Towards Agent-based Data Privacy Engineering

Kato Mivule
Computer Science Department,
Bowie State University
Bowie, MD, USA
kmivule@gmail.com

***Abstract* – While a number of agent-based software engineering frameworks have been proposed in the recent years, a few have been suggested specifically for the data privacy procedure. Yet still, one of the challenges in designing agent-based data privacy frameworks is that the very definition of privacy remains ambiguous and a case-by-case approach would have to be adopted. Therefore, as a contribution, we take a look at the literature on agent-based software engineering and present SIED (Specifications, Implementation, Evaluation, Dissemination), a conceptual framework that takes a holistic approach to the data privacy engineering process by looking at the Specifications, Implementation, Evaluation, and finally, Dissemination of the privatized datasets by autonomous intelligent agents.**

Keywords – Data privacy engineering; autonomous agents; statistical disclosure control.

I. INTRODUCTION

In 2009, a privacy-by-design challenge was put forward and described by Cavoukian [1], in which privacy is entrenched and embedded into the engineering requirements of different methodologies and technologies [2]. Moreover, recent revelations by Edward Snowden concerning covert electronic operations by US Government security agencies and the alleged infringement of personal privacy [3], have pushed to the forefront the importance and necessity of privacy by design, and in this case, engineering privacy into the design of software and autonomous multi-agents. Yet still, engineering data privacy remains an ongoing challenge largely due to what considerations the definition of data privacy should encompass [4][5]. Consequently, one of the problems of data privacy engineering, is that the notion of privacy is ambiguous, normally misidentified with data security, thus making it difficult to engineer and implement [4][5][6][7]. To appropriately design and implement data privacy agents, an all-encompassing approach for describing data privacy should entail the legal, technical, and ethical features; as such, providing an understandable logical context for all shareholders in the data privacy process [8]. While efforts have been made to theoretically explain data privacy, human perceptions such as, ambiguousness and evolutions of personal understanding of privacy, remain a crucial influence in the design and implementation of data privacy [9]. As a result, any design and implementation of privacy agents has to imperatively consider what personal information entities see as appropriate for public revelation [5][6][7]. Therefore, to assist in a thorough data privacy requirements elicitation, we employ software engineering concepts outlined by Sommerville (2010), and have been effectively used to capture ambiguous requirements in the software engineering domain [10].

As a contribution, a literature review on agent-based software engineering frameworks is presented; SIED, as conceptual framework for agent-based data privacy engineering, is suggested. Moreover, to aptly deal with the intricacy of data privacy engineering, the abstraction, decomposition, and hierarchical perspectives of dealing with complexity as outlined by Booch (1994) have to be considered [11].

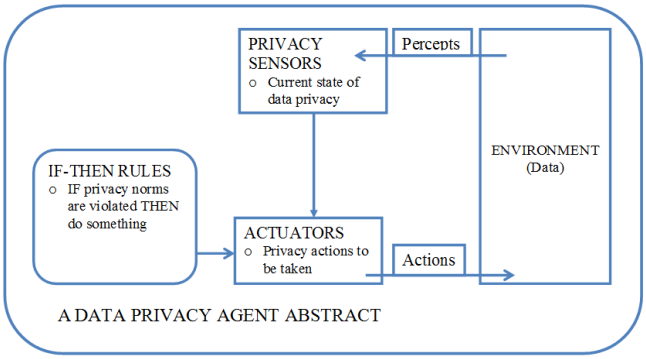

Figure 1. An Abstract of a data privacy agent.

Moreover, our aim in this paper is to give an abstraction and conceptual view (Figure 1) of an agent-based data privacy engineering framework, while keeping the decomposition and hierarchical aspects to future works. At the same time, while SIED is proposed as a framework for agent-based data privacy engineering, the same framework could be generalized for basic non-autonomous data privacy implementations. The rest of the paper is organized as follows. In Section II, a review of literature on related work is given. In Section III, the SIED conceptual framework is explained. In Section IV, a conclusion and future works is given.

II. RELATED WORK

While few works exist on engineering privacy in agents, considerable amount of work has been done in the area of agent-based software engineering; providing principles that could be applied in the data privacy engineering domain. For instance, Wooldridge (1997) [12] noted three essential

