ユビキタス環境に適した プライベート情報交換アーキテクチャの提案

今村 理[†1] 半井明大[†1] 大澤由憲[†1]
武田敦志[†2] 北形元[†3,†1]
白鳥則郎[†3] 橋本和夫[†1]

本稿では，次世代の高度なユビキタスサービスの実現のために，安全な個人情報の利活用に向けたプライベート情報交換アーキテクチャを提案する．ユビキタス環境でのプライベート情報交換には，(1) プライバシー権を保護すること，(2) 低コストで利用できること，(3) いずれのコンピュータに移動しても利用できることの 3 点が要求される．従来の Web での利用を想定したアーキテクチャでは，これらの要求全てを満たすことはできていなかった．本提案アーキテクチャでは，プライベート情報を各個人の管理下に保持し，情報要求元とネゴシエーションを行って情報提供する仕組みを公開鍵認証に基づいた分散認証基盤の上に構築することで，3 点の要求全てを満たした高度なプライベート情報交換を実現する．

Private Information Exchange Architecture for Ubiquitous Network Services

Satoru Imamura,[†1] Akihiro Nakarai,[†1]
Yoshinori Osawa,[†1] Atsushi Takeda,[†2]
Gen Kitagata,[†3,†1] Norio Shiratori[†3]
and Kazuo Hashimoto[†1]

This paper proposes a private information exchange architecture for ubiquitous network services. Private information exchange under ubiquitous network environment requires (i)preserving privacy, (ii)low cost availability, and (iii)portability across computer devices. Conventional private information exchange architecture for web services has not yet been able to satisfy all the above requirements. The proposed architecture satisfy all the three requirements by developing a private exchange scheme having private information under the owner's administration on which a private information recipient and the owner negotiate via distributed authentication infrastructure based on public key authentication.

1. はじめに

近年，サービスの展開に PI(Private Information:プライベート情報) を用いることが一般的になっている．ポータルサイト・ショッピングサイト・SNS 等，Web サービスでは PI を用いないサービスは非常に少ない．また，個人を識別する ID カードや携帯電話と連携することによって，Web サービスだけでなく実サービスでも PI を用いることが一般的になりつつある．今後のユビキタス化の進展に伴い，より多くのサービスで PI が利用され，様々な質の高いサービスが提供されると考えられる．

しかし，サービスプロバイダによる PI の利用が進むにつれ，PI を当人の意思に従って運用できる権利である PI コントロール権の侵害が問題となってきている．現在のサービスプロバイダは，サービス利用履歴や連絡先等の PI を大量に集め，当人の意思を確認することなくサービス展開に利用し，場合によってはアウトソーシングしているが，このことに不快感を感じる利用者も少なくない．サービス展開に必要以上の PI を収集し長期保持するサービスプロバイダも多く，セキュリティ上のリスクも高まっている．

一方で，利用者が PI を提供してサービスを利用したいと考える場面では，煩雑な PI の記入が必要となったり，サービスプロバイダが PI 管理コストを負えないためサービスが提供されないことが多い．特に小規模なサービス・地域限定のサービス・医療福祉サービスでは，PI を収集・管理するコストが負えず，利用者が何度も同じサービスを利用するとも限らないため，有効に PI を利用できない．

そのため，PI を個人の意思に従って有効に活用・保護する PI 交換アーキテクチャが望まれており，多数の研究が行われている．しかし，PI を当人の管理下に置きつつ，様々なコンピュータ環境でユビキタスに利用することのできるアーキテクチャは存在していない．そこで本研究では，PI を分散認証基盤上に置くことで外部の様々なコンピュータ環境から

†1 東北大学情報科学研究科
Graduate School of Information Science, Tohoku University
†2 東北学院大学教養学部情報科学科
Department of Information Science, Tohoku Gakuin University
†3 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University

安全に利用することを可能にし，かつ分散認証基盤上の PI サーバとサービスプロバイダが直接交渉することで利用者の意思に従った運用を可能にするアーキテクチャを提案する．

2. 関 連 研 究

2.1 ユビキタス環境における PI 交換

今後のユビキタス環境では，あらゆる場所にコンピュータが用いられ，そのコンピュータによるサービスの提供に PI が必要となってくることが想定される．例えば，次のような利用シナリオが考えられる．

