Pers Ubiquit Comput (2007) 11:157–169
DOI 10.1007/s00779-006-0100-9

ORIGINAL ARTICLE

MobiPass: a passport for mobile business

Robert Steele · Will Tao

Received: 28 May 2006 / Accepted: 1 August 2006 / Published online: 3 November 2006

Abstract While pervasive computing provides a potentially vast business opportunity for many industry participants, it also carries challenges along with it. In this paper, a passport-based architecture has been proposed to convert this unpredictable, highly dynamic pervasive environment into a trusted business platform. It utilizes the widely accepted passport concept here named MobiPass to evaluate and classify the potential mobile entities into a trustworthy form. It allows fine-grained access control without necessarily having had prior interaction with or knowledge of other parties and environments by setting customized rules against a MobiPolicy. A detailed case study has been introduced to demonstrate how the MobiPass architecture can help customers and retailers to build a strong trusted connection and how the shopping experience can be enriched and efficiency improved.

1 Introduction

Being regarded as the third wave of the computing revolution, ubiquitous computing is on the horizon to permeate modern business and community activities. As it has been stated, ''the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it'' [1].

Ubiquitous computing envisions a world embedded with a vast number of visible or invisible computing artifacts. Ubiquitous computing is producing a profound effect on the way people use services and information, enabling new types of context aware services. Ultimately, these technologies will support a world of ubiquitous commerce [2].

Enormous business opportunities from ubiquitous computing are emerging, from adopted services such as mobile banking to emerging services such as location-based services and remote monitoring services. Ubiquitous computing provides a huge platform to allow industry participants to catch this ''wave''.

However, for ubiquitous computing to gain widespread adoption and success, certain requirements must be satisfied. One of the major concerns to deter ubiquitous commerce is that currently there is no effective approach to building a trusted environment in such a highly dynamic, unpredictable environment; in other words, there must exist a feasible mechanism to protect sensitive information when mobile entities interact with each other while still allowing the necessary information to be exchanged for useful mobile interaction, so as to allow the success of ubiquitous business [3, 4].

As ubiquitous computing is based on a massive networked environment with a large population of diverse smart mobile entities, it poses a new challenge from traditional computing. It is hard to know in advance which entities will be interacted with and a request can come from unknown environments or entities where holistic information is not available [5,

R. Steele (✉) · W. Tao
Faculty of Information Technology,
University of Technology, Sydney,
P.O. Box 123, Broadway, NSW, Australia 2007
e-mail: rsteele@it.uts.edu.au

W. Tao
e-mail: wtao@it.uts.edu.au

6]. In this decentralized infrastructure, the mobile entity might have to make autonomous decisions with only limited information available. All these aspects introduce the new issue which is trustworthiness in ubiquitous business computing. This issue is regarded as the greatest barrier which may stop ubiquitous computing success in the longer-term [7–9]. Previous studies show that trust plays a critical role in customer relations [10] and the importance of trust has also been examined for eCommerce [11, 12] and mCommerce [13, 14].

In this paper, we propose an architecture named MobiPass to build a trusted and flexible environment for ubiquitous business computing. Our architecture aims to allow transaction entities to create trusted interaction without necessarily having prior knowledge of each other. By employing our architecture, mobile entities can interact safely with each other by enabling pre-set customized preferences. Our architecture adopts a user centric philosophy that delegates the final decision to the user, still with reasonable flexibility and extensibility. In our architecture, the mobile entity only talks with another trusted entity/environment which satisfies this customized access control rules.

This paper is structured as follows. Section 2 describes the overall architecture of MobiPass and interactions between different elements for establishing the trusted environment in mobile business computing. In Sect. 3, a representative case study is used to demonstrate our architecture. Section 4 discusses related work and Sect. 5 concludes the paper.

2 MobiPass architecture

2.1 Motivation

As we have emphasized, to pursue success in ubiquitous/mobile business, a trusted environment must be established via a practical approach. As mobile computing is a user-centric business model, the approach for creating a trusted network must be straightforward and easily adoptable by end users and it must provide fine-grained access control with an easily operable user interface. For example, complex security protocols should not be exposed directly to the end user. In our architecture, we have considered both technical factors and human factors and utilized the well-known passport concept combined with a preference wizard to allow users to easily set their customized rules to decide which service they want to use and which transaction entities they want to interact with in a flexible and understandable way.

The most significant difference in ubiquitous computing from traditional mainframe and personal computing is that the environment and network is unpredictable and keeps changing. The biggest challenge is that the mobile entity does not know which entity is trustworthy, including previously un-encountered entities. However, here, we claim that the mobile user usually will have a rough idea about which services they want to use and which group of entities they would like to interact with. So here we use the well known and proven passport concept, here migrated to a mechanism to allow dynamic authentication and authorization, to convert this unpredictability into a predictable, trusted form.

The new architecture is based on extending digital certificate technologies.

2.2 Overview of the MobiPass architecture

The infrastructure of the architecture utilizes a number of existing technologies such as digital certificates, certificate authorities (CAs) and asymmetric key encryption. There is some virtue in reusing existing technology building blocks in the infrastructure of mobile computing, as a large number of devices are already in the field. If the architecture is to build on top of existing and well-proven technologies, it can be easily adopted and implemented.

The new elements introduced in this architecture to enable the specific MobiPass functionality are as follows:

- Central Registry (not mandatory)
- MobiPolicy
- Extended Certificate Authority (ECA)
- MobiPass
- MobiManager

All the new elements are shown in Fig. 1, and the following sections discuss how these elements interoperate to create a flexible and trusted mobile business platform. The architecture will be addressed as follows. Firstly, we generally describe the elements in the architecture respectively, and then we explain how these elements can establish the trusted mobile environment, i.e., how these elements interact with each other.

