Privacy Preserving Social Mobile Applications

Venkatraman Ramakrishna, Apurva Kumar, and Sougata Mukherjea

IBM India Research Laboratory,
ISID Campus, Institutional Area, Vasant Kunj, New Delhi - 110070, India
{vramakr2,kapurva,smukherj}@in.ibm.com

Abstract. Mobile users can obtain a wide range of services by maintaining associations, and sharing location and social context, with service providers. But multiple associations are cumbersome to maintain, and sharing private information with untrusted providers is risky. Using a trusted broker to mediate interactions by managing interfaces, user identities, context, social network links, policies, and enabling cross-domain associations, results in more privacy and reduced management burden for users, as we show in this paper. We also describe the prototype implementations of two practically useful applications that require awareness of participants' location and social context: (i) targeted advertising, and (ii) social network-assisted online purchases.

Keywords: Privacy, Multi-Domain, Social Network, Policy Management, Middleware, Identity Management, Online Advertising, Online Payment.

1 Introduction

Ubiquitous data communication infrastructure enables mobile device users to access web services on the go. It is common for users to maintain long term associations with multiple service providers, such as online merchants. These providers attempt to customize their services to be relevant to a user's context, which primarily includes location and social associations. The active involvement of a user's social network often results in applications that provide benefits to a service consumer, his social network friends, and service providers. With social networks like Facebook and Google+ being always available to a connected user, richer application scenarios can be realized [22]. We refer to these applications as *social mobile applications*, and they have the following in common: (i) remote service providers, (ii) mobile users, and (iii) social networks. Take online shopping for example, in which relevance of product information and advertisements plays a huge part. Service providers can send more relevant advertisements to a mobile user if that user's context and activities are known to them. In addition, a user's social network links can be used to determine shared interests, and to send relevant ads to multiple people. Mobile shoppers will benefit from this, as will providers who get to expand their target advertisement base. A different example involves payments using mobile devices, either at Point-of-Sale terminals or to online service providers. Though multiple payment systems have been devised in recent years [1][5][7], they force the payer to rely on his personal financial

K. Zheng, M. Li, and H. Jiang (Eds.): MOBIQUITOUS 2012, LNICST 120, pp. 89–102, 2013.

accounts, which may occasionally run out of funds. It would be useful to offer the payer a backup option of obtaining money from his social network friends as part of the payment protocol. These advertising and payment examples seem straightforward on their own, yet they face common issues that must be resolved before practical applications can be realized. First is the issue of privacy. Though the mobile device must itself possess location and social *awareness*, a user may feel uncomfortable sharing such private information with untrusted service providers for fear of misuse, and also for fear of violating his social network friends' privacy [12]. To realize the payment scenario, the payer may need to know his friends' financial state and willingness to pay, which those friends would justifiably like to keep private. The second major issue is the fact that it may become cumbersome for both users and service providers to maintain long term associations in the face of change and heterogeneity. The provider's service interface may change over time, and so may a user's domain affiliations (e.g., social network account, financial account, etc.).

In this paper, we present the design of infrastructure to handle the common requirements of such social mobile applications, thereby making it easier to build and maintain them. Our system (i) protects users' privacy from each other, and from service providers, and (ii) manages changes in user identities, affiliations, and service characteristics, which would otherwise be a burden on users and service providers. The fulcrum of our system is a *broker* that mediates interactions amongst users and providers. This broker is hosted by a trustworthy infrastructure provider with a large subscriber base, like a telecom operator or Google, and is relied upon by providers and consumers to relay messages without leaking a user's private information. It is aware of the identities a user assumes with service providers and in domains like social networks and financial organizations. The broker associates a unique *mobile identity* with every user, and maintains links among social network members. It discovers and updates user context, and uses it to guide interactions and information flow. Using a third party authorization framework such as OAuth [10], it can obtain limited user identity and context information from another domain. Users can register with it through a web service interface, and providers can register their service descriptions. The broker implements generic and pluggable functions to parse and redirect messages in the course of an interaction. The broker also has a role in *policy management*. It maintains policies specified by users that govern message delivery and disclosure or manipulation of user information. Since policies governing such behavior might also be set by other domains a user belongs to, and which are involved in the interaction (e.g., social network), the broker must resolve these different sets of policies. It does so by dynamically configuring a multi-domain policy enforcement workflow based on the assumption that all participating domains manage policies using the standard policy management architecture consisting of *decision* and *enforcement points* [21]. Individual domains may keep their autonomy by revoking the broker's access to them at any time. To summarize, our original contribution in this paper is a centralized architecture that enables a large number of users with possible social network links to obtain services from each other and from untrusted providers without violating their privacy and their desired behavioral policies.