食事の推薦 PI を用いることで，食事メニューの推薦が容易になる．食事メニューの意思決定には，食べ物の好き嫌い・予算・アレルギー・宗教等が影響する．人によっては，一日の摂取カロリーや栄養素も考慮する．これらの要素はむやみに知られたくない PI であるが，利用することによって食事メニューの推薦やパーソナライズが可能になり，良いサービスが提供できる．

買い物の推薦 PI を用いることで，商店をパーソナライズすることができる．タバコのように毎回買う銘柄が同じである場合は，その銘柄の情報を利用することで買い物を容易にすることができる．また，衣類の買い物の際には，あまり他人に知られたくない体型の情報を利用することで最適な商品を推薦することができる．

医療・健康サービス 薬歴等の PI を利用できれば，より進んだ医療サービスを提供できる．薬局での薬の購入の際に，過去に摂取した薬とそれに対する反応の情報を用いることができれば，より適した薬を処方することができる．知られたくない病気等を隠したまま身体情報を利用できれば，スポーツ施設等のサービスもより良くすることができる．

上記のようなユビキタス環境での PI の利用を実現するには，レストランの備え付け端末のようなユーザが初めて用いる端末でも利用可能で，むやみには公開したくない情報も取り扱える PI 交換システムが必要である．さらに，店頭備え付けの端末のように本人の端末ほどの信頼性のない端末でも利用可能である必要がある．

そして，地域のレストランのように小規模なサービスプロバイダ (SP) でも PI が利用可能にならなければならない．従来の PI の取得・管理方法ではシステムの導入コストが高いことが多かったが，地域限定サービスや医療福祉サービス等でも利用できるようにする必要がある．

一方で，人権の一つとして認められている PI コントロール権は保護する必要がある．PI は本人の意思に従って運用されなければならない．

本論文では，PI コントロール権を保護しつつ，様々なコンピュータから利用可能で，小規模 SP でも利用できる PI 交換アーキテクチャについて論じる．これらの観点から既存研究を整理した上で，新たな PI 交換アーキテクチャを提案する．

2.2 シングル・サインオンと PI 交換システム

ユーザ認証が必要なサービスが増えるに伴ない，ユーザが自身の ID・パスワードを適切に管理することが困難となり，利便性やセキュリティ面で問題となってきた．そのため，一組の ID・パスワードによって全てのサービスを利用できるようにするシングル・サインオンの研究が盛んとなった．

Shibboleth[1] は，統一された運用ポリシーのもとで複数の IdP(Identity Provider) がフェデレーションを作り，SP(Service Provider) に対して単一の IdP のように振舞うことでシングル・サインオンを実現した．既存の認証基盤を連携させることで実現できるため SP に大きな変更が必要ないという利点があるが，参加する IdP や SP の密接な協力が必要という問題がある．そのため，参加 IdP 間の運用ポリシーが近く，IdP や SP の信頼性の確認も容易な大学間等でしか用いることができない．

Web のように，IdP 間の運用ポリシーや IdP・SP の信頼性が異なる状況で用いられるシングル・サインオンのシステムとしては，OpenID[2] が実用化されている．OpenID では，ユーザ端末が SP と IdP のインタフェースとなって認証情報を提供することで IdP 間の連携を省略している．しかし，IdP の信頼性の確認ができないという問題があり，普及はあまり進んでいない．

そして近年になって，認証情報のみならず PI が SP のサービス展開に重要となり，PI を交換するシステムが必要とされてきた．その要求を満たすため，OpenID Attribute Exchange[3] と呼ばれるプロトコルが提案された．それにより，IdP がプライベート情報を OpenID の認証情報とともに提供できるようになった．しかし，OpenID は IdP や SP の信頼性に対する考慮があまりなされておらず，PI を提供することを嫌がるユーザが多いため普及は進んでいない．

それに対して，個人で簡易に使えるシステムとして，Web フォームを自動的に埋める Sxipper[4] 等のソフトウェアがブラウザ・プラグインとして提供されている．しかし，プライバシーポリシーの考慮等 SP との連携が不可能で，便利なソフトウェアの枠を出ることができない．