2.2.1 Central registry

The central registry (CR) is a global, trusted service registry in our architecture. The purpose of introducing the CR into our architecture is to provide a solid base

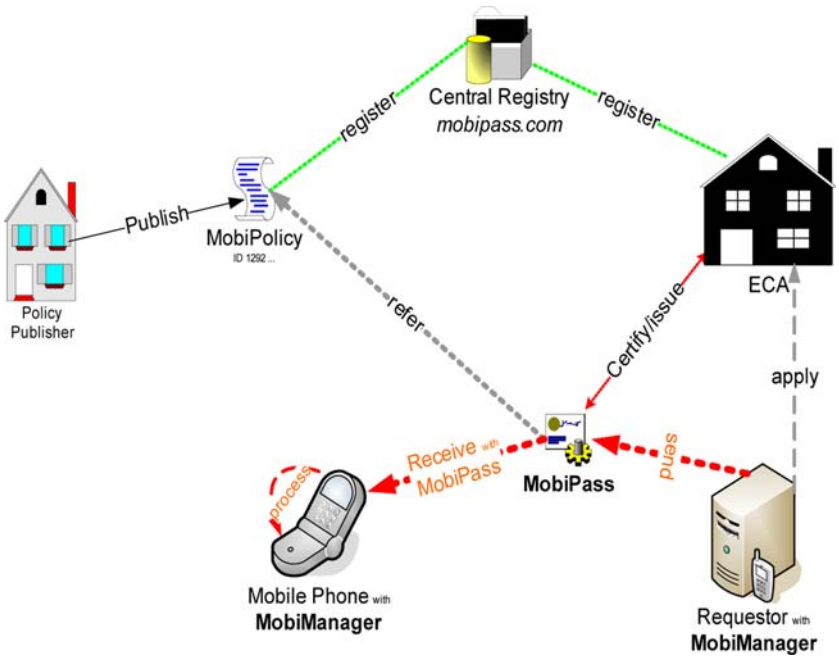

Fig. 1 The overall MobiPass architecture

reference for our trusted infrastructure, although it is not mandatory in the architecture.

The CR has responsibility for several issues:

- Accrediting of an ECA, assigning the unique ID to an ECA (ECA-ID), and providing the interface to allow a mobile entity to pull the ECA's public key, and check the relevant details of an ECA.
- Allowing organizations to register their MobiPolicies, assign the unique ID to the MobiPolicy (Policy-ID), and provide the interface to allow a mobile entity to pull the schema (description) of the MobiPolicy.

The detailed interaction between the CR and mobile entities will be discussed shortly. We recognize that building a trusted environment in ubiquitous/mobile computing requires more than an enabling technical infrastructure alone. Law, legislation and social norms [7, 15] are also needed to establish a working trustworthy system. For example, the operation of the CR will be administered by a trusted third party, e.g., a government authority. For practical reasons, we choose a geographical domain-based architecture to implement this CR. This means the word central in here is a logical concept, and the physical architecture of the CR is decentralized and self-regulated. It uses a URI to indicate/represent trustworthiness. For instance, we can say that the root trusted URI for the CR is mobipass.org, so all ECAs which operate in Paramatta, Australia can, e.g., be listed at city.state.country.mobipass.org and managed by, e.g., the ''Local City Mobile Service Authority''.

2.2.2 *MobiPolicy*

MobiPolicy allows a flexible and extensible approach to describing the particular service and evaluating mobile entities based on their backgrounds and other related information for this particular service. It is a service oriented description schema which is used to depict how and which mobile entities will interact with each other for this service. Different services will have different MobiPolicies and which information the MobiPolicy needs to describe it is entirely based on the characteristic of the service.

An XML Schema is used to describe the detailed data structure of each MobiPolicy, and each MobiPolicy will have a globally unique ID, a service describing section and the mobile entity (service description) describing section. It can either be assigned with a globally unique ID by the CR or be self-assigned with an ID in a certain range. Due to the complexity of real business services and the differences between service providers, XML schema is a good candidate for constraining the data structure.

The awareness/scope of MobiPolicies is very scalable. This means the policies can be published by, e.g., international standard bodies and recognized globally, or be published by an individual person and shared with friends. The reason to allow flexibility and scalability with MobiPolicies is that ubiquitous computing can be used for multitudinous purposes and industries—it is almost impossible to finalize with a rigid framework how to use all ubiquitous services. We

recognize this diversity and try to make our architecture open and extensible enough.

A MobiPolicy uses customized criteria to enable it to produce a fine-grained evaluation result. For instance, a MobiPolicy published by the National Accommodation Association, will have one section for describing the attributes for real estate agents, and another sections for describing properties such as their type, the number of bed rooms, location etc. For an example ''Friend-Finder'' service, the service provider can publish a MobiPolicy which tries to describe the person who subscribes to the service. For example, the name, gender, nationality and some other service related attributes could be described in the Friend-Finder MobiPolicy. It is important to note that a MobiPolicy to govern mobile interactions with an individual could equally well not contain personally identifying information, e.g., only their shopping interests or other preferences, enabling *privacy maintaining* mobile interactions.

In theory, any organization (person) can publish their MobiPolicies, there is no technical barrier to deter the organization (person) to publish their own MobiPolicy. For instance, either the Australian Taxation Office (ATO) or Paramatta Football Club can publish their own specific MobiPolicy to describe their service and which mobile entities will be involved in their services. This is absolutely acceptable and feasible in the technical sense. The reason to enable this is to meet the maximum flexibility and scalability in our architecture.

In terms of practical concerns (related to end user acceptance), although there is no limitation about the publishing ability of MobiPolicies, it is surmised that usually market forces would push the similar services to be merged and form a more general and commonly usable MobiPolicy, shared in by a larger user set/ market.

