer similar services. This can be achieved by standardisation of services and their description. A wildly accepted framework to compare services is not yet established but there is already research (e.g. [11]) how service comparison can be realised. Also a standardisation of describing Cloud services and their performance has to be found and an automatic way of closing contracts has to be introduced. The service description could be given by Service Level Agreements (SLAs) which are automatically signed for using a service as described by [12].

A federated Cloud creates a market for resources and potentially deals accounting and billing issues. This means every user and resource provider has a contract with the federation instance for providing or utilising resources, but it is not necessarily needed that the user has a contract with a resource provider. This allows supplier which use services or resources of other providers to offer more complex products or to provide services independent of resources and to select different resources for a service depending on the kind of data (related to real persons, anonymized data or data publicly available) to process [13]. In such a system the billing and financial clearing has to be done by the federation and accounting data has to be recorded on the resource provider. The general demands on such an accounting system are presented by [14] and [15] the specifics of federated Clouds are covered by [16].

The specific concept of federated Clouds with a widely accepted use case are so called Government Clouds. A Government Cloud is a Cloud-based systems to handle the computing and storage needs of administrative agencies. The resources could be public Clouds, private Clouds driven by the government or a Cloud provider, or local data centres of agencies, which form a federation to share there resources with other agencies. The advantage of such a concept compared with the direct use off local resources at each agency is that local peak demands can be resolved by using resources of other agencies. This allows to reduce the overall amount of resources, which are needed to process the given governmental tasks. One of the challenges for a Government Cloud is to respect several juristic limitations. This limitations depend on the data which has to be processed, and therefore the according service requests have to be categorised, e.g. if specific data needed for a service execution are not allowed to be transferred to a different site. Such restrictions can be constraints on the security level, e.g. this is a common requirement of legislation in federal states like Germany or the European Union. This shall ensure that data handling stays in the same juristic domain and that the data douse not leave the domain of governmental controlling authorities. For instance the Japan Kasumigaseki Cloud[3] has to deal with this juristic limitation. Some data have to be processed at the district the agency is located. This demands that a computing centre has to be located at each district. To realise a compensation between the agencies an accounting system has to be established. The concept of our accounting system could be deployed to such an infrastructure. Similar to the Kasumigaseki Cloud a computing infrastructure could be deployed for India [17]. For both Clouds our accounting approach can be considered.

3 Data Minimisation and Privacy Preserving

Data minimisation and privacy preserving for users is not a major topic of common accounting systems on a technical level. Data minimisation is a concept to protect privacy of users by reducing data to a minimal level, which is essentially needed to realise the accounting service. This can be realised by deleting data, which is not longer needed or by storing data only in a non-personal way. This can be illustrate by the following examples:

- Someone prefers to by products of a special brand from an online seller, which could result in a handicap, if the particular person tries to apply for a job in a company of an competing brand.
- The information of that someone buy food that is considered unhealthy, or that this person buys medicine, could be used by an insurance company to tend to increase the insurance rate.
- To do overtime can be interpreted as health risk, or that persons are not very good in their particular job.
- Buying products or searching for keywords which categorise someone in your family as pregnant could influence the credit rating, or could turn into a handicap to apply for certain jobs.

All those information could be extracted from your daily behaviour by operators of third-party services. In most cases the information are not simply to spy users, since there is usually a trustworthy relation between the users and the information holder. But there are situations in which these collected information could eventually passed to a different authority, even without the knowledge of the users, either by simply selling them to other companies, or if a company is sold or goes out of business. In the later cases the originally trustworthy relation has ended, but the sensitive user data are still present.

The given example describes a complex problem with a few words. Users leave digital footprints, which are individually not meaningful, but by combining all these

[3] More information on Kasumigaseki Cloud:
`http://www.cloudbook.net/directories/gov-clouds/gov-program.php?id=100016`

single footprints valuable information might be eventually extracted with profiling techniques at a later stage. Furthermore, this profiling can even led to the categorisation of users to groups with similar behaviour by so called group profiling [18].

The categorisation could be even more problematic than to extract single information, because the ranking of individuals is therefore typically dependent on group parameters. In other words, the individual might get disadvantaged by sharing this group, whose parameters are based on specific algorithms but eventually effected by statistical fluctuations. Additionally, this process is not transparent to users of the system. In an extreme example, the credit-risk of a person for a contract could be based on those group characteristics for which the person is member of, such as its place of residence or age [19]. In a similar manner to the given example the accounting data of the daily work of users can be interpreted to get information about their behaviour, including daily work habits, e.g. how the work proceed or simply that overtime is needed each second weekend. If users are not informed about the profiling they have no chance to check the results and have to live with the consequences.