 ISBN: 978-1-61208-340-7

considerations when engineering multi-agents, namely, (i) how agents should be specified, (ii) how to turn the specifications into agent implementation, and lastly (iii) how to validate that the newly developed agent meets the original specifications. The Wooldridge (1997) [12] three essential considerations are relevant for the design and engineering of data privacy agents, in that a meticulous elicitation of privacy specifications, in this case requirements, has to be done. Yet still, Wooldridge and Jennings (1999) [13] warned about some of the pitfalls when it comes to agent-based software engineering. Wooldridge and Jennings (1999) [13] noted that one of the common pitfalls, is the tendency to offer generic architectures for intelligent agents, yet such costly one-size-fits-all architectures would rarely work for every agent-based software problem. This observation by Wooldridge and Jennings (1999) [13] is an essential consideration for agent based privacy engineering, as each privacy problem tends to be unique and based on the privacy definition of that particular user [5]. Additionally, Jennings (2000) [14] observed that while agent based software engineering was being used to address real world problems, building such systems remained complex and difficult, due to the interactions between different components that are rigidly defined, and inadequate methods available to represent a systems architecture. The Jennings (2000) [14] study is applicable when it comes to agent based privacy engineering. The very definition of what constitutes privacy, makes building such systems complex and therefore a case-by-case perspective has to be done, especially when communication and data transaction between various autonomous privacy agents is taken into account.

Yet, from a legal perspective on technology, calls for Privacy Enhancing Technologies (PET) were issued as in the case of Borking and Raab (2001) [15], who made an elaborate elucidation of PETs and how data safety systems and legal processing of personal data could be enhanced by such technologies. Among the guidelines noted by Borking and Raab (2001) [15] are (i) reporting of data processing, (ii) transparent data processing, as required data processing, (iii) legitimacy of the data processing, (iv) data quality, (v) rights of parties involved in the data processing, (vi) data traffic across international borders, (vii) processing personal data by a processor, and (viii) protection against loss and unlawful processing of personal data. In the same period of time, Kenny and Borking (2002) [16] defined privacy engineering as a methodical endeavor to embed privacy applicable legal primitives into technological and governance blueprints. Kenny and Borking (2002) [16] proposed DEPRM, a Design Embedded Privacy Risk Management framework, to integrate both privacy engineering and risk management. While Kenny and Borking (2002) [16] did not distinctively define privacy engineering for the purposes of designing autonomous agent systems, their definition of privacy engineering certainly remains relevant and pertinent in designing privacy conscience agents today. Furthermore, Van Blarkom, Borking, and Olk (2003) [17], in their case for PETs in intelligent software agent systems argued that PETs would be helpful in tackling privacy threats caused by intelligent agents that illegally disclose a user's personal information. PETs would also help in dealing with threats caused by external intelligent agents that act on behalf of adversaries via traffic flow monitoring, data mining, and covert attempts to obtain personal information directly from a user. On a remarkable note, the term "privacy engineering" by Kenny and Borking (2002) [16], was being used and appearing in legal literature then, while mainstream software engineering and data privacy domains would begin to pick up this term at a later point. Research on privacy enhancing technologies was ongoing in the legal communities while such efforts were not obvious in the software engineering domain.

On the issue of norms and behaviors in intelligent agents, y López, Luck, and d'Inverno (2004) [18], proposed a normative framework that would instruct agents on how to behave prescriptively, socially, and under peer pressure. y López et.al, noted that autonomous agents while working to satisfy their own goals, still have to comply with social responsibilities [18]. While a number of norms could be considered for agent based software engineering, in this article, we are interested in what privacy norms an autonomous agent could be engineered to observe. For example, not revealing an entity's sensitive information could be considered as a social norm that an autonomous agent would be expected to observe. On agent-based software engineering, Bresciani, Perini, Giorgini, Giunchiglia, and Mylopoulos (2004) [19], proposed Tropos, an agent based software engineering methodology that utilized the very definition of an intelligent agent, its, goals, plans, and environment in software requirements and implementation phases. While Tropos provided a framework for the development of agent-based software, engineering privacy in the design of such agents was not the main focus, a trait in many earlier agent-based software engineering frameworks. Besides, Zambonelli and Omicini (2004) [20] observed and argued at that time, that while agent-based software engineering was experiencing a great amount of research, one of the challenges included how to turn generated agent-based software abstractions into practical tools to solve complex problems. Yet still, to this date, the same challenge remains when it comes to privacy. Given the complex and ambiguous definition of privacy, turning generated agent-based privacy engineered abstracts into real useful tools that could help solve some of the privacy problems, remains a challenge.

On the other hand, Sooyong and Vijayan (2005) [21], proposed using a goal based approach in the problem domain requirements analysis such that each autonomous agent could appropriately get mapped to the system's refined goals. In this paper, we take a similar approach to Sooyong, and Vijayan (2005) [21], by emphasizing the specifications phase of the engineering process to comprehensively map out the environment, goals, and actions of a data privacy agent. Bellifemine, Caire, Poggi, and Rimassa (2008) [22], gave an elaborate overview on JADE, a Java based software framework for developing multi-agent applications. While JADE is still a popular framework utilized to this date, the challenge is how to implement agent based privacy engineering using JADE.