The Policy-ID is very important in our architecture because it allows for the reuse, recognizing and managing of the MobiPolicy. It is a globally unique 32 bit digital number. The Policy-ID range has two blocks, which are the reserved block and self-using block. IDs with the first digit a 0 belong to the reserved block and the remainder of IDs are in the self-using block (Fig. 2). The Policy-ID in the reserved range can be only assigned by the CR. If a MobiPolicy is created by someone who does not want it to be registered in the CR (e.g., a MobiPolicy shared with a few friends), a self-using block is available. Generally, random 32-bit numbers are chosen to work as the Policy-ID. As the set of available numbers is relatively large, it mitigates against, (but not totally eliminates) the collision of the self-using Policy-IDs. Non-collision is assured for Policy-IDs from the reserved block.

Policy-ID (32 bit digitals)	
Reversed Block (issued by CR)	[0]…. …. …. …. (31 bits)
Self-Using Block	[1]…. … …. ….. (31 bits)

Fig. 2 Policy ID allocation

The purpose of the Policy-ID is that it allows the user (or his/her MobiManager) to recognize which service an incoming MobiPass is related to. By checking the Policy-ID, it enables the MobiManager to recognize and discard and/ or process all different types of MobiPasses.

By having this specific standard for incoming information to a mobile user, describing the unpredicted environment and surrounding entities, if this information can *also be authoritatively certified*, the mobile entity can trust other entities which present the certified evaluation results, even a previously unseen mobile entity, based on the particular required characteristics of the entity that have been certified. For example, by employing the standard for describing a furniture shop, when the mobile entity receives the message formatted in accordance with a MobiPolicy for a shop-finder service, it can easily have a quickly attained and trusted idea of the seller and its business. Due to the pre-defined standard that a MobiPolicy represents, requestees will be able to design customized rules in advance through their knowledge of the criteria provided by a MobiPolicy.

2.2.3 Extended certificate authority

As the name suggests, the ECA is a functionally extended CA. Beyond issuing the digital certificate, it can also evaluate the mobile entities according to the MobiPolicy, and *digitally sign the results of the evaluation*.

In our architecture, the ECA is a trusted third party. Trust is (partially) delegated to the ECA, and the *output produced by the ECA is the MobiPass*. The ECA is the connector which links the potentially communicating but untrusted entities into a scalable and safe network.

The ECA can be the MobiPolicy publisher or the third party who is asked to certify the related business entities and activities. It can vary according to the degree of trustworthiness needed in the service. Or in some scenarios, the ECA might not have the real authority to evaluate mobile entities and all relevant

documents for evaluating may still go through the government authority/ agency to be certified/ evaluated, then mobile entities would bring these hard copies to the ECA. After checking these ''stamped'' documents, the ECA represents the evaluation results in signed electronic form according to the MobiPolicy. The digitally signed output using the ECA's private key is the MobiPass.

As we discussed, it is hard to build a trusted environment based only on technological features. For practical concerns, in here, we assume ECAs can be categorized into three main types: (in the real case, maybe there is also the potential for a different approach to arranging ECAs)

- Accredited ECA
- Registered ECA
- UnRegistered ECA

All ECAs are supposed to have a unique ECA-ID. An ECA-ID has a similar structure with the Policy-ID; 32 bits long with reserved and self-using blocks. The accredited ECA usually are highly trusted third parties, such as the Justice Department and the Fair Trading Association, and are registered with the CR. Accredited ECAs perform the similar duty as they do for current real businesses, such as issuing the permission to trade, suspending a license if a rule is breached etc. In our architecture, the MobiPass issued by an Accredited ECA should be fully trusted. Following the structure of the CR, the accredited ECA is structured according to a geographical domain-based architecture, and the number of accredited ECAs is strictly limited/co-coordinated (this helps to enforce non-competing/non-overlapping MobiPolicies for given domains in given geographical areas).

The Registered ECA means the ECA itself has been examined by the CR, and certain business information has been verified, such as business name, business registration date and business description. Both accredited and registered ECAs will be assigned a unique number from the reserved block, and the public key of the accredited and registered ECA can be retrieved from the CR by specifying the ECA ID. The ECA's business details can also be queried by providing the ECA-ID to the CR. A non-registered ECA can only pick a number from the self-using block, and the public key cannot be stored in the CR.

2.2.4 *MobiPass*

A MobiPass is described by XML which complies with the corresponding XML schema represented MobiPolicy. It contains the real data describing a particular mobile entity in relation to a certain service. For instance, if the MobiPolicy is used to describe which business entities will use a location-based service to promote their products, the data in the MobiPass can provide a signed description of a particular business entity such as the registration time of the business entity and the self-description of the business entity

Here is the general structure of the MobiPass, all contents in the certified elements need to be certified by the ECA such as the business profile and the mobile entity (service provider)'s public key, and non-certified content can be put into the non-Certified elements as extra description such as the price they sell the particular model of sofa at, at that particular time.

There are five dedicated elements in the MobiPass to ensure trustworthiness. The structure is shown in the Fig. 3.

- <eca/> the ECA element has two sub elements, which are eca and ecaLocation, which contains the ID of the ECA(ECA-ID) and the URI from which you can obtain the ECA's public key. The protocol to retrieve the public key of the ECA is included as well. In here, cr:// means the public key resides on the CR. Also, for a non-registered ECA, it is possible to get the public key by short-range wireless, such as bt://, Bluetooth. The protocol for retrieving should be very flexible and the no-Internet access situation is fully supported.
- <mobiPolicy/> the MobiPolicy has a similar structure with the <eca/> element, with ID and the location to retrieve the schema. Also, the protocol to retrieve the schema is flexible.
- <certifiedDigest/> this element is used to store the digitally signed digest value by the ECA. It is used for checking the integrity/ non-corruption of the content in the certified MobiPass element, such as the public key of the service provider.
- <mobiPassDigest/> to ensure the authentication, a <mobiPassDigest/>: this element is used for storing the whole mobiPass digest value. And it is digitally signed by the service provider itself. This means <certifiedDigest/> element is used for ensuring the public key of the service provider is true, and the <mobiPassDigest/> element is used for ensuring the MobiPass is sent by this particular service provider.
- <timestamp/> the timestamp element is used to indicate the time frame the MobiPass lives. And the value of <notBefore/> and <notAfter> will be signed by the sender's public key. This is used to prevent man-in-middle attack. The difference value