一方で，Microsoft は自社が信頼性のある単一の IdP として SP に認証を提供する.NET Passport[5] を実用化したが，SP が Microsoft に依存することを嫌ったため普及しなかった．

そのため，Microsoft は新たに Windows CardSpace[6] を提供した．CardSpace では，ユーザが財布の中に所属の異なる複数の名刺や社員証・クレジットカード等を入れておくように各 PI セットを管理し，ユーザクライアント中心の認証を提供する．さらにこのサービスでは，ユーザクライアント内に情報を保持して SP に提供する Personal Card と，IdP に認証を要求する Managed Card の 2 種類のカードを使い分けることで，PI を提供したいだけの場面とその証明を行いたい場面の両方に対応できるようになっている．

また，研究機関主導の次世代の PI 交換システムとして PRIME Archtecture[7] が提案されている．PRIME プロジェクトでは，PI をユーザが安心して利用できるようにするためには，ユーザの PI 開示ポリシーと SP の PI 利用ポリシーを定義し，PI 交換時にそれらのネゴシエーションを行うことが必要と認識された．その重要性は OpenID のコミュニティでも認知され議論されいる．Windows CardSpace においても，SP のプライバシーポリシーを PI 提供可否の判断材料としてユーザに提供する WS-SecurityPolicy をサポートしているが，それらはまだ十分ではない．それらと比較して，PRIME Architecture は強いポリシーのネゴシエーション機能を備えている．SP とユーザクライアント双方に PRIME middleware を導入することで高度な PI 交換のネゴシエーションを可能にし，SP が取得した PI にラベルを付けて管理することでネゴシエーションで決められた利用ポリシーに従った運用が強制できる．

2.3 既存アーキテクチャの比較

既存の PI 交換アーキテクチャは，PI を保持している場所から外部 IdP 型とユーザ端末型に分類することができる．外部 IdP 型では，IdP が PI を保持し，ユーザもしくは SP が PI 利用時に IdP に PI を要求する．ユーザ端末型では，PI はユーザ本人のユーザ端末 (UT) に格納され，ユーザ端末が SP からの要求を受けて直接 PI を提供する．

外部 IdP 型のシステムとしては OpenID が挙げられ，OpenID は図 1a の構成をとる．SP から IdP に対して，ユーザ端末を介して PI 要求が行われる．この方式の利点は，PI の保管場所がサーバに固定であるため，信頼できる端末であれば利用するユーザ端末を変えても PI を利用できるという点である．しかし，ユーザが完全に IdP を信頼して PI を預けなければならないためユーザ不安の解消が難しく，また IdP が情報を漏洩した時のリスクが高い．

一方，ユーザ端末型のシステムとしては PRIME Architecture があり，図 1b の構成をとる．ユーザ端末型では，外部の IdP は存在せず，PI は自身のユーザ端末に保持される．そのため，プライバシー保護の観点では優れている．しかし，PI を利用できる環境は PI を保持している端末に限られてしまうという欠点がある．PI をレストランで用いるためにはモバイル端末等で持ち歩かなければならず，物理的な紛失の危険性や利便性の観点から好ましくない．

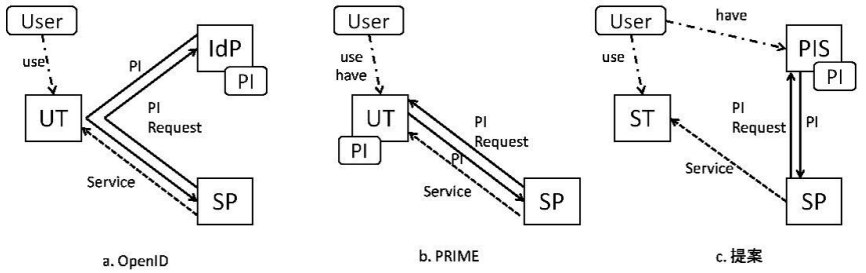

図 1　各アーキテクチャの構成

それぞれの既存アーキテクチャは PI の保管場所に特徴があるが，保管場所によって PI のデータを保持し閲覧できる主体も決まっている．本研究で考慮している PI には，病歴等の使われ方によってはユーザに不利益をもたらす情報も多いので，PI はユーザ本人が直接保持し，閲覧できるのは本人のみであることが好ましい．