More generally, there is a need to deal with the right to informational self-determination in an appropriate manner. In short, informational self-determination is the right of an individual to control which personal information are used under which circumstances. This right was first formulated as a discrete right [20] by the German Bundesverfassungsgericht [21]. Nowadays, similar rights are established e.g. for the European Union and United States of America [22].

In case of scientific communities, there is no direct commercial interest of categorising people. E.g. the D-Grid (the German Grid community) uses resources of data centres of universities, which are in principal operated in the same way and therefore, the universities as resource providers have to respect the right to informational self-determination. This means there is the demand to avoid that detailed personal or project related information can be extracted out of the users behaviour, in particular if an external provider is used.

In the academic domain the user groups are rather manageable, limited in number and not highly dynamic. But there is also the trend to combine computing infrastructures over institutional boundaries (e.g. in Grids) or incorporate other service providers.

In this context, accounting data are again sensitive information and could potentially be used to analyse the users behaviour by third parties. To minimise the danger that users are traced, the accounting information have to be reduced to a minimal level, which is only needed to operate a billing service. This reduction mechanism is what we call data minimisation. We will show that data minimisation can be easily introduced for various systems (in the following we will show this for accounting systems).

4 Accounting Using Data Minimisation and Anonymisation

In the following we consider an architecture which allows the federation of services across different providers. By combining services from different providers it is possible to create work-flows or high level services. To ease this process it is handy to introduce an abstraction high-level access layer, where all services are presented within a global address space.

In such a service infrastructure we still expect that the basic services are operated by distinct operators but have to be registered at a central point, usually available via service repositories (e.g. [23,24,25]). The basic model assumption is schematically visualised in fig. 1.

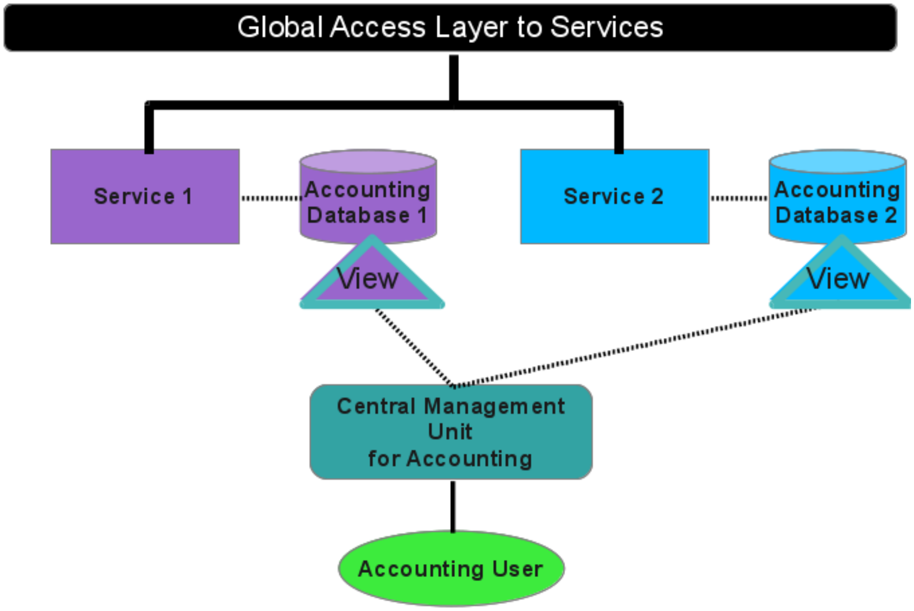

Fig. 1. General architecture of an accounting system. The solid lines show the access to data and services, while the dotted lines show the transfer of monitoring information.

In this paper we only consider accounting as a separate component which can be operated independently from any generalised global access layer, in our abstract scenario provided by the federation services. The global access layer (shown as separate component in fig. 1) can include a global name-space, management of user accounts and enabling Single Sign on (SSo) to all services.

Especially in federated systems, to bring together all relevant user data for necessary financial clearing, a central management component to access accounting data is demanded to realise an easy to use accounting and billing system. This component does not directly access the providers data store, but only indirectly via so called "views". Properties of these views are explained more in detail in section 4.2.

4.1 Aspects of a General Accounting Architecture

The main components of our approach are connectors to the local storage systems, in which the relevant accounting data are stored at each service provider. This way, the storage systems from each provider remain clearly separated and from the central management unit only the relevant aggregated information can be accessed. This ensures that the users behaviour is potentially only still gettable by the local providers. It is therefore not directly possible to combine the knowledge of multiple providers by a central unit to rank or profile users. This minimal management unit provides the central access point to regulate accounting and access reports for statistical analysis, billing or financial clearing by using an aggregation service (view), which is under control of the service providers. This architecture is shown in fig. 1.