We describe motivating examples in Section 2, and present our solution architecture in Section 3. In Section 4 we describe prototype implementations of two applications, and analyze our system in Section 5. We conclude the paper with a related work survey in Section 6, and thoughts on future work in Section 7.

2 Motivating Scenarios

Described below are two idealized scenarios from a mobile user's perspective.

2.1 Social- and Location-Aware Advertising

Alice enters a *My Style* store in a shopping mall and makes purchases. At checkout, she registers herself with the store using her mobile device, which uses the *MyTel* telecom data network, and opts to receive product updates. Subsequently, she receives an option to sign up for reward points in exchange for sharing and/or recommending relevant store advertisements and product information with her Facebook friends. When Alice receives an ad on her mobile phone, she indicates the names of friends who may find it interesting and relevant. One of those friends, Bob, who happens to be in the mall premises, receives a text message (and a graphical notification, if he possesses a smart phone.) and walks toward the *My Style* store to examine its wares. Jack, who is sitting in a book store in the mall, does not receive a notification as his phone has a 'Do Not Disturb' policy applied to incoming messages. Another friend, Carol, is not in the vicinity, and therefore receives an email instead of a text message.

2.2 Social Network-Assisted Shared Payments

Alice uses her mobile phone to make a purchase at a store where she has an account, and subsequently receives the purchase details. Next, the phone prompts her to pay using one of her credit cards. Payment attempts fail because of insufficient card balances, and Alice is presented with an option to request money from her social network friends (including relatives). Some of these friends hold joint accounts with her, and withdrawals require multiple authorization. Others may be willing to loan money from their accounts to Alice's. Friends likely to agree to payment requests from Alice receive query messages if their phones are turned on, and if they possess sufficient funds to make the payment. If one or more friends see the message and agree to pay, the store receives the payment, and the purchase completes.

2.3 Discussion

The above applications involve interactions among mobile users, their social network friends, and service providers. Users have multiple affiliations, and messages are sent based on relevance and context. Similar characteristics are desirable in other settings, like (i) a museum that could serve its patrons and their friends better with more contextual knowledge, and (ii) a traffic monitoring and guidance service that could make better decisions with inputs from more commuters (i.e., by including social network friends). These scenarios can be generalized to a larger class of social mobile applications, which give mobile users access to context-sensitive services and enable them to perform useful tasks using their mobile devices. If users' identities, context, and social network associations were made publicly available to each other and to service providers, such applications could be built easily as the users and providers would have the information to make optimal decisions. Yet exposing location and social network relationships would be a violation of privacy, and subject users to undue risks [12]. A social network member with a lax privacy standard (e.g., Alice) may inadvertently compromise the privacy of a user with a stricter standard (e.g., Bob) by revealing their association to an untrusted service provider (e.g., advertiser). Such a provider may use this information for spam, or something more nefarious. Building applications with privacy protection is possible, but only if a trusted broker or mediator is employed, as we will see in the following section.

3 Social Mobile Application Middleware

We model social mobile applications using service providers, consumers, social networks, resources or accounts owned by consumers, and data communication infrastructure. Every consumer/user has a mobile device that is capable of data communication, and stores its owner's personal data, preferences, and credentials to access the domains a user belongs to. (Here we define a *domain*, or a *security domain*, as a group of computing entities that are centrally managed and which enforces desired behavior within its boundaries through policies. E.g., social network, financial organization, professional group, etc.) Providers offer web services through which other entities can interact with them. Telecom and wireless networks are used for data communication. To support these applications, we design infrastructure that will (i) mediate provider-consumer interactions, (ii) offer interfaces for users and providers to register and send messages, (iii) manage user identities and associations, (iv) maintain user context, (v) access resources across domain boundaries, and (vi) manage policies. Figure 1 illustrates the functional diagram of the infrastructure.