Fig. 3 MobiPass format

```
<mobiPass>
  <eca>
    <ecaID>SYDAU102...23474</ecaID>
    <location>cr://paramatta.nsw.au.mobipass.com</location>
  </eca>
  <mobiPolicy>
    <policyID>1223-3328-...24</policyID>
    <location>paramatta.nsw.au.mobipass.com</location>
  </mobiPolicy>
  <expireAt>1124030906734</expireAt>
  ...
  <certified>
    <businessName>UTD Furniture Sydney</businessName>
    ......
    ......
    <publicKey>mQENBEJvo....</publicKey>
  </certified>
  <nonCertified>.....</nonCertified>
  <certifiedDigest>f5f7f7...8a53</certifiedDigest>
  <mobiPassDigest>6fh7f1...cda84</mobiPassDigest>
  <timestamp>
    <notBefore>1132622517640</notBefore>
    <notAfter>1132622519640</notAfter>
    <signature>skz...==z</signature>
  </timestamp>
</mobiPass>
```

of notAfter and notBefore will be fairly small to get maximum security.

Here it must be noticed that if the first digit of the ECA-ID is 0, then the location (URI) of the ECA must be in the sub-domain of the CR (*mobipass.org in our case)*, and the same for the MobiPolicy. Furthermore, if the Policy-ID is registered (0 for the first digit), then it can only be verified by the accredited ECA.

MobiPass can be regarded as a special form of digital certificate which carries necessary and context aware information for mobile entities. As such it does not just provide authentication as a normal certificate does but also contains information to drive authorization decisions.

From Fig. 1, we can see that all mobile entities are virtually connected through the ECA by referring to the MobiPolicy in the ubiquitous computing environment. The MobiPass expires after a certain time depending on the functional and nonfunctional requirements, and how sensitive the data is inside the MobiPass, and it is also renewable. Mobile entities can renew their MobiPass at any time to prevent their expiration and increase their degree of trustworthiness.

2.2.5 MobiManager

MobiManager is a software package which has all the necessary functionalities to manage and process MobiPasses in our architecture. For instance, the MobiManager will parse the MobiPass, can contact the CR and automatically generate the preference selection interface for a MobiPass by referring to the schema in the MobiPolicy and so on.

2.3 A trusted interaction using MobiPass

In a highly dynamic and unpredictable ubiquitous computing environment, it is neither efficient nor fea-

sible to use username and password combinations as an approach to build a trusted and stable mobile business environment, due to the vast number of interactions possible and the frequent encountering of previously unseen entities. Also, from the usability perspective, if the mobile entity is a mobile phone, it is difficult to require of the user to type in username and password combinations in such small devices [16]. In our approach, the simple preference setup is sufficient for mobile entities to protect and prove themselves. Mobile entities only need to pre-define the customized rules (via preference) by referring to the MobiPolicy and using these rules they can manage and build the trusted relationship with potentially a large set of previously unknown mobile entities automatically.

In this section, we will provide an in-depth explanation of the work processes in the MobiPass architecture, and in the case study in the following section, more concretely grounded examples will be discussed.

Before interaction with the MobiPass, the user (device owner) can set a primer mode as to how to deal with MobiPasses/ECAs. For example, in the optimistic mode, any ECA will be trusted automatically without prompt. And in the pessimistic mode, if the ECA-ID/public key is not found already loaded in the MobiManager, a prompt dialogue will be used to ask the user to make a decision, even if the ECA is an accredited ECA. The default mode is that all MobiPass issued by an accredited ECA can be trusted automatically, and the dialogue will only appear if the ECA-ID is not found and the ECA is not an accredited ECA, also. For non-registered ECA, a ''highly dangerous'' alert will be presented.

MobiPass is the key enabler to apply these rules as it carries within it certified data which is highly related with the service and the requestor. The key point here is that once the device has set customized rules via preferences, it can ''smartly'' decide using trusted information which interaction should be allowed by examining the MobiPass. In our architecture, we call this *push mode*, because the requestor pushes the MobiPass to the requestee to have access. The requestor can also use the *pull mode* to ask for a requestee's MobiPass. This means that not only can the requestee check the requestor's MobiPass to decide whether or not to conduct an interaction with requestor or not, but also the requestor can choose who it wishes to communicate with by checking the requestee's MobiPass. This model has several advantages. Firstly, it can decrease the risk on the requestor's side. Secondly, by employing and pre-checking the MobiPass, it provides the chance to provide smarter context aware services because it might contain more detailed information the device owner wants to share. This usually can improve the quality of the service.

The full interaction of a mobile device upon receiving a MobiPass can be described as five main steps, which are:

- Validity checking and retrieving the public key of the ECA
- Verifying the MobiPass
- Enabling the MobiPolicy
- Setting the rules
- Applying the rules

If the mobile service requestor decides to interact with a particular requestee, the service requestor will send its MobiPass to the requestee as Fig. 1 indicates. If the MobiPass is not ignored by the requstee, then the first step (verify the ECA) will be activated.

2.3.1 *Validity checking and retrieve the public key of the ECA*

If the requestee receives the MobiPass, it will retrieve the public key of the ECA. Firstly, after getting the ECA-ID with the location of its public key, and Policy-ID with the policy location, the MobiManager will apply the general checks at first. That is, whether the ECA-ID is in the reserved block (assigned by the CR) or whether it is in the self-using block. If it is in the reserved block then the location (URI) must be a subdomain of mobipass.com. And if the Policy-ID is in the reserved block the schema has to be in the trusted URI as well. If the MobiPass breaches the general rules it is considered not valid and it will be rejected immediately.