The accounting management unit provides a user interface for easy configuration and adaptation of price and billing models. It also provides a central interface to get all data needed for billing and financial clearing based on the SLAs which signed by the resource providers and users, which might belong to companies or organisations.

As can be seen in fig. 1, the accounting system is not integrated in the global access layer. This means the transfer of the aggregated accounting data is triggered by the management unit. Also, the addresses to the views (provided by the service providers) and the corresponding logins have to be registered at the management unit.

In comparison to the accounting concepts of the systems mentioned in section 2, we introduce a concept of data minimisation. This is done by aggregation and anonymisation of the accounting data. The complete accounting data are only accessible for the local service provider. In a federated system it is of course not desired to make them visible for providers of other services[4] within the same environment or even the management unit. To restrict this direct access the management unit can not address the providers sensible accounting data as a whole. In order to realise a billing service, only the summed up information are transferred to the management unit, which are provided by the views.

4.2 Views

To transfer only relevant billing data, our approach to realise a data minimalistic access to this sensitive user data, is based on so called views. A view is a transformation of the accounting data to a report. This transformation only considers the needed minimal set of available user data to provide necessary information for the billing service.

The service provider is responsible for collecting the local accounting data. Accounting data for other services are completely out of scope for this provider, even if the operated service is part of a complex service orchestration. To be responsible for local accounting means to define which events and parameters for each service request have to be recorded. Therefore, the concrete realisation strongly depends on the provided service. E.g. a service for storing data probably needs to record the time stamp, the local account name of the user, the file size and whether the file was written or read, while a search service probably needs to record how much computing power was used to perform the request or how many data sets are read to give an result.

[4] Even if multiple service providers are needed to fulfill a single user request each provider has only access to the accounting data of the work processed on his service.

In the responsibility of the service provider is the safety and security of the accounting data, which are strongly related to individual persons. This also includes not to give information about users to other parties, or only in an anonymous way if necessary for billing purposes. A view can periodically be created (e.g. once a month) and contains the information which resources are used and how much has to be paid for this usage. The depth of detail which is required for such a report is low in most cases. To illustrate this let us give some examples:

1. The utilisation of a service can be calculated by knowing the number of requests to the service. It is not needed to know at which particular time or who triggered the request.
2. To bill a user, only the aggregated price information over the billing period is needed and not the individual services used.
3. A more detailed report is also possible (if demanded by the user). This could be the number of uses and the price individually for each service and each time slice with a special price. Such a report could contain the number of files stored during rush our (period with high price), stored during normal working time and during periods with low system utilisation (at night and weekend). The data are still aggregated and it is not reported at which exact time a service was used by the user.
4. Often it is not needed to bill single users. If users are part of a company or an organisation the report dos not contain the users identity. The report can be structured like in the examples above with the expect that only the summed up usage of all user of a group are presented. This results in an anonymity of individual users within the group.

These reports are based on SLAs between users, user groups or their representatives (e.g. VOs) and resource providers and describe the information. As already mentioned the accounting data are recorded by the resource providers and stored locally, e.g in a database (shown in fig. 1). Accessible by the central management unit are only the views. Technically a view provides a report which is an aggregation of accounting data. Depending on the particular aggregation process it can also anonymise by simply hiding information with directly link to individual users, like exemplified given by example 4. The view represents the instructions how the data are aggregated and how the price is calculated on the basis of this informations.

In the example 1 from above, selected are all records of the accounting data (the limitation to the reporting interval is automatically added by the system). Based on this view a report is created which hides the records itself and just contains the overall usage for the billing period.

Lets consider example 3, where a view for each price category is needed. One specific view selects all accounting data of the user for which the rush hour price has to be paid and calculates the price for a billing period (e.g. number of requests multiplied by the price per request). The views for the other pricing models are used in the very same way. Taking that example 1 and example 3 rely on the same price model like in example 1 only one view is sufficient for both cases. This way it is hidden whether the user performed many requests in a time period, for which a low price is active, or less during rush hour with a higher prize per service utilisation. In example 1 the view has to select all records of the user and the price calculation has to respect the individual price model for each record type. Thus, the complexity of the price calculation is slightly larger compared to example 3, but to the management component only the result of this calculation is reported.

If many users are combined in one view (example 4) the selection has to opt out the users e.g. by their account or group names. The price calculation is done similar to example 1. In this way the view combines the records of many users which results in anonymity within the group.