また，近年提案されたアーキテクチャには PI 開示のネゴシエーション機能を持つものがある．ネゴシエーション機能は，PI 開示の可否を都度選択することと比べ PI 開示の際のユーザの思考負荷を軽減しつつフィッシング攻撃を防ぎ，PI を安全かつ潤滑に交換するための機能である．PI を利用する状況（SP の信頼性・PI 利用目的・PI 利用方法等）を理解し，状況に合わせて必要最小の PI を提供する．既存のアーキテクチャによっては，ネゴシエーション機能と呼ばれている機能は通信路のセキュリティに関するネゴシエーションである場合があるが，それについては本論文ではネゴシエーション機能と捉えていない．

ここまで紹介したアーキテクチャを，PI の管理者と PI の保管場所，ネゴシエーション機能の有無についてまとめると表 1 となる．PI の管理者はプライバシー保護の観点からユーザ本人が望ましく，PI の保管場所は物理的な紛失の危険性や利便性の観点から固定である方が望ましい．IdP 型のアーキテクチャでは保管場所については優れているが PI の管理者については望ましくなく，ユーザ端末型のアーキテクチャでは逆の特徴を示している．また，ネゴシエーション機能については，PI コントロール権を保護しつつ，自律的な PI 交換を行うために必要な機能であると考えられる．

表 1　アーキテクチャの比較

アーキテクチャ	PI の保持者	PI の保管場所	ネゴシエーション機能
Shibboleth	× IdP	○固定	×なし
OpenID	× IdP	○固定	×なし
Sxipper	○ユーザ	×移動	×なし
.NET Passport	× IdP	○固定	×なし
CardSpace	○ユーザ	×移動	×なし
PRIME	○ユーザ	×移動	○あり
提案手法	○ユーザ	○固定	○あり

表 1 のように，従来のアーキテクチャでは PI の管理者をユーザとしたまま PI の保管場所を固定とすることができず，理想的な PI システムが実現できていなかった．本研究においてはそれを可能とし，さらにネゴシエーション機能によって PI のコントロールも可能にするアーキテクチャを提案する．

3. 提 案 手 法

3.1 PIS(PI Server) の導入

本研究では，PI の管理者をユーザ本人としつつ PI の保管場所を固定とする PI 交換アーキテクチャを提案する．そのためにまず，その要求を満たす PI の保管場所として PIS(PI Server) を導入する．

PIS は，例えば家庭のブロードバンドルータに組み込むなどユーザサイドに設置するサーバで，SP にユーザ認証や PI を提供するパーソナル IdP の役割を行う．図 1c のように，ユーザ端末である ST(Service Terminal) を介さず SP に PI を提供する．ユーザ端末から離れた場所に PIS があることで，ひとつのユーザ端末に縛られることなく様々な端末から PI を利用することができる．また，PIS は，PRIME が備えるような PI ネゴシエーション機能を持ち，全ユーザが自身でこれを管理することで PI コントロール権が保護される．

しかし PIS を導入すると，いかにしてなりすまし攻撃を防ぎ認証を可能にするかという課題と，どのように PIS と SP や ST との間で連携するかという課題が生じる．前者の課題については，全 PIS に PKI による鍵認証を行えばこの課題は解決できる．しかし，PIS は一人一台保持されるので，全 PIS の証明書を一台のサーバに保管し検索可能にするには，大きな計算能力と通信能力が必要となる．また，システムに公共性が必要であることを考慮すると，一つの主体が管理することは好ましくない．そのため，本研究では分散認証基盤を用いることで解決する．我々の開発した分散認証基盤[8] を用いることにより，全 PIS の識別子 (PIS-ID) の一意性が保証できる．そして後者の課題について，次節で提案を行う．