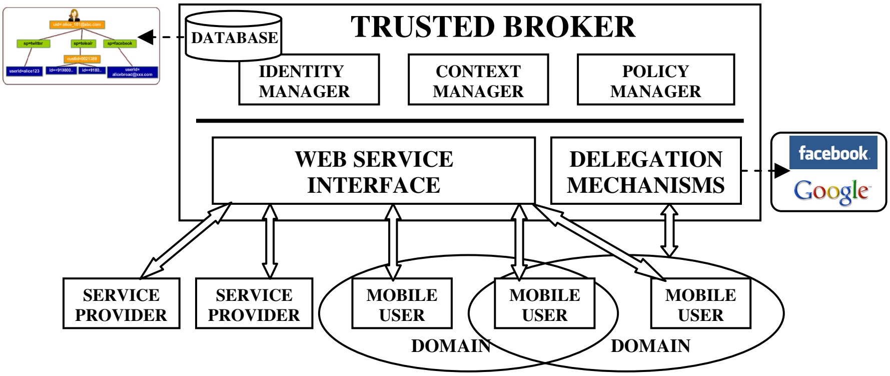


Fig. 1. Platform Architecture and Functions

Building an application that relies on the services provided by the broker involves creating client applications for mobile devices, creating web services a provider will offer, user-provider protocols, and policy rules. In some cases, new third party delegation mechanisms may be plugged into the broker. Messages among providers and users are processed and relayed by the broker, and are based on HTTP.

3.1 Interaction Mediation through a Trusted Broker

The infrastructure we envision is managed by a single entity to which multiple mobile users and service providers are subscribed. This entity mediates (or brokers) interactions amongst users, and between users and service providers, and is trusted to relay messages while maintaining integrity and confidentiality. Users trust the broker with the knowledge of their multiple identities and affiliations with various service providers and domains, and to protect their privacy. The broker itself interacts with users and providers through web services. It offers a service-based interface for users to subscribe and for providers to register their service offerings (e.g., using WSDL). Users register their affiliations with various service providers. They also register their identities as members of other domains (e.g., social network, a bank account holder, a professional organization). To realize a new application scenario, we must devise a new protocol, or a sequence of messages. These messages include service invocations (from user to provider), information requests and resource requests (user to another user or provider), or information dissemination (provider to users: e.g., ads). Every message is relayed by the broker as is, or after some semantically meaningful processing. If the message target is a group whose individuals are identified not through names but through attributes, it is the broker's function to determine the appropriate destinations and convey suitable messages to them. In the remainder of this section, we will discuss what information the broker needs to maintain in order to make suitable decisions in a given application and in a particular context.

In practice, the role of a broker can be played by a large trusted infrastructure provider, like a telecom/cellular network operator (e.g., Airtel, AT&T) or an IT services provider like Google or PayPal. Such providers may already have access to users' location context, and are reputed enough to be entrusted with other private information, especially if it makes applications easy to design and use. Telecom operators may be particularly suited to play this role in emerging markets as they are prominent entities with large subscriber volumes. Reputed cloud infrastructure providers like Amazon may also feasibly offer brokerage services.

3.2 Identity Management and Inter-domain Associations

The broker plays the role of an identity manager for mobile users. In this capacity, it maintains associations among multiple identities belonging to the same user (at multiple service providers, and in multiple domains) as well as associations among users with social network links. We refer to this unified, or federated [14], view as the *mobile identity* of the user. The mobile identity is used in conjunction with consent management protocols like OAuth [10] that provide web-based workflows for temporarily delegating privileges of a user account to the broker without explicitly sharing login credentials. Privileges could mean access to friend lists, contact information, or other private attributes. This enables the broker to act on behalf of the user and to acquire information from the user's account in another domain, such as a social network. Major social network providers like Facebook, Google and Twitter provide support for delegating privileges to third parties using OAuth.

3.3 Context Manager

The broker must be aware of a mobile user's contextual attributes for the purpose of guiding messages to appropriate entities. Users share their context with the broker and not with service providers (or even friends) because they trust the former to keep the information private. E.g., advertisements tagged with a certain location ("*in the mall*") in the scenario in Section 2.1 are relayed by the broker to only those users who are determined to be present at that location. The service provider need not know the identities or locations of the friends, but relevant ads must be sent only to the right people. Context includes location, information about a user's current activities, his organizational affiliations, his financial state, etc. Location can be detected using sensors on the mobile device or through cellular network triangulation by a telecom operator (if it plays the role of the broker). A user can set filter policies to prevent revelation of context information even to the broker based on his level of trust in it.

3.4 Multi-domain Policy Management

In our architecture, we assume that every domain enforces its policies in a centralized manner using a PAP, PIPs, PDPs, and PEPs [21]. But our scenarios involve the intersection of a user's multiple domains, which have policies framed independent of each other (e.g., the broker and the social network have different policies governing who may send messages to a user and at what times); hence a different workflow is necessary to resolve and apply the right policies. Inter-domain negotiation protocols are plausible candidates, but infeasible as they may run to arbitrary lengths [17].