The views are created on the central management unit. The request for changing views are automatically transferred to the resource providers which have to check and implement the views. Once the view is active the central unit can get the reports. Altering or deleting a view is the same procedure like creating a new one. In this case it has to be ensured that deleted or overwritten views can still be checked by the resource providers. All requests from the central unit to alter a view are logged by the resource provider.

5 Reference Implementation

The accounting concept presented in this paper was developed for the knowledge infrastructure WisNetGrid[5], which offers a uniform access layer to data, information and knowledge. The access layer can connect sources from different providers using technically different storage and access systems and solutions for authentication and authorisation. By combining different sources of data, information and knowledge it is possible to use services for knowledge processing and knowledge generation.

The reference implementation of the presented accounting concept is part of a federation system and consists of components for user management, authentication and a web portal, which allows the use of services, such as searching and browsing of data, or tools for service management and workflow composition.

The accounting concept is implemented by following the concepts introduced in section 4.2. The accounting data are recorded by the operators of the potentially distributed resources. Within WisNetGrid we have realized a specialized federation entity to different types of data sources, such as databases, or Grid storage systems, which are necessary to create the common access layer. Each operator of a connected system stores the recorded accounting data in a separate accounting database. For this we use a H2 database[6] because this allows to drive the database as part of the resource federation entity, which is implemented using Java.

[5] The WisNetGrid Project is funded by the German Federal Ministry of Education and Research (BMBF), more information at: `http://wisnetgrid.org/`

[6] For more information about H2, see: `http://www.h2database.com`

The interface for billing is a central component within the WisNetGrid architecture. It offers different visualisation features to get an overview which price models had been used and how much has to be charged. A comparison to actual price models can also be made and visualised if desired. To use the results of this centralised accounting component in other programs (e.g. for the process of financial clearing) the data can be exported as CSV files. CSV is a common format and can be used by various programs for further processing.

The accounting component offers a restricted database access, which is realised by the views introduced in section 4.2. In this specific implementation the addresses and logins to the views are part of the configuration of the resource federation. This information is automatically transferred to the accounting component by registering a resource federation entity as part of the global access layer. If the resource provider offers accounting, the aggregated accounting information are automatically integrated to the central billing interface.

The management of the views is done in two steps. The accounting component offers a graphical user interface which can be accessed via a browser (by users authorised as accounting users) to delete, create or alter views. For this the selection and price calculation parts have to be specified. This is done by filling a form with SQL syntax. After submitting the form the accounting component extracts the information and stores them in a database at the resource federation instance. The second step is done manually by the operator of the resource federation or automatically by implementing a trigger on the database. Which mechanism is used depends on the configuration of the resource federation entity. Based on the request a SQL statement is build to create, alter or delete a database view. The “WHERE” clause is based on the selection part and the price column is based on the price field information from the filled form of the first step. Additionally, a “GROUP BY” clause over the reporting interval is added (e.g. “GROUP BY year, month” where year and month are fields of the accounting data). Afterwards the new view can be used by the portal to visualize accounting results according to the selected view.

6 Conclusion

We have presented a concept for accounting with privacy preserving for users, which is taking also data minimisation and anonymization into account. This was presented on a concrete implementation for the knowledge infrastructure WisNetGrid. This accounting concept allows to perform billing and financial clearing in a similar way compared to common centralized accounting systems, which are usually not designed with a strong focus on privacy preserving. A valuable field of application outside of the concrete implementation can be spotted for Grid and Cloud Computing, which was shortly discussed throughout this paper. Furthermore, we hope to inspire readers to further strengthen the user right of informational self-determination for all kinds of projects, which combine services from different partners or providers, where user data and behaviour are always sensitive information.