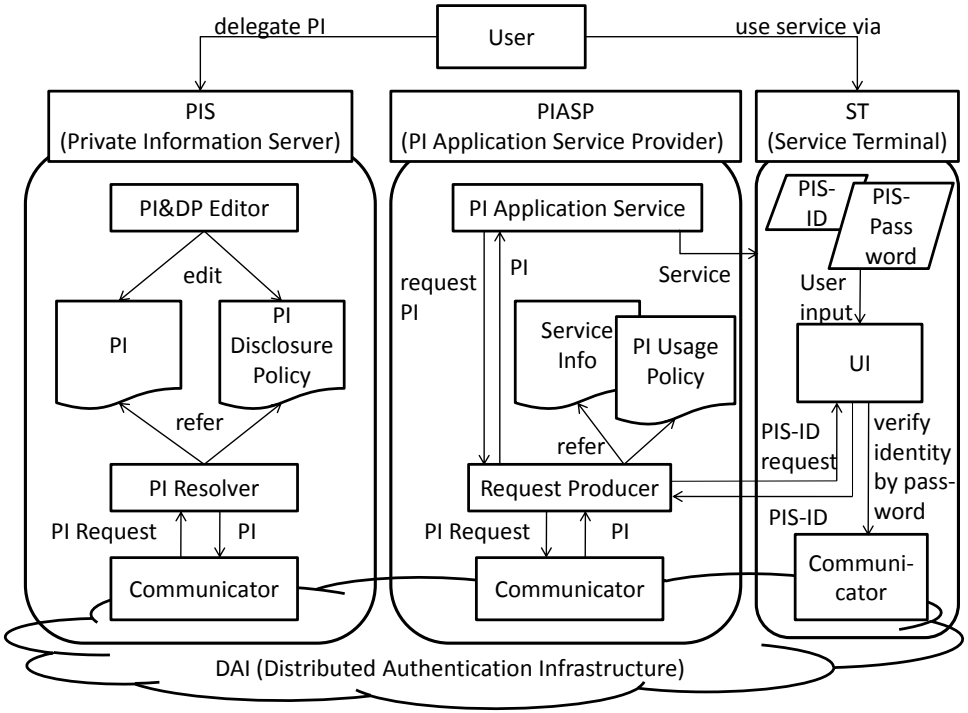

図 2　提案アーキテクチャ

3.2 提案アーキテクチャ

本研究では，図 2 のアーキテクチャを提案する．提案アーキテクチャは，大きく分けて，DAI(Distributed Authentication Infrastracture:分散認証基盤)，PIS，PIASP(PI Application Service Provider)，ST(Service Terminal) から構成される．

DAI は PIS と PIASP の Communicator に対して，ID の一意性を示す認証機能と公開鍵を用いた暗号化通信を提供する．PIS はユーザの PI とその開示条件 (PI Disclosure Policy) を保持し，PIASP との PI 開示ネゴシエーションを行う PI Resolver を持つ．PIASP は，ユーザに提供されるサービスの本体である PI Application Service とそのサービス情報 (Service Info)，取得した PI の利用方法等を記述した PI Usage Policy，そして Service Info

や PI Usage Policy を用いて PIS への PI の要求を PI Request としてまとめる Request Producer から構成される．ST は，PIASP からサービスを受ける機能とユーザが PIS-ID を入力するインタフェース (UI) を持つ．ユーザ (User) は，PIS に PI を委譲し，ST を介して PI を用いたサービスを受ける．

3.3 各モジュール間の連携

本節では，ユーザがレストランでメニューの推薦を受ける時を例として，提案アーキテクチャの各モジュール間の連携を説明する．

ユーザは，事前に PI&DP Editor を用いて，食べ物の嗜好等の PI と，その PI Disclosure Policy を “サービスタイプがレストランで仙台市の商店街に属するなら開示” というように登録する．レストラン（SP）側は，“仙台市の商店街に属するレストラン” というような自身の情報を Service Info に登録する．PI Usage Policy には，“PI はメニューの推薦のみに用い，アウトソーシングせず，取得後一日以内に削除する” というような PI 利用条件を記述する．

メニュー推薦サービス（PI Application Service）で PI が必要になると，ST からユーザの入力した PIS-ID が取得される．PIASP は Service Info と PI Usage Policy から PI Request をまとめ，PIS に PI を要求する．PIS では受け取った Service Info や PI Usage Policy を PI Disclosure Policy と比較し，一致すれば PI が PIASP に転送される．そして，PIASP から PI を利用したメニュー推薦サービスがユーザに提供される．