However, Weyns, Parunak, and Shehory (2008) [23] argued that despite the interest in agent based software engineering research, implementation was still a challenge due to disconnect between proposed frameworks in academia and implementation in industry. Weyns et al. (2008) [23] observed that this disconnect between academia research and adoptability in industry was largely due to a poor understanding of industry needs. To address this problem in the privacy domain, we suggest a thorough case-by-case requirements analysis in the specifications phase of an agent development. Cossentino, Gaud, Hilaire, Galland, and Koukam (2009) [24], proposed ASPECS, a framework that utilizes a holonic structural meta-model and offers a step-by-step monitoring, from requirements to implementation, with modeling in each phase of the development cycle. However, Léauté and Faltings (2009) [25], proposed an agent based privacy engineering solution using constraint satisfaction model by mapping out privacy constraints in the domain. In such a scenario, each agent makes decisions that keep with the privacy norm – constraints in this case; for example, by not revealing sensitive information when communicating with other agents [25]. On the subject of meta-modeling, Gascueña, Navarro, and Fernández-Caballero(2011) [26], observed that Model-Driven Engineering (MDE) allowed developers and stakeholders to use abstractions closer to the domain than generalized computing concepts. However, due to the relatively growing research on agent-based data privacy engineering, not many such models exist.

Furthermore, Cavoukian (2011) [27] outlined seven privacy by design principles that included: (i) proactive, not reactive privacy design; preventative not remedial design approach; (ii) engineering privacy as the default; (iii) privacy embedded into design; (iv) full privacy functionality by avoiding needless trade-offs; (v) end-to-end security and life cycle protection privacy design; (vi) visibility and transparency of privacy practices; and finally (vii) respect for user privacy. However, to fully meet the seven privacy by design principles outlined by Cavoukian (2011) [27], we strongly believe that a comprehensive specifications and requirements solicitation and analysis has to be done, especially when it comes to engineering data privacy agents. More recently, Such, Espinosa, and Garcia-Fornes (2012) [28], in their extensive survey on privacy in multi-agent systems, noted that the concern of privacy in multi-agents is still a problem, and has increased due to the robust growth and utilization of the internet for data transaction. Among the privacy violations that autonomous agents engage in, as noted by Such et.al., include, (i) secondary use, such as profiling, (ii) identity theft, (iii) spy agents, (iv) unauthorized access, (v) traffic analysis, and (vi) unauthorized dissemination of data [28]. Such et.al., argued that to combat some of these agent based privacy vices, agent based privacy solutions should be incorporated in the design of information technology systems [28].

Nevertheless, Aggarwal and Singh (2013) [29], presented a mechanism for the reuse of already existing software agents in the development of specific software, by utilizing the abstract description of an agent and reusing such systems in other specific domains. Still, as in the case with Gascueña et.al. (2011) [26], on model-driven engineering and reusing abstractions that are closer to the domain, the Aggarwal and Singh (2013) [29], model of reuse, would not be without challenges in the data privacy domain. Engineering such agents remains difficult and would have to be done on a case by case basis, due to the very subjective definition of what privacy is among various entities. As we noted in Mivule, Josyula, and Turner (2013) [6], the definitions of privacy vary, are fuzzy, indistinguishable, and are largely attached to how humans see privacy and what data they are willing to share or consider private. However, despite such challenges, intelligent autonomous agents offer possibilities when it comes to engineering privacy in agents. For instance, agents could be designed to learn privacy norms after a methodical privacy requirement analysis is done for that specific case.

III. THE SIED FRAMEWORK

The motivation behind the SIED framework is to create a systematic outline that can be followed for the data privacy engineering process. Given any original dataset *X*, a set of data privacy engineering phases should be followed from start to completion in the generation of a privatized dataset *Y*.

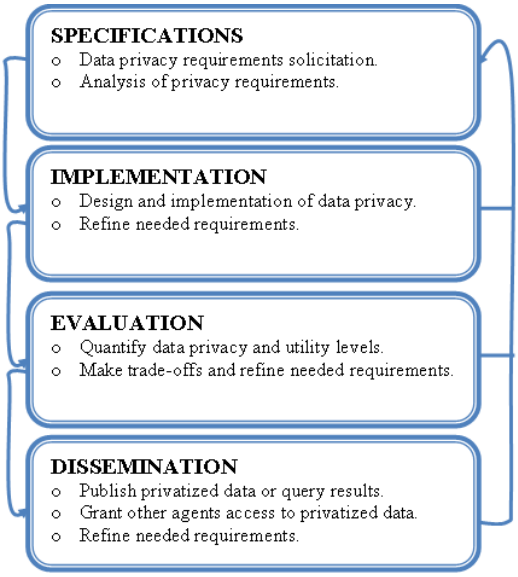

Figure 2. The SIED conceptual framework.