After passing the general rules check, the MobiManager examines whether this ECA-ID has already been stored in the device. If the ECA-ID is found in the MobiManager, it means the ECA's public key can be used directly. Otherwise the key has to be retrieved remotely.

If the public key is not on the trusted URI (belongs to the mobipass.com in this case) or the ECA-ID is from an unknown party, MobiManager will suspend the processing and prompt the user that this is the first time to communicate with this ECA and whether to proceed or not. In addition it will indicate whether this ECA is accredited or not, with business name and description. If the ECA is not accredited, a high-risk symbol should be displayed. After a short period of use, the MobiManager should have stored the public keys of the common ECAs in operation in the area.

Whether this ECA is trusted or not depends on how the user decides upon it. As an accredited ECA must

be verified by the CR, the business name of the accredited ECA cannot be forged. If the business name of the ECA is the Fair Trading Association in Australia, which belongs to a government department, it would be readily trusted by users. If the accredited ECA is a small, faceless company, although it has been certified by the CR with its business name, it may still be suspected by the users. If this user does not trust this ECA, the MobiManager can also help the user to block this ECA this once, at certain times or always.

If the user chooses to proceed and trust the ECA, MobiManager will retrieve the public key associated with the ECA-ID, stored in the MobiManager. Otherwise, the MobiPass will be discarded and this ECA might be banned for some period. This depends on how the user has set the primer preferences.

2.3.2 Verifying the MobiPass

After getting the public key of a trusted ECA, the next step is to verify the MobiPass. In this paper, we assume that MD5 is used for generating the digest. The MobiManager will firstly check whether the MobiPass is expired or not by examining the <expireAt> element, if the MobiPass is expired, it will be discarded immediately. If the MobiPass is not expired, the MobiManager will use the ECA's public key to decrypt the certifiedDigest element, get the original digest value, rehashing the certified element again to compare whether the MobiPass is tampered with or not. If the fingerprint is not a match, the MobiManager will ignore and discard this MobiPass immediately, otherwise, the public key of the requestor will be extracted from the certified element, and used to compare the digest value of the whole MobiPass, if the digest value is matched, it means that the MobiPass is originally sent by the requestor, and sent in the reasonably immediate past. At this point the rules corresponding to this particular MobiPolicy/ service will determine what happens next.

2.3.3 Enabling the MobiPolicy and setting the rules

If the digest value is accepted, now the MobiManager will pull the schema and generate the preference automatically.

When the ECA is trusted, the MobiPass will look up the Policy-ID in the MobiPass. If the Policy-ID has been found already in the MobiManager, it means that the preference for this particular MobiPolicy has been already set, so the MobiManager will process the MobiPass directly to see whether the requestee should get access or not. This will be done *without needing to interact with the mobile device user*.

If the Policy-ID is not found in the PolicyManager, then this is a previously unseen service. After retrieving the MobiPolicy schema, the interactive interface will be displayed to ask the user to enter their preferences. As XML schema is used to describe the specific service, our previous research work Xplorer can automatically generate the service specific interface by analyzing the MobiPolicy schema, and the preference interface is also relatively easy to be understood by the user [17]. After setting up the preference, it means the rules have been already created in this understandable and user-friendly interactive way and the Policy ID will be stored in the MobiManager. Alternatively, the user can drop the MobiPass into the <To-Do> folder in the MobiManager, and set the preferences later or from another device, such as a desktop computer.

2.3.4 Applying the rules

If rules have been created for this particular MobiPolicy, this means the MobiPass which is associated with the same Policy ID will be processed directly. The typical access control will be 'yes' or 'no', but in the complex case, Role-Based Access Control (RBAC) can be employed to decrease the complexity of the authorization issues.

To build a trusted environment in ubiquitous computing, mobile requestors are supposed to be recognized by just presenting their MobiPass. In our architecture, we do not encourage users to hide or simply turn off some functionalities due to worry about privacy or security concerns. Our architecture provides a considered balance with trusted interaction and richer services.

3 Case study

In this section, we use a representative and complex case study to demonstrate how to use the MobiPass architecture to enhance the retailing industry, so as to enrich the shopping experience and provide new capabilities in all stages of the retailing process, i.e., how to use the MobiPass architecture to achieve ubiquitous retailing. By employing the MobiPass architecture in the retailing industry, retailers and shoppers, can establish a trustworthy connection for purchasing and information gathering, regardless of whether they have prior knowledge of each other or

not. Information gathering can be more intelligent and accurate and it enlarges the range of the customer's options and helps the retailer to filter to more likely genuine potential buyers.

3.1 Mobile retailing background

Retailing means selling quantities of goods to the ultimate customer directly. It consists of the sale of goods for personal consumption either from a fixed location such as a department store or kiosk, or unfixed location and the related subordinated services. With the rapid development of mobile computing, hand-held computers like the mobile phone and personal digital assistant (PDA) have been getting used for such tasks as voice communication, text messaging, scheduling, calendar management and more complicated shopping aid services. There are some retailers who have already adopted mobile retailing to improve their business. For example, the Broadway shopping center in Sydney, Australia has employed short-range communication technology to guide the customers who have Bluetooth enabled mobile phones [18]. MyGrocer is a second generation pervasive retailing system which provides new solutions for home inventory replenishment [19, 20]. Easi-Order is a PDA application which is used to create and send a shopping list to the grocery remotely [21]. KleverKart [22] is an interesting on-cart device to inform the customer of in-store information such as which products are on sale, some detailed information for certain products and so on. While these products/architectures are mainly focused on how to help customers purchase goods, in this paper, we demonstrate how the MobiPass architecture can go beyond this to help and to develop new retailing capabilities for potential customers in *all the stages of retailing* including in the stage of forming customer product and need awareness, product information gathering by customers and in comparing alternatives as well as in purchasing.