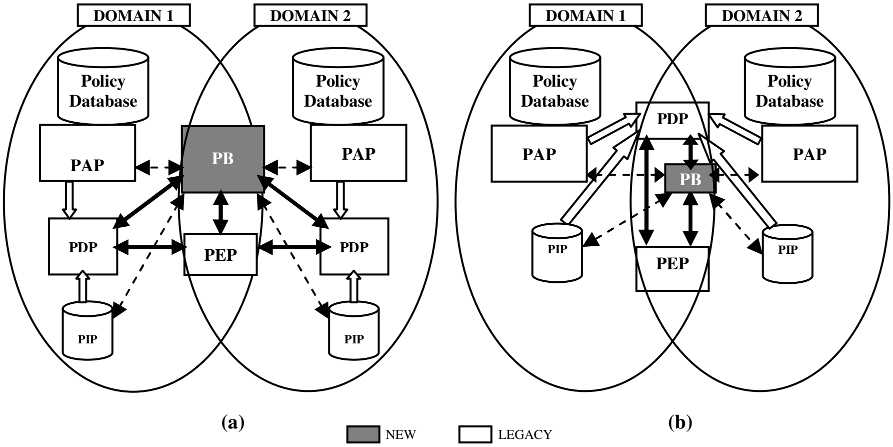


Fig. 2. Two-Domain Policy Management Dynamics using a Shared PEP and a Policy Broker

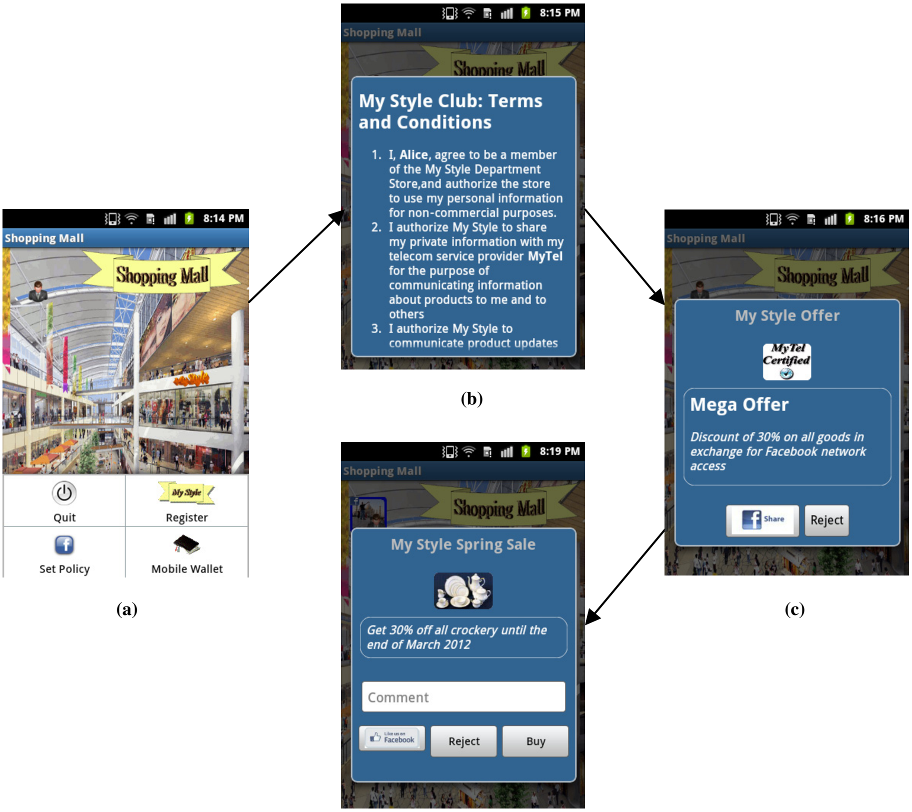
Instead, we extend the centralized policy management architecture to handle multi-domain intersections where a PEP (Policy Enforcement Point), represented by a user's mobile device, is shared by all the domains. To avoid changing the core nature of the PEP, our trusted broker also runs a module that we refer to as a **Policy Broker (PB)**; this is similar though not identical to the Policy Negotiation Point that has been proposed as a standard [9]. The PB can query the PAP and PIP modules, and mediates the PEP-PDP query-response protocol. It can work in either of two modes, illustrated in Figure 2 for a two-domain scenario. In the first mode, the PDPs of both domains are consulted independently by the PB, which obtains multiple decisions in response. The PB runs a reconciliation algorithm, which by default (but not mandated) is the Boolean AND (*conjunction*) and returns a final access decision to the PEP (Figure 2a). In the second mode, the PB selects a PDP from one of the domains and dynamically configures it to access policies and state information from the PAPs and PIPs of both domains. The PDP, having all the required information, now makes an access decision which is relayed by the PB to the PEP (Figure 2b).