In this article, SIED, as shown in Figure 2, is proposed as a holistic conceptual approach that could be employed for the data privacy engineering process. The four main phases of the SIED data privacy engineering framework are as follows:

A. *Specification phase*

In this phase, data privacy engineers gather data privacy specifications and requirements from the client. In the suggested SIED framework, requirements solicitation and analysis is the most crucial phase of the agent-based data privacy engineering process. While a series of questions could be generated to comprehensively assess the data privacy requirements of a user, we suggest the following

questions as being essential for a holistic agent-based data privacy specifications analysis:

- What are the data privacy legal and policy compliance requirements?
- What is the client description for Personal Identifiable Information (PII), quasi, sensitive, and non-confidential attributes?
- What are the current client data privacy threats or vulnerabilities?
- How far would client data be affected by auxiliary data?
- How is the client planning on dissemination of privatized dataset?
- Will privatized data access be by query access, published categorical data, or tabulated data?
- Will the privatized dataset be in microdata or macrodata form?
- What type of original data from the client is to be handled, continuous or categorical?

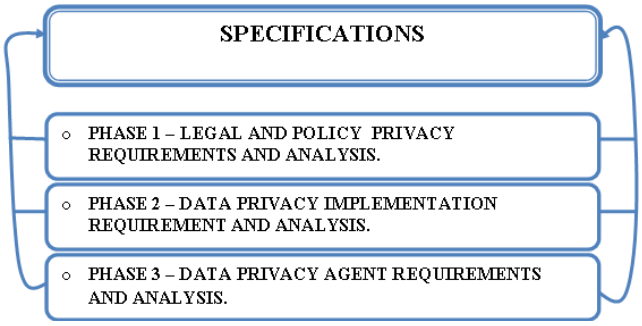

Figure 3. Suggested specification phases.

- What is the variable size for the original data, univariate or multivariate?
- What type of partitioning on the original data will be required, horizontal or vertical partitioning?
- What types of Statistical Disclosure Control (SDC) methods are required by client, non-perturbative, or perturbative?
- What nonfunctional requirements are suggested by the data privacy engineers?
- What is the client expected data privacy needs?
- What is the client expected data utility needs?
- What trade-offs can be accommodated between data privacy and utility needs?

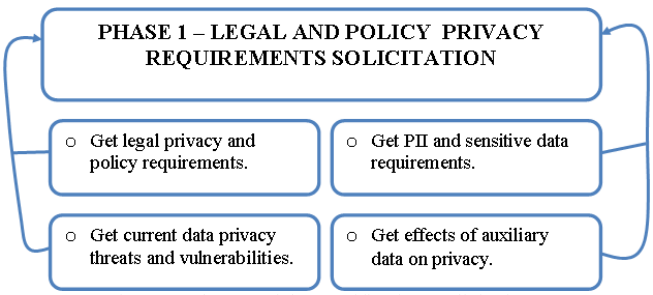

Figure 4. Phase 1 of the specifications solicitation.

While this set of solicitation questions is not exhaustive, the responses generated could be used to construct a set of beliefs and inference rules for the data privacy autonomous agent. In addition, specification is done in three requirements phases, as shown in Figure 3, namely, (i) Legal data privacy requirements solicitation and analysis, (ii) Data privacy application requirements solicitation and analysis, and (iii) Data privacy agent requirements solicitation and analysis.

Phase 1: Legal data privacy requirements solicitation and analysis: In the first phase of specification analysis, a review of issues pertaining to legal privacy and policy compliance is done, as illustrated in Figure 4:

- Solicitation and analysis of legal privacy and policy compliance requirements is done.
- Assessment of user description of what constitutes PII, quasi, sensitive, and non-confidential attributes is carried out.
- Assessment of current client data privacy threats or vulnerabilities is done.
- Assessment of how user data privacy would be affected by auxiliary data, such as, posts on social media is carried out.
- Assessment of privacy threats and vulnerabilities, including effects of auxiliary data, is done in Phase 1.

Requirements solicitation and analysis generated from this phase will later be used in creating a beliefs set and IF-THEN rules for the autonomous privacy agent.

Phase 2: Data privacy application requirements solicitation and analysis: In the second phase of the specification analysis, a review of issues pertaining to how data privacy will be implemented by the autonomous privacy agent is done, as shown in Figure 5.

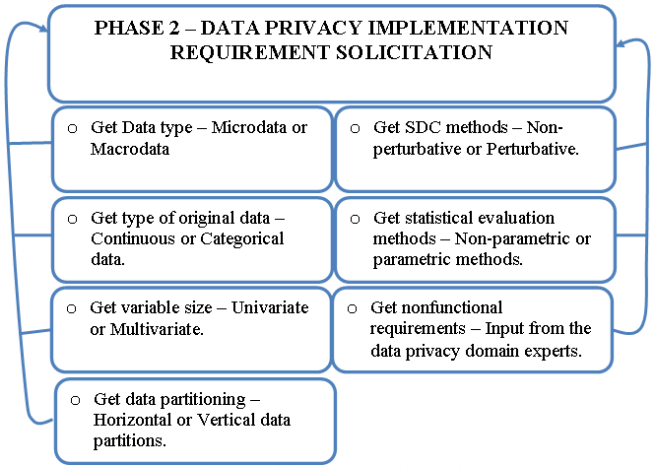

Figure 5. Phase 2 of the specifications solicitation.