3.4 提案手法の特徴

提案手法の一番の特徴は，利用する ST を選ばないということである．PI の保管場所が固定であるため，どの ST からでもアクセスでき，ユビキタス環境での利用に適している．さらに，ST の信頼性の要求も従来のアーキテクチャと比較すると低い．PI の交換は ST を介さず PIS と PIASP の間でネゴシエーションに基づいて行われ，本人認証のために PIS-ID と PIS-Password の入力を行うが，それに関してもワンタイムパスワード等を用いれば保護できるためである．このような特徴を有しているにもかかわらず，PI の管理は完全にユーザ本人が行えるため PI コントロール権は強く保護されている．

さらに，PI 交換が PIS と PIASP 間で行われるため ST の行う処理や通信量は比較的少なく，バッテリ消費量が問題になるモバイルデバイスや太陽光発電で動作するデバイス等での利用にも適している．この点でも，あらゆる場所にコンピュータが設置されるユビキタス環境に適している．

また，IdP の分散と分散認証基盤の採用により，スケーラブルなアーキテクチャとなっていることも特徴である．理論的には全世界の人々が PIS を持ったとしても利用可能で，ユビキタス社会のインフラとしての機能を十分に果たすことができる．

4. 設計と実装

我々は，提案アーキテクチャの実現に向けた実装を進めている．一つの利用シナリオに適用するアプリケーションを作成し，提案アーキテクチャの有用性を示す予定である．

アーキテクチャのうち，DAI と PI Resolver については既に実装を行った．DAI については文献[9] に，PI Resolver については文献[10] に示している．本章では，DAI と PI Resolver の設計と実装の概略について説明する．

4.1 DAI の設計・実装

リング状のオーバレイ認証ネットワークを用いたノードの認証と暗号化通信によって，ノード間で PI を交換できる基盤を開発した．

オーバレイ認証ネットワークでは，基盤に参加する各ノードが互いに P2P 通信を行って公開鍵の交換を行う．通信先ノードの ID を用いて，そのノードの公開鍵を取得することができる．各ノードがリング上のネットワークを構成し，公開鍵の取得はリング状の複数のノードを介して行うことで，各ノードに必要なストレージ容量や通信負荷を下げている．そのため，スケーラビリティが非常に高く，社会的な基盤として十分な能力を持っている．欠点としては信頼の輪を用いているため不確実であることが挙げられるが，ある程度の共謀攻撃にも耐えられるようになっている．

オーバレイ認証ネットワーク上の PI 交換アプリケーションは次のように動作する設計・実装を行った．ユーザの PI とパスワードは事前に PIS ノードに登録されている．まず，ST ノードが PIASP ノードに対して PIS ノードの ID を渡してサービス要求を行う．PIASP ノードは PIS ノードに PI を要求し，それを受けた PI ノードは ST ノードに PI 提示許可の要請とパスワードによる ST ノード利用者と PI 所有者の同一性の確認を行う．PI 提示許可と同一性の確認を完了できれば，PIASP ノードに PI を提供し，PIASP ノードは ST ノードにサービスを提供する．

4.2 PI Resolver の設計・実装

PI Resolver は，ユーザ側の PI 開示ポリシーと SP 側の PI 利用ポリシーをマッチングすることで PI の公開可否を決定する．

我々は，PI Resolver が用いる PI や PI 開示ポリシーと，PIASP の情報や PI 利用ポリシーを図 3 のように設計した．各 PI には属する PI タイプがあり，開示条件を示す PI 開示