4 Implementation: System and Application

We have implemented the core broker functions of identity, context and policy management, and built prototypes of the applications described in Sections 2.1 and 2.2 to demonstrate the utility and necessity of using a trusted broker. Users in our implementations were represented by Samsung Galaxy Ace S5830 phones running Android Linux v2.2 and v2.3. The broker was built as a Java application deployed on an IBM WebSphere Application Server 7.0 instance running on a SuSE Enterprise Linux 11 server. The IBM Tivoli Directory Server v6.3, which offers an LDAP User Registry and a DB2 database, was used to store user data. Services offered by providers were also implemented as Java applications exposing REST APIs that ran on similar WebSphere configurations. Mobile devices communicated with servers using WiFi. IBM Tivoli Security Policy Manager (TSPM) [3] instances were configured to protect the WebSphere applications, and used to store users' policies. For a social network, we used Facebook, as it provides an HTTP-based Graph API and delegation using OAuth, enabling us to build rapid prototypes. On the downside, Facebook does not run a policy manager based on our specification (or allows us to control its policies), thereby limiting the nature of applications we could build.

4.1 Social- and Location-Aware Advertising

An Android app enables a user representing Alice to register with a merchant, represented by a remote web service, and establish a customer account after providing identity and phone number and agreeing to the presented terms (Figures 3a and 3b). The merchant makes an offer to give reward points in exchange for allowing targeted ads to Alice's Facebook friends to the broker, which is hosted by the *MyTel* operator. *MyTel* can identify Alice by her phone number and associates it with her mobile identity (established earlier through a protocol that is out of scope of this paper). The



My Style Club: Terms and Conditions

1. I, **Alice**, agree to be a member of the My Style Department Store,and authorize the store to use my personal information for non-commercial purposes.
2. I authorize My Style to share my private information with my telecom service provider **MyTel** for the purpose of communicating information about products to me and to others
3. I authorize My Style to communicate product updates

Mega Offer

Discount of 30% on all goods in exchange for Facebook network access

Get 30% off all crockery until the end of March 2012

(a)

(b)

(c)

(d)

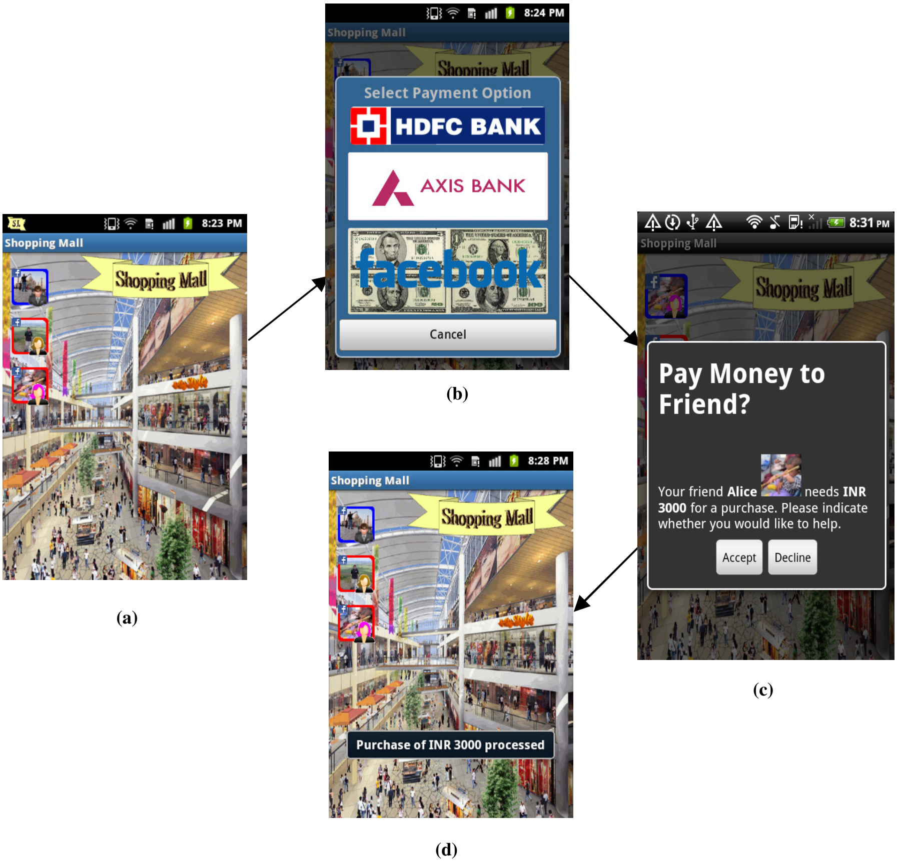
Fig. 3. Targeted Advertising Application Screenshots on an Android Device

broker, co-located with *MyTel*, relays the offer to Alice, who accepts the offer (Figure 3c). Under the covers, the OAuth protocol ensues among the client, broker and Facebook (through its Graph API) [18]. Using a Facebook app created for and hosted by the broker, a time-limited token to access Alice's friend list and post messages on her behalf is delegated to the broker after Alice signs in using her Facebook identity. The merchant is notified when this protocol completes.