References

1. Sandholm, T.: Design Document: SweGrid Logging and Usage Tracking Service, LUTS (2003)
2. Piro Rosario, M., Andrea, G., Giuseppe, P., Albert, W.: Using historical accounting information to predict the resource usage of grid jobs. Future Generation Computer Systems 25(5), 499–510 (2009)
3. Elmroth, E., Gardfjäll, P., Mulmo, O., Sandgren, Å., Sandholm, T.: A Coordinated Accounting Solution for SweGrid Version: Draft 0.1.3 (October 7, 2003)
4. Mihailescu, M., Teo, Y.M.: Dynamic Resource Pricing on Federated Clouds. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 513–517 (May 2010)
5. Parameswaran, A.V., Chaddha, A.: Cloud Interoperability and Standardization. SETLabs Briefings 7(7) (2009)
6. Lee, C.A.: A perspective on scientific cloud computing. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, HPDC 2010, pp. 451–459. ACM, New York (2010), http://doi.acm.org/10.1145/1851476.1851542
7. Sotomayor, B., Montero, R., Llorente, I., Foster, I.: Virtual Infrastructure Management in Private and Hybrid Clouds. IEEE Internet Computing 13(5), 14–22 (2009)
8. Nurmi, D., Wolski, R., Grzegorczyk, C., et al.: The Eucalyptus Open-Source Cloud-Computing System. In: 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009, pp. 124–131 (May 2009)
9. Rochwerger, B., Breitgand, D., Epstein, A., et al.: Reservoir - When One Cloud Is Not Enough. Computer 44(3), 44–51 (2011)
10. Buyya, R., Ranjan, R., Calheiros, R.N.: InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. In: Hsu, C.-H., Yang, L.T., Park, J.H., Yeo, S.-S. (eds.) ICA3PP 2010, Part I. LNCS, vol. 6081, pp. 13–31. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-13119-6_2
11. Repschlaeger, J., Wind, S., Zarnekow, R., Turowski, K.: A Reference Guide to Cloud Computing Dimensions: Infrastructure as a Service Classification Framework. In: Hawaii International Conference on System Sciences, pp. 2178–2188 (2012)
12. Bernsmed, K., Jaatun, M., Meland, P., Undheim, A.: Security SLAs for Federated Cloud Services. In: 2011 Sixth International Conference on Availability, Reliability and Security (ARES), pp. 202–209 (August 2011)
13. Badger, L., Bohn, R., Chu, S., et al.: NIST Special Publication 500-293, US Government Cloud Computing Technology Roadmap, Release 1.0 (Draft), Volume II Useful Information for Cloud Adopters
14. Sekar, V., Maniatis, P.: Verifiable resource accounting for cloud computing services. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, CCSW 2011, pp. 21–26. ACM, New York (2011), http://doi.acm.org/10.1145/2046660.2046666
15. Ruiz-Agundez, I., Penya, Y.K., Bringas, P.G.: A Flexible Accounting Model for Cloud Computing. In: Proceedings of the 2011 Annual SRII Global Conference, SRII 2011, pp. 277–284. IEEE Computer Society, Washington, DC (2011), http://dx.doi.org/10.1109/SRII.2011.38
16. Elmroth, E., Marquez, F., Henriksson, D., Ferrera, D.: Accounting and Billing for Federated Cloud Infrastructures. In: Eighth International Conference on Grid and Cooperative Computing, GCC 2009, pp. 268–275 (August 2009)

17. Chandra, D., Borah Malaya, D.: Problems & prospects of e-Governance in India. In: 2011 World Congress on Information and Communication Technologies (WICT), pp. 42–47 (December 2011)
18. Hildebrandt, M.: Profiling: From data to knowledge. Datenschutz und Datensicherheit - DuD 30, 548–552 (2006), http://dx.doi.org/10.1007/s11623-006-0140-3, doi:10.1007/s11623-006-0140-3
19. Metz, R.: Scoring: New Legislation in Germany. Journal of Consumer Policy 35, 297–305 (2012), http://dx.doi.org/10.1007/s10603-012-9191-z, doi:10.1007/s10603-012-9191-z
20. Hornung, G., Schnabel, C.: Data protection in Germany I: The population census decision and the right to informational self-determination. Computer Law & Security Review 25(1), 84–88 (2009), http://www.sciencedirect.com/science/article/pii/S0267364908001660
21. Bundesverfassungsgericht: BVerfGE 65, 1 – Volkszählung. Urteil des Ersten Senats vom 15. Dezember 1983 auf die mündliche Verhandlung vom 18. und 19. Oktober 1983 – 1 BvR 209, 269, 362, 420, 440, 484/83 in den Verfahren uber die Verfassungsbeschwerden (1983), http://www.datenschutz-berlin.de/gesetze/sonstige/volksz.htm (access November 21, 2006)
22. Rehm, G.M.: Just Judicial Activism? Privacy and Informational Self-Determination in U.S. and German Constitutional Law (January 2000), available at SSRN http://ssrn.com/abstract=216348 or http://dx.doi.org/10.2139/ssrn.216348
23. Wu, Y., Yan, C., Ding, Z., et al.: A relational taxonomy of services for large scale service repositories. In: 2012 IEEE 19th International Conference on Web Services (ICWS), pp. 644–645 (June 2012)
24. Weiping, L., Weijie, C., Li, L., Fuliang, G.: A semantically enhanced service repository for service oriented application system development. In: World Conference on Services - II, SERVICES-2 2009, pp. 41–48 (September 2009)
25. Agarwal, S., Junghans, M., Jäkel, R.: Semantic modeling of services and workflows for german grid projects. In: Grid Workflow Workshop 2011 (2011)