社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－UBI－12（3）
2006／11／9

Secure Semantic Tunnelingによる異種スマート環境接続の設計

田中 宏一[†]　　河口 信夫[‡]　　西尾 信彦[§]

概 要

ユビキタス社会では，多種多様なスマート環境が遍在すると考えられる．多種多様なスマート環境を相互に接続し，サービスを連携させるためには，数多くの問題がある．特に，スマート環境は，通信方法，サービスアーキテクチャ，アプリケーション等の様々なレイヤで異なっていることが考えられる．本論文では，それらのスマート環境間の差異を乗り越えて，セキュアな相互接続・サービス連携を可能にする，Secure Semantic Tunneling モデルを提案する．また，Secure Semantic Tunneling モデルを，多種多様なスマート環境に対してスケーラブルにするための設計として，Semantic P2P Network を提案する．更に，これらのモデル，設計の実現方法として，スマート環境間の通信，スマート環境内のオブジェクトの検索システム，異種のサービスアーキテクチャにおけるプロトコル変換，及びスマート環境の適応的な連携について述べる．

System Design of Secure Semantic Tunneling for Cooperating Heterogeneous Smart Environments

Koichi Tanaka[†]　**Nobuo Kawaguchi**[‡]　**Nobuhiko Nishio**[§]

ABSTRACT

Ubiquitous society in future, it's contemplated that there will be many kinds of Smart Environments. There are a number of problems for connecting different Smart Environments each other and cooperating services. It's just conceivable that Smart Environments especially differ by various layer, communication method, service architecture, application, and so on. In this paper, we suggest that the Secure Semantic Tunneling model can interconnect Smart Environments and incorporate services throw their difference between Smart Environments. And we also suggest that Semantic P2P Network as design that make Secure Semantic Tunneling Model scalable for some different Smart Environments. While, we describe communication inter-Smart Environments, object in Smart Environments retrieval system, protocol translation for Heterogeneous service architecture and cooperating Smart Environments adaptive for realizing these model and design.

1　はじめに

将来のユビキタス社会では，多数のセンサや情報機器，各種ネットワークによって構築される環境，スマート環境が遍在すると考えられる．

そのようなユビキタス社会では，ユーザは一つのスマート環境にいるのではなく，様々なスマート環境，例えば，家，駅，オフィス，店舗，学校，ホテルなどを利用する．このとき，移動先の空間と普段の生活空間を接続することができれば，各スマート環境にあるコンテンツやサービスを，その使用が許可されたユーザが快適に利用することが可能になる．例えば，出張先のホテルや飛行場で，自宅の電話やインターホンからの呼び出しを受け取ることも可能になろう．このように，複数のスマート環境が相互に接続することで，ユーザに最適なサービスが提供できるようになると考えられる．

近年，スマート環境の研究は盛んに行われているが，既存研究は単一のスマート環境内に閉じた研究が多く，複数のサービスアーキテクチャ/スマート環境間を接続し，相互に利用可能にする，という研究は少ない．異種スマート環境の相互接続の実例として，2002 年の慶應義塾大学徳田研究室の Smart Space Lab と東京大学青山・森川研究室の STONE Room をインターネット上で相互接続した例が挙げられる．しかし，これは特定の異種スマート環境における相互接続であった．

我々はこれまで，スマート環境で稼動させるサービスアーキテクチャとして，United Spaces[1]，cogma[2][3] を研究してきた．特に United Spaces では，分散したスマート環境間でセキュアにサービスの連携を行な

[†]株式会社内田洋行
UCHIDA YOKO COMPANY LTD.
[‡]名古屋大学大学院工学研究科/情報連携基盤センター
Graduate School of Engineering, Nagoya University
[§]立命館大学情報理工学部
Department of Computer Science, Ritsumeikan University

本研究は総務省戦略的情報通信研究開発推進制度 (SCOPE)「異種スマート環境間をセキュアに動的接続・構成する基盤技術」の支援を受けて実施されている．

う機構を実装している．しかし，現状では同じサービスアーキテクチャのスマート環境の接続にとどまっている．同様に，同じサービスアーキテクチャによるスマート環境の接続を行っている研究として，Instant Matchmaking[4] がある．

多種多様なスマート環境を相互に接続し，サービスを連携させるためには，数多くの問題がある．スマート環境は，通信方法，サービスアーキテクチャ，アプリケーション等の様々なレイヤで異なっていることが考えられる．また，外部との接続を考慮されていないスマート環境も多い．更に，外部からスマート環境がコントロールできるようになっても，ただ接続・連携させるだけでは，それは重大なセキュリティホールになり得る．スマート環境はセキュアに接続されなければならない．