Ads from the merchant are relayed by the broker to Alice's device (Figure 3d), who chooses to recommend it to her friends. The broker uses the Facebook access token to determine that Bob and Carol (who also have mobile identities) are on her friends' list. Based on its knowledge of their location contexts (which we simulated), the broker sends a text message to Bob's phone, as Bob is in the vicinity; whereas Carol, who is in a different location, gets a Facebook notification, which she sees the next time she logs into her Facebook account.

4.2 Social Network-Assisted Shared Payments

Our prototype of this scenario uses the relationship established among the parties in the advertising app. When Alice receives an ad notification, she has the option to '*Buy*' a product (Figure 4a) using a *mobile wallet* (money account) [5] she maintains with the broker. (Money can be credited to the wallet of a client from her bank accounts or credit cards using standard payment gateway protocols.) The broker is at first unable to authorize the purchase as the wallet has insufficient funds. Therefore it presents Alice with a list of payment options (Figure 4b). Alice selects the '*Facebook*' option, indicating that she wants to request her friends for money. The broker determines the list of her Facebook friends using its OAuth access token. This list is filtered through a variety of criteria. Alice may have specified a candidate list using a policy rule. The broker, using its delegated permissions, can mine Facebook information and activity to determine how close the friends are, whether they reside in the same city, etc. Knowing the mobile identities of the friends also helps the broker



Pay Money to Friend?

Your friend **Alice** needs **INR 3000** for a purchase. Please indicate whether you would like to help.

Purchase of INR 3000 processed

Fig. 4. Social Network-Assisted Payment Application Screenshots on an Android Device

determine whether they have sufficient funds in their wallets. Finally, a small (5-10) number of friends are selected, and query messages are sent to their personal devices in turn (Figure 4c). If a friend accepts, money is transferred from one wallet to another. If the requisite amount is collected, a payment is made to the merchant (Figure 4d).

4.3 Controlling Behavior through Policies

We show examples here of how different policy configurations can result in different behavior in our two scenarios. The policy rules are framed in XACML [20], but we describe them in plain English owing to lack of space.

1. In the advertising scenario, the user Bob frames a policy with the broker that has the effect of allowing advertisement notifications to be delivered to him only between 12 pm and 9 pm. Under normal operation, the ads will be displayed only if sent during that period. Subsequently, he adds a policy within his social network domain indicating that no more than 10 advertisements recommended by Alice should be displayed within a calendar day. The policy broker resolves the two policy rules, resulting in the blocking of ads recommended by Alice (even between 12 pm and 9 pm) when the count exceeds 10.
2. In the payment scenario, the default policy allows requests from one user to be sent to another user. But if Bob, as a social network member, frames a policy rule blocking payment requests from Alice, the broker will keep him off its list. In another variation, Bob frames a policy allowing any payment request of $50 or less to be automatically approved. In this case, the payment is automatically processed. Subsequently an email containing details of the payment is sent to Bob as per obligations associated with the policy.

5 Analysis

We analyze the security and privacy characteristics of our platform, using examples from the two applications. First, we want to ensure that participating entities are not susceptible to attacks by external entities. We do this by restricting all interactions to a service oriented model. Service providers, mobile users, and the broker interact through remote web service invocations. The broker itself does not allow direct access to its internal resources and offers a web service interface. Web services are *secure to the extent that they are implemented according to specification*; hence the security of the broker, service providers, or clients cannot be compromised through any process introduced by our platform. The broker's access to a user's domains is likewise service-oriented, limited in privilege, and can be revoked at any time. E.g., access to Facebook is limited by its Graph API. Getting a Facebook access token using OAuth is provably secure [10], and therefore does not create a new attack vector.

We examine our system based on two modes of privacy:

i. *Entity privacy*: entity A is considered to keep its privacy from entity B if B cannot identify A or communicate directly with A using any available mechanism.
ii. *Information privacy*: entity A keeps information X private from entity B if B cannot determine X through any communication channel with A, and cannot link X with A if it were to determine X through some other available mechanism.