The main question asked in this phase is what action the data privacy agent would take in response to a privacy violation. The type and characteristics of the data should be analyzed and included in the specifications. As such, the data privacy agent should be able to do the following:

- Assessment of the data type – microdata or macrodata form.

- Assessment of the type of original data from the client to be handled – continuous or categorical.
- Assessment of the variable size for the original data – univariate or multivariate.
- Assessment of the partitioning on the original data – horizontal or vertical partitioning.
- Assessment of SDC methods required – non-perturbative or perturbative.
- Assessment of evaluation requirements – non-parametric or parametric methods.
- Assessment of the nonfunctional requirements from the data privacy engineers.

Non-functional requirements, as noted by Summerville (2010), are requirements not suggested by the client but suggested by an expert in the field who might see other necessary needs that a non-expert might not see; in this case, the expert is the data privacy engineer [10].

Phase 3: Data privacy agent requirements solicitation and analysis: The requirements solicitation and analysis done in the Phase 1 and Phase 2 is utilized to generate specifications for the data privacy agent. Following Wooldridge's (1997) [12] articulation on the characteristics of an agent, it is at this point that subsequent features of a detailed agent-based data privacy requirements is done as outlined in Figure 6.

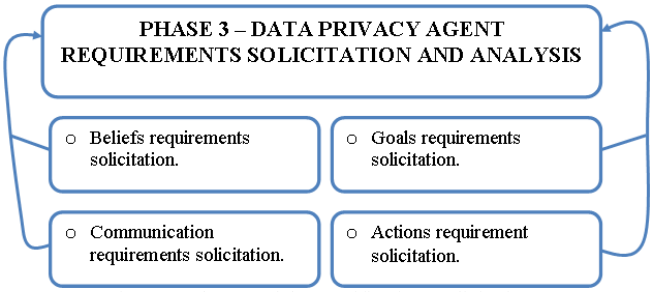

Figure 6. Phase 3 of the specifications solicitation.

- *Beliefs requirements solicitation*: Beliefs could include information about the environment concerning privacy, and the inference rules that need to be generated. Generated belief requirements could then be transformed into privacy centric IF-THEN statements and rules [30].
- *Communication requirements solicitation*: At this point, it would be important to know what privacy specifications and rules the agent should follow when communicating with other agents. This could include what privacy information is shared, when and how, with other agents.
- *Goals requirements solicitation*: In this solicitation phase, the overall goal of the agent should be stipulated. In the case of privacy, the aim of the agent is to ensure confidentiality of data by following a set of beliefs and inference rules.
- *Actions requirement solicitation*: Some of the questions that could be asked in this phase could include, what action should an agent do when PII is detected in a data set? In this case, the agent could decide to suppress, generalize, or perturb such sensitive information based on the inference rules.

B. Implementation phase

In this stage, design, application, and implementation of the appropriate specifications for the data privacy agent is done. Specifications are then used to build both the belief and action set of the data privacy agent. Appropriate data privacy algorithms for the appropriate data types are selected as part of the belief system for the data privacy agent. The implementation phase takes the specification analysis recommendations for implementation and executing the data privacy process. Various data privacy algorithms are chosen based on the specifications and requirement analysis and added to the agent belief set and action plan, as shown in Figure 3. A detailed description of the data privacy algorithms, statistical disclosure control methods, data characteristics, statistical analysis methods, and data partition methods, is given by Mivule and Turner (2013) [31]. The goal of the implementation phase, is to have the appropriate data privacy belief set generated from the requirements solicitation, that the agent could appropriately use to apply data privacy. Some of following set of IF-THEN statements and rules, generated and recommended from the user requirements, could be used as a set of beliefs for the data privacy agent, as highlighted in Figure 7:

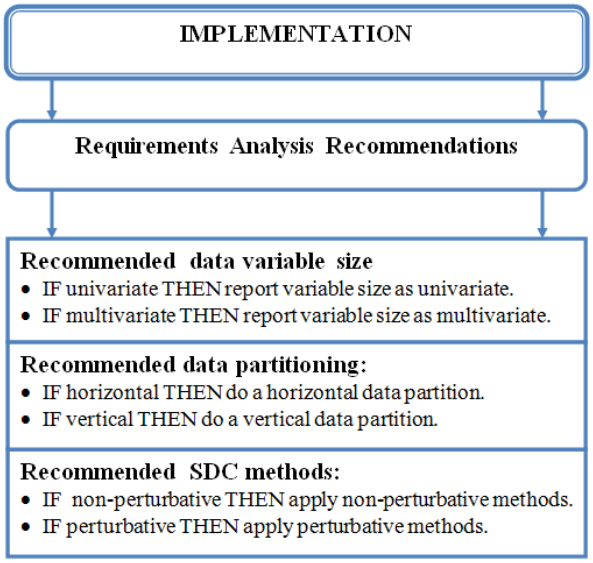

Figure 7. The SIED Implementation phase.