上記の問題を解決し，異種のスマート環境を接続・連携するため，本研究では Secure Semantic Tunneling というモデルを提案する．Secure Semantic Tunneling は，異種スマート環境を接続するトンネルを構築する (tunneling)．このトンネルは，スマート環境間の違いを適応的に判断して，容易につなぐことを可能にする (semantic)．また，許可されたユーザ・スマート環境・サービスだけが通ることができる，という安全性を持つ (secure)．

本稿では，まず，2 章で異種スマート環境の連携における問題を整理する．次に 3 章でその問題を解決するための，Secure Semantic Tunneling モデルを提案する．続いて Secure Semantic Tunneling モデルを実現するための設計を，4 つの章に分けて述べる．4 章では，スマート環境間の通信を確立する方法について，5 章では，外部からスマート環境内のオブジェクトを検索するシステムについて，6 章では，異種サービスアーキテクチャのプロトコル変換について述べる．4 章から 6 章で，異種スマート環境を相互接続・連携するための設計を述べた上で，7 章で適応的な異種スマート環境の連携を検討する．最後に 8 章にて，本論文をまとめる．

2 異種スマート環境の連携における問題

異種スマート環境の連携にいては，大きく二つの問題がある．

一つは，スマート環境で利用されるサービスアーキテクチャの違いである．異なるサービスアーキテクチャ同士では，サービスの発見・実行の通信プロトコルが異なる上，サービスの記述形式，やり取りされるデータのデータフォーマットが異なる．

次に，スマート環境間の通信に関する接続性の問題がある．スマート環境は，グローバルネットワークからのアクセスを考慮されたものは少ない．そのため，スマート環境を接続させるには，VPN の設定や，グローバルアドレスと NAT の設定が必要になる．これでは，必要に応じて任意のスマート環境を接続させることは難しい．また，IPv6 でスマート環境のネットワークを構築することで，スマート環境外部から接続することも可能だが，IPv6 のネットワークインフラはまだ十分ではない．

このように，異種スマート環境にあるノードが相互に接続してサービスを行うとき，それらのノードは，通信，サービスアーキテクチャ，データフォーマット等の違いを乗り越えなくてはならない．

上記の問題に加え，スマート環境の相互接続と，セキュリティを両立させる必要がある．

3 Secure Semantic Tunneling モデル

3.1 Secure Semantic Tunneling のコンセプト

2 章で述べた問題を解決するため，我々は Secure Semantic Tunneling モデルを提案する．

Secure Semantic Tunneling が構築するトンネルは，異種スマート環境をつなげるトンネルであり，どのような通信・サービスの差異 (IPv4/v6，NAT，サービスアーキテクチャなど) も吸収することができる．このトンネルを通すことで，異種のスマート環境間を越えて，自由かつセキュアにサービスを連携させることができる．

通信の差異を吸収するアプリケーションフレームワークとしては，Skype[12] が挙げられるが，Skype は通信環境の差異のみを吸収するだけである．また，Skype API を利用した Skype アプリケーションにしか対応できない．Secure Semantic Tunnel は，通信とサービスの両方のレイヤを対象として，通信の差異，サービスの差異を吸収する．Secure Semantic Tunnel は，スマート環境を構成するサービスアーキテクチャだけでなく，既存の一般的なネットワークアプリケーションも通ることができる．

3.2 Secure Semantic Tunneling の設計

本節では，Secure Semantic Tunneling の設計について述べる．

まず，異種スマート環境の接続のパターンについて検討する．異なるサービスアーキテクチャの機器を接続するためには，以下の 4 つの観点で設計の選択肢がある [6].