3.2 Case study

It is very important to associate and acquaint customers with potential products that relate to their needs, as it is the entry point to trigger the potential customer's motivation to purchase and the first stage of the retailing process. In the first stage of retailing, MobiPass can be used to achieve an intelligent and trusted approach to guide potential customers, find out their specific interests, and to help shoppers to recognize their needs in an effective manner.

This case study assumes a dedicated MobiPolicy for the retailing industry will be published and both retailers and shoppers will provide relevant information to apply for their MobiPasses from the ECA according to that MobiPolicy. The ECA, e.g., can be a Fair Trading Department in the local government. The shoppers will have mobile phones with MobiPass support (MobiManager), and the retailers will also have a MobiPass enabled device with can be used to communicate with shopper's MobiManager and perform all necessary functions.

The basic principle to enhance awareness is that by using the MobiPass architecture it is possible to help the customer entity to selectively receive incoming guiding information such as an advertisement, and it is also possible for the sender (retailer) to also have the option to choose the target buyer precisely and reliably. This solves the two main issues in current retail promotion, which are spam (receivers, mostly shoppers, do not have the choice to reject unsolicited and non-relevant information) and untargeted broadcast (seller does not know who should most relevantly get the information).

The way that MobiPass helps the potential customers in the awareness stage of retailing can be considered under two main approaches, which are:

- *The push* approach
- *The pull* approach

The push approach is based on the traditional advertisement principle, i.e., the seller's piece of information is broadcast by the sender to get maximum effect. There are a number of ubiquitous advertising research projects that have been conducted recently, e.g., push-based location-aware mobile advertising systems. However, these ubiquitous advertisement systems do not ensure the sender/source of the information (advertisements) is trustworthy and moreover, they do not provide a practical approach to filter the unwanted information (spam) in current mobile/ubiquitous advertising. By employing the MobiPass architecture, these issues can be solved in an efficient, flexible and reliable manner.

As mentioned previously, we assume both retailers and customers have adopted the MobiPass architecture and have the enabled equipment ready. For instance, a supermarket will have a MobiPass enabled device which connects to its intranet, which in this more realistic and extended case study might include an internal database server, sensor network and application server. As the database server has all inventory information such as SKU, number of each item in stock and some ordering information, and it

is updated in real time as transactions at the front counters occur. The supermarket also has a number of sensors to collect data about, e.g., easy-to-expire products. For instance, the meat sensors will collect data about pork/beef products to see how fresh they are regularly. If some meat products are expired, it will be reported to the application server. The role of the application server, and a fundamental contribution to retailing considered in conjunction with the MobiPass architecture, is to calculate and communicate effectively and provably to customers the most suitable price in real-time based on the inventory data from the database server, customer numbers and product turnover, the data collected by the sensors and some additional conditions such as weather and the previous sales record. As now the supermarket is able to calculate the ''best'' price in real time, the problem is how to attract the potential buyer to buy these ''real time special'' products. For some service-based retailers such as a hair salon, it would be much better if they are able to inform potential customers of real time special prices based on their current busy-ness, i.e., if there are only a few customers in the shop, the nail beauty shop can decrease the price significantly to reach the maximum profit/ utilization of staff. And the problem the hair salon is facing is how to inform the customers via an effective approach, as usually not every potential customer would like to enter the shop to query whether it has a special price if they do not have some urgent requirement. For these two examples we can see that actually the retailing industry is a very dynamic industry as the context is continually changing in real time and this could heavily affect some important factors such as price. To get the retailing service efficiency improved, it is quite useful and important to broadcast certain dynamic information, and the information sent should be trusted by the receiving party (buyers).

Firstly retailers can check the relevant MobiPolicy and apply the relevant MobiPass. The MobiPass for the retailers usually contains real information about retailers such as business name, registration date and location, and it also contains rating information which is given by some trusted third party. After applying for and receiving their MobiPass, the retailers can use their MobiPass enabled devices to broadcast their context-based advertising information, as the MobiPass is a trusted source to prove the identify of the business with rich information, it provides a secure approach for which the receiver (customer) can totally trust the information they receive. This can ensure that the customer can:

- Know that the information they received is from some real business entities, and the received information is primarily trustworthy. If the customers find that they received false information in some of the detailed information, they can report to the relevant MobiPass issuer.
- Get the basic idea of the sender (retailer)'s background. For example, they can know this message is from a well-known retailer or from some retailer they have never heard of. Furthermore, they also can know some detailed information about this retailer—information such as the number of the employees and the registration date etc.
- Be able to create a filter, to-filter unwanted messages. For instance, they might only receive information about music CDs and never receive the information for the retailer whose credit rating is lower than 6 out of 10.

By using MobiPass, we can see that retailers can build an effective approach to advertising their products that embeds their business information. This greatly builds the confidence for their potential customers to make the mobile adverting trend more workable and practical.

On the other side, the MobiPass architecture also enables the pull approach for shopping navigation. The principle is based on the bid system as the roaming potential customer can set some criteria in their MobiManager, e.g., price, and broadcast this information to the retailers he/she passes by, with the MobiPass, and each potential seller can decide whether to sell the product to this buyer based on the received criteria and the corresponding MobiPass.

For the criteria-based pull approach, it is based on how the potential customer sets the criteria to look up the desired products by criteria. The criteria can be:

- The criteria of the products, such as product name, model, price or even some keywords for this product.
- The background of the retailer, who provides the selling of this product, apparently that the customers are willing to buy the products for a reliable, good reputation retailer.

For example, if a potential customer wants to buy a digital camera with the lowest price she/he wants to pay, this potential customer can firstly set up the criteria in his/her MobiManager and the sample data can be as given in Fig. 4.

Here, we can see that the potential customer has set the criteria for the digital camera he/she wants to buy, and we also can see that this potential customer only

wants to buy a digital camera from a medium or large size retailer with good customer support. The potential buyer can activate this pull information in his/her MobiManager, and keep sending these criteria to the MobiPass enabled retailer with MobiPass by some short-range communication protocols such as BlueTooth. After a retailer receives the MobiPass with these criteria, the retailer will parse the MobiPass to see whether this is a valid MobiPass. If it is an invalid MobiPass, the transaction will be terminated immediately, otherwise, the retailer will analyze these criteria, with the certified data inside the MobiPass, to decide whether to offer this particular item to this potential customer. The model provides several significant advantages for the retailing industry.