- *Detecting the data variable by agent*:
  - IF univariate THEN report variable size is univariate. IF multivariate THEN report variable size is multivariate [31].
- *Data partitioning by agent based on user requirements*:
  - IF horizontal THEN do a horizontal data partition. IF vertical THEN do a vertical data partition [31].
- *Type of SDC methods to be applied by agent*:
  - IF non-perturbative THEN apply the following non-perturbative methods:

 ISBN: 978-1-61208-340-7

Suppression, Generalization, k-anonymity, l-Diversity, among others [31].
    - IF perturbative THEN apply the following perturbative methods based on the dissemination method: Noise Addition, Multiplicative noise, Logarithmic multiplicative noise, Differential data privacy, Data swapping, and Synthetic datasets [31].
- *Statistical analysis to be applied on data by agent*:
    - IF numerical data THEN apply parametric methods. IF categorical THEN apply non-parametric methods.
- *Data dissemination by agent*:
    - IF privatized query results are requested THEN apply Differential Privacy. IF Privatized published micro and macro data are requested THEN apply Noise Addition [31].

C. Evaluation phase

In this phase, as shown in Figure 8, statistical evaluation of both original and privatized data is done by the data privacy agent. The goal of the agent at this stage would be to test if acceptable levels of both data privacy and utility are met, based on user requirements for a particular data set. Some of the evaluation questions that could be raised in this phase include:

- What are the expected client data privacy needs? What is the expected client data utility needs?
- What trade-offs can be accommodated between data privacy and utility needs?
- Answers to these questions would help formulate the evaluation belief set of the data privacy agent.

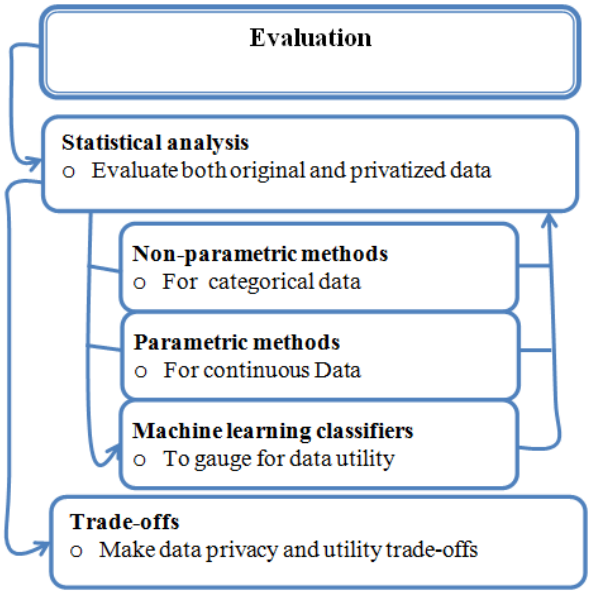

Figure 8. The SIED Evaluation phase.

Moreover, the data privacy agent would take into consideration the type of original data being handled. If the data type is numerical (continuous) then the agent would apply parametric statistical methods in evaluation, such as, mean squared error and entropy. If the data is categorical, the agent would apply non-parametric methods such as frequency count analysis. Because finding an optimal balance between privacy and utility needs is intricate [32], trade-offs are a necessity and made by the agent in this phase. It is in this phase that metrics used to measure data utility are implemented by the agent. Such metrics could include the mean, entropy; mean squared error, and classification error. The data privacy agent would measure the statistical traits of both the original and privatized data and find the difference. If the difference, say between the mean values of both the original and privatized data is higher than a set threshold (set by user or derived from the requirements solicitation), then data utility is low but privacy is high. The goal would be to find an optimal balance.

Furthermore, trade-offs are decided at this point in the evaluation phase. In addition, data privacy and utility expectations of the client are taken into consideration. Therefore, a data privacy agent would need to know at what point to autonomously make such trade-offs. While this is a difficult problem and would be one of the most challenging for the data privacy agent to make, a methodical specification analysis would help generate the appropriate belief sets and inference rules needed for the agent to take action. A number of evaluations could be considered in this phase:

- Assessment of the client expected data utility needs.
- Assessment of data privacy and utility trade-offs.
- Assessment of how privatized data would be disseminated.