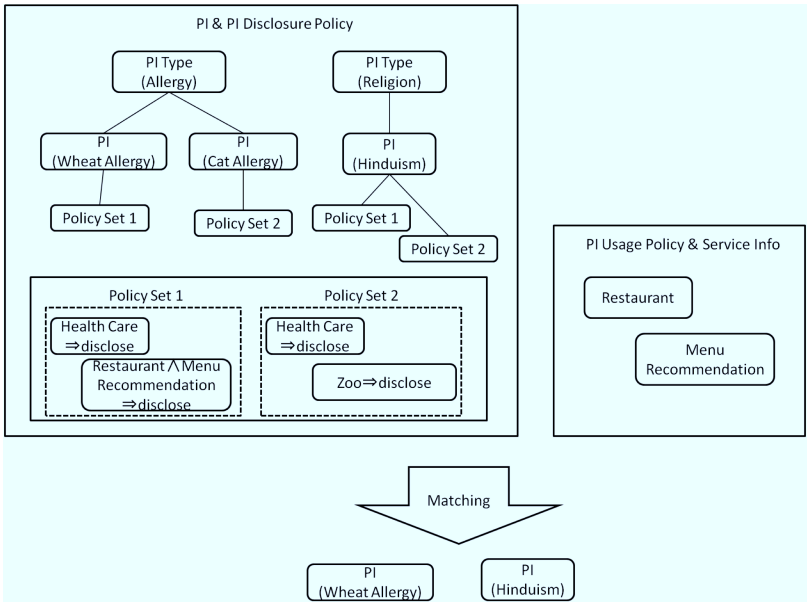

図 3 PI Resolver の概要

ポリシーに関連付けされている．PI 開示ポリシーはいくつかの開示条件を含む．開示する PI の抽出は，PI 開示ポリシーと PIASP の提示する条件をマッチングすることで行われる．

我々の実装では，PI と PI 開示ポリシー，そして PI 利用ポリシーを RDF データベースに二項関係のルールとして記述している．PI 要求を処理する際に，PI 開示ポリシーと PI 利用ポリシーを Allegro Common Lisp で解決し，開示できる PI が示される．

PI Resolver の実装には，具体的にルールを記述するための要素の選び方やユーザにわかりやすい記述方法について課題が残っており，今後研究を進めていく予定である．

5. ま と め

本論文では，ユビキタス環境に適した PI 交換アーキテクチャについて論じた．

従来提案されていた PI 交換アーキテクチャでは，利用できるユーザ端末が限定されてしまい，シームレスに様々なコンピュータを使い分けるユビキタス環境に適していなかった．そのため，本論文では様々なユーザ端末においても利用できる PI 交換アーキテクチャを提案した．

PI コントロール権を保護するため，PI をユーザサイドに置きつつユーザ端末に信頼性を要求しないアーキテクチャを目指し，本研究ではこれをユーザの近くに PIS(PI Server) を導入することで実現した．PIS の導入に伴う外部からのセキュアなアクセス方法の課題やなりすましの問題は分散認証基盤の利用によって解決した．そして，PI 開示ポリシーによる PI の運用を可能にして PI 利用時のユーザ操作を低減し，様々なコンピュータからシームレスに利用できる PI 交換基盤を構築した．

謝辞 本研究の一部は，情報通信研究機構 (NICT) の委託研究「ダイナミックネットワーク技術の研究開発」の助成を受けて実施したものである．

参 考 文 献

1) Shibboleth, http://shibboleth.internet2.edu/
2) OpenID, http://openid.net/
3) OpenID Foundation, “OpenID Attribute Exchange 1.0,” http://openid.net/specs/openid-attribute-exchange-1_0.html
4) Sxipper, http://www.sxipper.com/
5) Microsoft, “.NET Passport overview,” http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/passport_about.mspx
6) Microsoft, “Introducing Windows CardSpace,” http://msdn.microsoft.com/en-us/library/aa480189.aspx
7) D. Sommer, et al., “PRIME Architecture V3,” https://www.prime-project.eu/prime_products/reports/arch/pub_del_D14.2.d_ec_WP14.2_v3_Final.pdf
8) A. Takeda, et al., “A New Scalable Distributed Authentication for P2P Network and its Performance Evaluation,” WSEAS Transactions on COMPUTERS, vol.7, pp.1628-1637, 2008.
9) A. Nakarai, et al., “An Overlay Authentication Network for Active Utilization of Private Information,” International Symposium on Applications and the Internet, 2010.
10) Y. Osawa, et al., “A Proposal of Privacy Management Architecture,” International Symposium on Applications and the Internet, 2010.