Firstly, in terms of convenience to potential customers, they can customize very fine-grained criteria to search for the product they want in a spatial area and they can easily avoid irrelevant, annoying advertisements. In addition, it greatly increases the efficiency of goods browsing, as potential customers can search all the shops he/she passes without any extra effort and even does not require the potential customer to enter the shop.

Secondly, the retailer can also pick the customer who has good credit rating (reflected by customer's MobiPass) with whom to conduct the transaction. Furthermore, the retailer can use the certified information in the customer's MobiPass to make real-time decisions about what product or at what price it should be offered to a particular customer. For example, different customer segment groups could automatically receive different price offers, e.g., corporate buyers, government buyers or individual consumers. In addition, certified information about their membership of certain retail groups, e.g., membership of an airline frequent flyer club, could trigger the calculation and offering of a special price. The MobiPass architecture allows many and complex customer attributes to be automatically and reliably communicated to retailers and this enables more complex real-time price setting, in this case based on *customer attributes* (as distinct from the previously discussed retailer attributes).

Thirdly, by collecting this criteria information sent by potential customers, retailers can more easily figure out the actual customer's requirements and change their goods structure to improve their business.

In this case study, we can see that by applying the MobiPass architecture in the retailing industry, it is very easy to help customers and retailers to build a trusted connection and unwanted information can be effectively controlled by presetting criteria. The MobiPass architecture does not only help the potential customers to find out what they need to purchase, it also assists the customer to compare different products and retailers.

```
Brand: {Canon|Nikon|FujiFilm}
Weight: 5 -7 oz
width: 2-3 in
height:1.2-1.5 in
deep: 2.2-2.6 in
Sensor resolution: 3.2 -6 mega pixels
Digital zoom: 4X - 8X
Optical sensor type: CCD
Special effects: (Indoor , Foliage , Fireworks , Land-
scape, Portrait mode, Night snapshot}
Max shutter speed: 1/2500 - 1/2000 second
Min shutter speed: 13-15 second
Price: between 380 -410 USD
Retailer Rating: > 8/10
Retailer Employees: > 30
Customer Service Response Rating: > 9/10
On Site Repair: Yes
```

Fig. 4 The pull criteria data for digital camera

4 Related works

The MobiPass architecture is partly based on our previous Web Services research [23]. As both ubiquitous computing and service oriented computing have to deal with highly dynamic environments, it is essential to have an architecture to create a trusted business platform in such unpredictable environments. Also, our research is built on the Xplorer [17] framework, which is used to generate interfaces automatically by referring to XML schema.

Kagal et al. [24] propose the Centaurus system which provides fine-grained access control in their Smart Office ubiquitous computing scenario. The system utilizes the distributed trust approach and extends RBAC to allow a foreign user (from another security domain) to be granted proper privileges to gets access. Based on their implementation, Yang and Rhee [25] extend Centaurus by adding threshold of proxy signature schema to enable the validation period of the system. However, the approach of controlling the delegation of trust has not been solved perfectly in both architectures as both architectures do not provide a way to let foreign users to discover and find the most appropriate delegate.

A labeling protocol P3P [26] provides a rich vocabulary for detailing what data will be collected and the intention of collection. Although P3P is initially supposed to be used in the Web environment, the work has been extended to the ubiquitous area. PawS [27] is the security architecture which is used to help users to build a trusted, private protection system. However, as P3P-based systems can not help the requestor to clearly know the certain necessary background or detailed information of the requestee and service infor-

mation, it can only build the general agreement between requestor and requestee. The P3P labeling architecture also does not guarantee the genuineness of the information during the interaction between mobile entities, as this is the critical element in building a trusted infrastructure.

Hong and Landay [7] propose the architecture to perform the authentication and authorization in ubiquitous computing by assigning tags to pieces of information; information is associated with policy and indicated by the tag. The architecture is over conservatively designed as any small set of data needs a related tag and it assumes that the environment is administrated by a single entity but it does solve the problem of how to ensure the trust level in intermediate nodes.

Our approach is superior in its ability to allow parties that have previously not interacted to establish trust and make fine-grained access control decisions. This is required in realistic pervasive computing. The architecture runs fully in the distributed style, any organization can be an ECA or policy publisher, which provides enough flexibility to the ubiquitous and mobile computing area. And as the different service might have totally different sets of parameters, MobiPass is proposed to encapsulate the descriptive attributes to create the trusted platform which does not required the transaction entities to have much previous knowledge. This is quite different with current digital certificate and PKI architecture as the current digital certificate is mainly used to prove the identity of a certificate holder, which is very hard to be extended to make into a diversity of services and the data structure are also very limited.

Park and Sandhu [28] proposed a concept named the Smart Certificate for improving scalability in web servers, which has some interest to our work. The smart certificate is an extended version of X.509 certificate with several remarkable features. When accessing the web server, CA issues a smart certificate, including a subject's identity and role information, and Web servers will check the smart certificate and assign the privileges to the requestor depending on the role list. However, the smart certificate with RBAC model still has several shortcomings. Firstly, the key value pair attributes are not comprehensive and flexible to describe real business entities. Secondly, in this model, users have to go to the role server at first to retrieve the role list; this step is not necessary and can lead to two problems, which are flexibility and scalability in role servers. One of the reasons to contact role server is that there is not comprehensive information in the smart certificate themselves—if a smart certificate carries enough information, the service provider can assign the privileges to an incoming request automatically.