We assert that these properties are maintained if the broker does not abuse the trust invested in it. In our two applications, only one mobile user (Alice) can be explicitly identified by a service provider, and this association was explicitly made by Alice. The identities of her friends remain unknown to the service provider even though they receive ads and payment requests; this is because the broker acts as a relay. Hence our system ensures *entity privacy*. *Information privacy* is also ensured by our system as follows. Bob's location is known to and used by the broker, and remains unknown to the service provider. Similarly, only the identity of the final payer is revealed to Alice, who does not get to know which of her friends rejected her request and how much money they have in their wallets. Also, allowing clients to set, enforce, and dynamically change policies within their domains allows them to keep control of their privacy irrespective of the motivations and actions of the service providers or the broker. Resolving policies on the basis of least privilege ensures that clients' wishes are respected in the most privacy-preserving manner. For example, if a client, or the social network she belongs to, chooses not to reveal friends' lists, the broker will be unable to obtain that information even if it possesses a valid access token.

The caveat is that an untrustworthy and unreliable broker could harm users by using delegated permissions to access user accounts for nefarious purposes, or collude with service providers to reveal private client information. Our entire system is based on the premise that the broker is a large and public entity that cannot escape legal bounds and obligations, and can be held accountable for any transgressions. In the future, we will attempt to increase the trustworthiness of this entity, by using frameworks like the Open ID Trust Framework [23]. We will also investigate the feasibility of distributing broker functionality among multiple agencies to limit the potential harm caused by a single centralized broker that abuses its power.

6 Related Work

To the best of our knowledge, no existing system mediates service interactions while protecting user privacy. The integration of identity, context, and policy management into a unified brokerage service that enables a range of social mobile applications is our original contribution. Research relevant to our work has focused exclusively on (i) areas like identity management or policy resolution, (ii) social network privacy [12], (iii) applications: e.g., targeted advertising, online payments, (iv) service composition.

Google is the closest approximation of our trusted broker in practice. It provides social networking services through Google+, mobile payment services using Google Wallet [7], and tracks user movements through Google Latitude. Possessing a Google account is very similar to possessing a mobile identity. Yet, though Google could plausibly play the role of a mediator, its primary aim is to be a service provider itself and provide better search results by gathering and mining data.

The MobiSoC middleware manages location and social context, and provides a development platform for mobile social applications [22]. It supports a different set of applications than our system does; also, it focuses only on interactions among users and does not consider service providers. MobiSoC enforces privacy constraints using policies, and the authors assert that a trusted centralized middleware provides better privacy guarantees than a distributed middleware. Related to this is a distributed privacy-conscious data sharing model of *personal networks* and *agents*, proposed by Connect.Me [24], though no working system has yet been produced.

Mediating access to heterogeneous web services in a *Service-Oriented Architecture* (SOA) is also a well-researched subject. Mediators can dynamically discover and compose web services [4], or match and translate data flowing from one web service to another [6]. Mashup services can be realized using a location service broker that enables service providers to access user context through lightweight APIs [13]. The Open Group's SOA specification [11] uses an integration layer to mediate interactions by relaying and routing messages. All these solutions conceive of the mediator as just a layer of indirection, whereas our design adds privacy protection, identity management, and policy management to the functions performed by the broker.

Research in the area of online advertising is relevant, but cannot be generalized to other application scenarios. The Privad system uses a mediator to convey ads from publishers to clients and to report click feedback to the publishers [8]. The mediator itself is not privy to much information about the client, and offers a higher level of privacy than our system does, but it does not handle social network associations. In the social networking world, advertising strategy is often ad hoc, such as Facebook allowing a user's activities on an external website to be displayed to his friends without their consent via the Beacon app. Needless to say, Facebook was forced to add stringent controls to Beacon following a popular outburst[1]. In contrast, our advertising app allows users to frame policies, which the broker must enforce.

Much research has been done to enable mobile financial transactions, both in academia [5] and industry. A number of solutions have sprung up in both developed and emerging markets to enable a user to make cardless payments at a PoS terminal using his mobile device. Boku [2], Airtel Money [1], and M-PESA [15] are based on mobile wallets maintained by the telecom operator who provides the data communication channel, whereas Google Wallet [7], Square [19], and Mobile Pay USA [16] rely on a user's existing bank and credit card accounts. All of these systems rely on mediation or infrastructure provided by an IT giant like Google (or Amazon or PayPal) or telecom operators like Airtel or Verizon. Yet these solutions focus only on ease of use, and do not provide privacy preservation or social awareness.