For this particular data privacy engineering framework, we envision evaluation using machine learning classification as a gauge, as outlined in Mivule and Turner (2013) [7]. Basically, in the initial phase, the data privacy agent would apply privacy on data and pass the results through a machine learning classifier. The classification accuracy would be measured, with higher accuracy indicating better data utility but perhaps lower privacy. If the classification accuracy meets a set threshold, then the agent would proceed to publish the results, otherwise, the data privacy agent would adjust the parameters in the data privacy algorithm and then re-classify the data. The agent would repeat this process until the threshold criteria is attained.

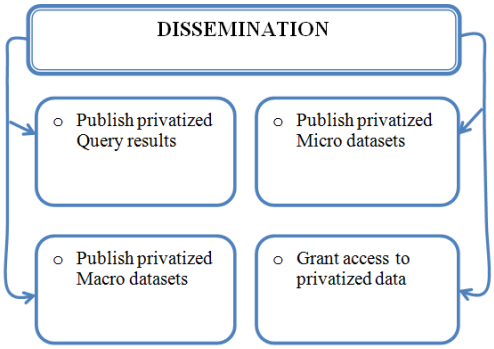

Figure 9. The SIED Dissemination phase.

 ISBN: 978-1-61208-340-7

D. Dissemination phase

The last phase of the SIED framework is concerned mainly with how the data privacy agent disseminates privatized data, as shown in Figure 9. Publication of the privatized data set would largely depend on user requirements. For instance if the user requires that query results to be privatized, differential privacy could be applied in the initial stages on the original data and the disseminated results would be privatized query results. On the other hand the user might require publication of micro and macro tabulated results. However, the requirements might be that the agent communicates privatized results to other agents for further processing. Therefore, agent dissemination of privatized data would largely depend on the user requirements.

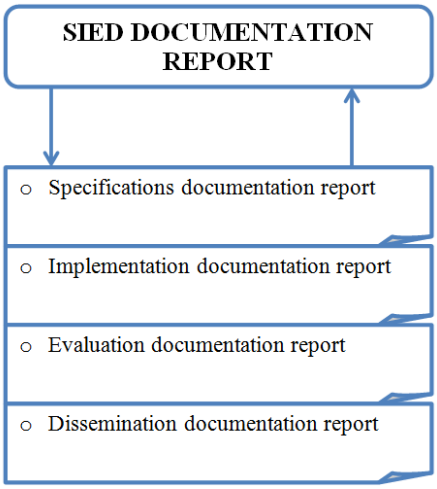

Figure 10. The SIED documentation outline.

As illustrated in Figure 10, documentation is done at every phase of the SIED data privacy engineering process, resulting in a final complete documentation report on the data privacy engineering process for that particular data set.

IV. CONCLUSION AND FUTURE WORK

We have presented an abstract view of SIED, an agent-based data privacy engineering framework that could be employed for a systematic data privacy design and implementation. We believe that this a contribution to the privacy-by-design challenge. In addition, we presented a literature review of related work on agent-based software engineering and the influence of such work on data privacy engineering. While SIED is proposed as a framework for agent-based data privacy engineering, it could be generalized for other non-agent-based data privacy processing as well. The subject of data privacy remains a challenge and more research, design, implementation, and metrics is needed to accommodate the ambiguous and fuzzy privacy requirements of individuals and entities. We believe that intelligent agent-based architectures offer optimism for data privacy solutions. For future works, we plan on expanding this study to take a decomposition approach in dealing with the complexity of agent-based data privacy engineering.