5 Future work and conclusion

In this paper, we have proposed a passport-based architecture to convert the unpredictable, highly dynamic ubiquitous network into a trusted environment and describe the architecture's application to mobile commerce with a case study of mobile retailing in particular. It utilizes the passport concept to evaluate, classify and certify the attributes of mobile entities. Users can easily set preference rules against the MobiPolicy to instruct the mobile entity as to how to act with previously seen or unseen entities. Case studies have been used to show how the MobiPass architecture can create, for example., an ubiquitous retailing environment and also show some of the resulting benefits and new retailing capabilities for the transaction parties (retailers and shoppers) that can be achieved. As the architecture provides a solid foundation for building a trusted and autonomous platform, it can greatly increase the users' confidence in interacting in pervasive environments and boost mobile business. A generic MobiPass prototype has been developed using J2ME to verify the architecture and feasibility. Our ongoing research is to refine the MobiPolicy structure and to make the generic MobiPass architecture generic enough to be able to interface with other trustworthiness-based systems that may be utilized by mobile systems.

References

1. Weiser M (1991) The computer of the 21st century. Sci Am 265(3):66–75
2. Fano A, Gershman A (2002) The future of business services in the age of ubiquitous computing. Commun ACM 45(12):83–87
3. Ackerman MS (2004) Privacy in pervasive environments: next generation labeling protocols. Pers Ubi Comp 8(6):430–439
4. Dourish P, Grinter E, Delgado de la Flor J , Joseph M (2004) Security in the wild: user strategies for managing security as an everyday. Practical problem. Pers Ubi Comp 8(6):391–401
5. English C, Nixon P, Terzis S, McGettrick A, Lowe H (2002) Dynamic trust models for ubiquitous computing environments, workshop on security in ubiquitous computing, UBICOMP, Göteborg Sweden. Available at http://www.teco.edu/~philip/ubicomp2002ws/organize/paddy.pdf
6. Cahill V, Gray E, Seigneur J-M, Jensen CD, Yong Chen Shand B, Dimmock N, Twigg A, Bacon J, English C, Wagealla W, Terzis S, Nixon P, Di Marzo Serugendo G, Bryce C, Carbone M, Krukow K, Nielson M (2003) Using trust for secure collaboration in uncertain environments. Pervasive Comput IEEE 2(3):52–61

7. Hong J, Landay J (2004) An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the 2nd international conference on mobile systems, applications, and services, pp 177–189
8. Siau K, Shen Z (2003) Building customer trust in mobile commerce. Commun ACM 46(4):91–94
9. Varshney UL, Vette RL (2002) Mobile commerce: framework, applications and networking support. Mobile Netw Appl Springer 7(3):185–198
10. Nah F, Davis S (2002). HCI research issues in electronic commerce. J Electron Com Res 3(3):98–113
11. Jarvenpaa SL, Tractinsky N, Saarinen L, Vitale M (1999). Consumer trust in an Internet store: a cross-cultural validation. J Comput-Med Commun 5(2):44–71
12. iShankar V, Urban GL, Sultan F (2002) Online Trust: a stakeholder perspective, concepts, implications, and future directions. J Strategic Inf Syst 11:325–344
13. Sarker S, Wells JD (2003) Understanding mobile handheld device use and adoption. Commun ACM 46(12):35–40
14. Siau K, Sheng H, Nah F, Davis S (2004) A qualitative investigation on consumer trust in mobile commerce. Int J Electron Bus (IJEB) 2(5):283–300
15. Price BA, Adam K, Nuseibeh B (2005) Keeping ubiquitous computing to yourself: a practical model for user control of privacy. Available at http://www.mcs.open.ac.uk/ban25/papers/ijhcs-2005.pdf
16. Bardram JE (2005) The trouble with login: on usability and computer security in ubiquitous computing. Pers Ubi Comp DOI 10.1007/s00779-005-0347-6
17. Steele R, Gardner W, Rajugan R, Dillon TS (2005) A design methodology for user access control (UAC) middleware. In: Proceedings of the 2005 IEEE international conference on e-technology, e-commerce and e-service (EEE'05), pp 385–390
18. Turner A (24/10/2004) Reasoning about naming systems. The Sydney Morning Herald
19. Kouriuthanassis P, Roussos G (2003) Developing consumer friendly pervasive retail systems. IEEE Pervasive Comput 2(2):32–39
20. Moussouri T, Roussos G (2004) Consumer perceptions of privacy, security and trust in ubiquitous commerce. Pers Ubiq 8(6):416–429
21. Bellamy R, Swart C, Kellogg WA, Richards J, Brezin J (2001) Designing an E-grocery application for a palm computer: usabilityand interface issues. Pers Commun IEEE 8(4):60–64
22. Newcomb E, Pashley T, Stasko J (2003) Mobile computing in the retail arena. In: Conference on human factors in computing systems, Proceedings of the SIGCHI conference on human factors in computing systems. Ft. Lauderdale, Florida, USA, pp 337–344
23. Steele R, Tao W (2005) An architecture for unifying web services authentication and authorization. In: 3rd International conference on service oriented computing. Amsterdam, The Netherlands, pp 582–587
24. Kagal L, Finin T, Joshi A (2001) A security architecture based on trust management for pervasive computing systems. Grace Hopper Celebration of Women in Computing, October 2002
25. Yang JP, Rhee YH (2005) Securing admission control in ubiquitous computing environment. In: 4th International conference on networking, vol 3421. Lecture notes in Computer Science, Reunion Island, France, April 17–21, pp 972–979
26. W3C, Platform for Privacy Preferences (P3P) Project, (2006), available at http://www.w3.org/P3P/
27. Langheinrich M (2002) A privacy awareness system for ubiquitous computing environments. In: UbiComp 2002, ubiquitous computing: 4th International conference, Göteborg, Sweden, p 237
28. Park JS, Sandhu R (1999) RBAC on the Web by smart certificates. In: Symposium on access control models and technologies, Proceedings of the 4th ACM workshop on role-based access control. ACM Press, New York, pp 1–9