7 Conclusion and Future Work

Though increased physical and social awareness in mobile applications benefit service providers and consumers, the latter have legitimate concerns about untrusted

[1] Thoughts on Beacon, `http://blog.facebook.com/blog.php?post=7584397130`

providers misusing their private information. In this paper, we have presented a solution architecture based on a trusted broker that manages user identity, context, and policy. Interaction mediation by this broker enables mobile users to keep their privacy while interacting with other users and service providers. Relying on the broker to associate identities, manage context and policies, relay messages, and access cross-domain information using authorization delegation enables the rapid creation and deployment of a variety of social mobile applications, like targeted advertising and collaborative online payment. In the future, we intend to conduct user studies and run large scale experiments with our system. We will measure the broker's scaling properties with respect to the number of clients, and make suitable improvements. Enhancing the broker's functionality to support service discovery is another promising line of research. Also, our assumption of a single broker who is trusted by all mobile users and service providers must be tested in the real world. We may instead employ multiple brokers to serve disjoint sets of users, and configure these brokers to interoperate and establish trust relationships. We will also investigate and build more application scenarios that rely on our system. Lastly, we will explore ways to prevent the broker from abusing user trust and becoming a single point of failure.

References

1. Airtel Money, `http://airtelmoney.in`
2. Boku, `http://www.boku.com`
3. Buecker, A., et al.: Flexible Policy Management for IT Security Services Using IBM Tivoli Security Policy Manager. IBM Red Paper Publication REDP-451200 (March 17, 2009)
4. Cimpian, E., Mocan, A., Stollberg, M.: Mediation Enabled Semantic Web Services Usage. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 459–473. Springer, Heidelberg (2006)
5. Dahlberg, T., Mallat, N., Ondrus, J., Zmijewska, A.: Past, Present and Future of Mobile Payments Research: A Literature Review. Journal: Electronic Commerce Research and Applications 7(2), 165–181 (2008)
6. Fauvet, M.C., Aït-Bachir, A.: An Automaton-based Approach for Web Service Mediation. In: Proceedings of the 13th ISPE International Conference on Concurrent Engineering (ISPE CE 2006), Antibes, France, September 18-22 (2006)
7. Google Wallet, `http://www.google.com/wallet`
8. Guha, S., Cheng, B., Francis, P.: Privad: Practical Privacy in Online Advertising. In: 8th Usenix Conf. on Network Systems Design and Implementation (NSDI), Boston, MA (March 2011)
9. Haidar, D.A., Cuppens-Boulahia, N., Cuppens, F., Debar, H.: Access Negotiation within XACML Architecture. In: Proceedings of the Second Joint Conference on Security in Networks Architectures and Security of Information Systems (SARSSI), Annecy, France (June 2007)
10. Hammer-Lahav, E., et al.: The Oauth 2.0 Authorization Protocol (January 2011), `http://tools.ietf.org/pdf/draft-ietf-oauth-v2-12.pdf`
11. Integration Layer, `http://www.opengroup.org/soa/source-book/soa_refarch/integration.htm`

12. Krishnamurthy, B., Wills, C.E.: On the Leakage of Personally Identifiable Information via Online Social Networks. SIGCOMM Comput. Comm. Rev. 40(1), 112–117 (2010)
13. Loreto, S., Mecklin, T., Opsenica, M., Rissanen, H.M.: Service Broker Architecture: Location Business Case and Mashups. Comm. Mag. 47(4), 97–103 (2009)
14. Maler, R., Reed, D.: The Venn of Identity: Options and Issues in Federated Identity Management. IEEE Security and Privacy 6(2), 16–23 (2008)
15. Mas, I., Morawczynski, O.: Designing Mobile Money Services: Lessons from M-PESA. Innovations 4(2), 77–92 (2009)
16. Mobile Pay USA, http://www.mobilepayusa.com
17. Ramakrishna, V., Reiher, P., Kleinrock, L.: Distributed Policy Resolution Through Negotiation in Ubiquitous Computing Environments. In: Proceedings of IEEE PerCom 2009, Galveston, TX (March 2009)
18. Server-Side Authentication, http://developers.facebook.com/docs/authentication/server-side/
19. Square Inc. (US), https://squareup.com
20. Verma, M.: XML Security: Control Information Access with XACML, http://www.ibm.com/developerworks/xml/library/x-xacml/
21. Westerinen, A., et al.: RFC 3198: Terminology for Policy-Based Management (November 2001), http://www.ietf.org/rfc/rfc3198
22. Gupta, A., Kalra, A., Boston, D., Borcea, C.: MobiSoC: A Middleware for Mobile Social Computing Applications. Mobile Networks and Applications Journal 14(1), 35–52 (2009)
23. Open Identity Exchange, http://openidentityexchange.org/what-is-a-trust-framework
24. Conect.Me Trust Framework, https://connect.me/trust