REFERENCES

[1] A. Cavoukian, Privacy by Design… Take the Challenge. Information & Privacy Commissioner of Ontario, 2009.
[2] A. Cavoukian, S. Taylor, and M. Abrams, “Privacy by Design: Essential for Organisational Accountability and Strong Business Practices.,” Identity Inf. Soc., vol. 3, no. 2, 2010, pp. 405–413.
[3] V. G. Cerf, “Freedom and the social contract,” Commun. ACM, vol. 56, no. 9, 2013, pp. 7.
[4] S. Spiekermann, “The challenges of privacy by design,” Commun. ACM, vol. 55, no. 7, Jul. 2012, pp. 38.
[5] V. Katos, F. Stowell, and P. Bednar, “Surveillance , Privacy and the Law of Requisite Variety,” 2011, pp. 123–139.
[6] K. Mivule, D. Josyula, and C. Turner, “An Overview of Data Privacy in Multi-Agent Learning Systems,” in The Fifth International Conference on Advanced Cognitive Technologies and Applications, 2013, no. c, pp. 14–20.
[7] K. Mivule and C. Turner, “A Comparative Analysis of Data Privacy and Utility Parameter Adjustment, Using Machine Learning Classification as a Gauge,” Procedia Comput. Sci., vol. 20, 2013, pp. 414–419.
[8] R. Dayarathna, “Taxonomy for Information Privacy Metrics,” J. Int. Commer. Law Technol., vol. 6, no. 4, 2011, pp. 194–206.
[9] G. J. Matthews and O. Harel, “Data confidentiality: A review of methods for statistical disclosure limitation and methods for assessing privacy,” Stat. Surv., vol. 5, 2011, pp. 1–29.
[10] I. Sommerville, Software Engineering, 9th ed. Addison-Wesley, 2010, pp. 27–50.
[11] G. Booch, Object Oriented Analysis & Design with Application. Santa Clara, CA: Addison-Wesley, 1994, pp. 14–20.
[12] M. Wooldridge, “Agent-based software engineering,” IEE Proc. - Softw. Eng., vol. 144, no. 1, 1997, pp. 26.
[13] M. J. Wooldridge and N. R. Jennings, “Software engineering with agents: Pitfalls and pratfalls.,” IEEE Internet Comput., vol. 3, no. Jun, 1999, pp. 20–27.
[14] N. R. Jennings, “On agent-based software engineering,” Artif. Intell., vol. Volume 117, no. 2, 2000, pp. 277–296.
[15] J. J. Borking and C. Raab, “Laws, PETs and other technologies for privacy protection.,” J. Information, Law Technol., vol. 1, 2001, pp. 1–14.
[16] S. Kenny and J. Borking, “The Value of Privacy Engineering,” J. Information, Law Technol., vol. 2., no. 1, 2002.
[17] G. W. Van Blarkom, J. J. Borking, and J. G. E. Olk, Handbook of privacy and privacy-enhancing technologies. The Hague: College Bescherming Persoonsgegevens, 2003, pp. 1–4.
[18] F. L. y López, M. Luck, and M. D’Inverno, “A Normative Framework for Agent-Based Systems,” Comput. Math. Organ. Theory, vol. 12, no. 2–3, 2004, pp. 227–250.
[19] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, “Tropos: An agent-oriented software development methodology.,” Auton. Agent. Multi. Agent. Syst., vol. 8, no. 3, 2004, pp. 203–236.
[20] F. Zambonelli and A. Omicini, “Challenges and research directions in agent-oriented software engineering,” Auton. Agent. Multi. Agent. Syst., vol. 9, no. 3, pp. 253–283.
[21] S. Park and V. Sugumaran, “Designing multi-agent systems: a framework and application,” Expert Syst. Appl., vol. 28, no. 2, Feb. 2005, pp. 259–271.
[22] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, “JADE: A software framework for developing multi-agent applications. Lessons learned,” Inf. Softw. Technol., vol. 50, no. 1–2, Jan. 2008, pp. 10–21.
[23] D. Weyns, H. V. D. Parunak, and O. Shehory, “The Future of Software Engineering and Multi-Agent Systems,” Int. J. Agent-Oriented Softw. Eng., vol. 3, no. 4, 2008, pp. 1–8.
[24] M. Cossentino, N. Gaud, V. Hilaire, S. Galland, and A. Koukam, “ASPECS: an agent-oriented software process for engineering complex systems,” Auton. Agent. Multi. Agent. Syst., vol. 20, no. 2, Jun. 2009, pp. 260–304.
[25] T. Leaute and B. Faltings, “Privacy-Preserving Multi-agent Constraint Satisfaction,” in 2009 International Conference on Computational Science and Engineering, 2009, pp. 17–25.
[26] J. M. Gascueña, E. Navarro, and A. Fernández-Caballero, “Model-driven engineering techniques for the development of multi-agent systems,” Eng. Appl. Artif. Intell., vol. 25, no. 1, 2012, pp. 159–173, Feb.

 ISBN: 978-1-61208-340-7

[27] A. Cavoukian, Privacy by Design The 7 Foundational Principles Implementation and Mapping of Fair Information Practices. Office of the Information and Privacy Commissioner (2011), 2011.
[28] J. M. Such, A. Espinosa, and A. Garcia-Fornes, “A survey of privacy in multi-agent systems,” Knowl. Eng. Rev., 2012, pp. 1–31.
[29] D. Aggarwal and A. Singh, “Software Agent Reusability Mechanism at Application Level,” Glob. J. Comput. Sci. Technol., vol. 13, no. 3, 2013, pp. 8–12.
[30] H. Hayashi, S. Tokura, F. Ozaki, and M. Doi, “Background sensing control for planning agents working in the real world.,” in Agent and Multi-Agent Systems: Technologies and Applications, 2008, pp. 11–20.
[31] K. Mivule and C. Turner, “A Review of Privacy Essentials for Confidential Mobile Data Transactions,” Int. J. Comput. Sci. Mob. Comput. ICMIC13,, no. December, 2013, pp. 36–43.
[32] A. Krause and E. Horvitz, “A Utility-Theoretic Approach to Privacy in Online Services,” J. Artif. Intell. Res., vol. 39, 2010, pp. 633–662.

 ISBN: 978-1-61208-340-7