1. Translation Model (Direct or Mediated)
2. Semantic Distribution (Scattered or Aggregated)
3. Intermediary Semantics Granularity (Coarse-grained or Fine-grained)
4. Location of Interoperability Layer (At-the-Edge or Infrastructure)

本研究では，多種多様なスマート環境の相互接続を優先すべき要件とする．そのため，翻訳すべきサービスアーキテクチャの種類が不特定かつ多数であることを想定する．よって我々も，個々の機器・サービスに翻訳機能を実装して様々なサービスアーキテクチャに対応するのではなく，異種の機器・サービスの間に仲介者をおく (Mediated，Infrastructure) パターンを選択する．スマート環境の間に仲介者を置いたパターン (centralize) を図 1 に示す．

図 1 では，スマート環境のサービスアーキテクチャの違いを，サービスを表す図形の違いで表している．これは以降の図でも同様である．

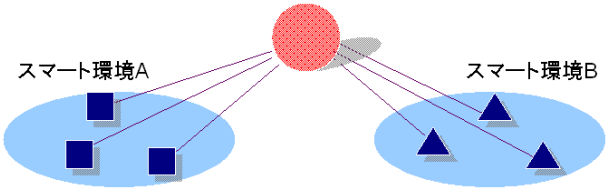

図 1: シンプルな仲介者の構成 (centralize)

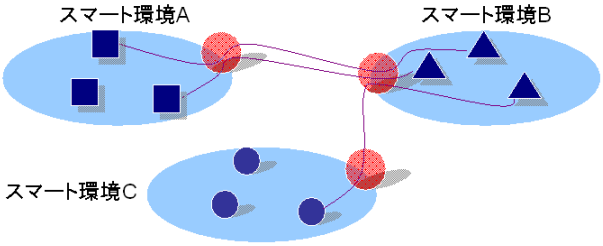

図 2: シンプルな仲介者の構成 (gateway)

この構成 (centralize) では，スマート環境は多数存在することを想定すると，仲介者がボトルネックとなりスケールしない．そのため仲介者は，各スマート環境の中で，他のスマート環境とつながるエンドポイントに存在するべきである (gateway)．この構成を図 2 に示す．

また，異種スマート環境を仲介するために翻訳を行うレイヤは，物理層からアプリケーション層まで多岐にわたり，また，それぞれのレイヤで翻訳を行うプロトコルの種類も，Ethernet と Bluetooth，IPv4 と IPv6，United Spaces と cogma と UPnP，と様々である，と考えられる．これを図 3 に示す．

3.3 Semantic P2P Network

このように，仲介者が行う翻訳は，物理層からアプリケーション層まで，多くのレイヤにわたる．また，各レイヤでも，多くのプロトコルの違いがある．そのため，エンドポイントにある仲介者だけに，上記に述べたようなすべての翻訳機能を実装することは困難である．また，新しいアーキテクチャで実装されたスマート環境と接続する場合に，仲介者に新しい翻訳機能をインストールする必要がある．

上記の問題を解決するため，本研究では，仲介者の機能を分散化した P2P ネットワークを構成し，P2P

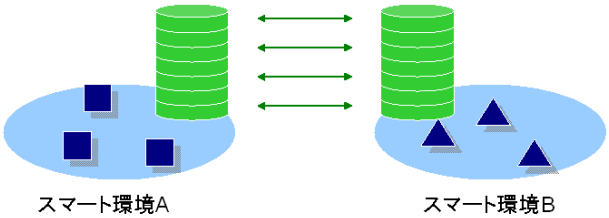

図 3: 多層的な仲介機能

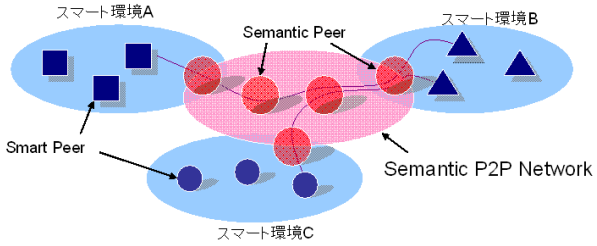

図 4: Semantic P2P Network

ネットワーク全体で仲介の機能を実現する，という設計を提案する．この設計を図 4 に示す．

まず，スマート環境を構成するピアを，Smart Peer と呼ぶ．Smart Peer は特に他のスマート環境と通信・連携する手段は持たず，単一のサービスアーキテクチャに対応する．

P2P ネットワークを構成するピアは，通信・サービスの差異を理解し，翻訳するため，Semantic Peer と呼ぶ．Semantic Peer は，スマート環境を構成する Smart Peer の一つだが，外部と通信する手段 (グローバルアドレス，IPv6，など) を持つ．

Semantic Peer が構成する P2P ネットワークは，Semantic P2P Network と呼ぶ．Semantic P2P Network は，特定のプロトコルの翻訳機能を持つ複数の Semantic Peer を組み合わせて，異種のスマート環境をつなぐトンネルを構成する．このトンネルを Semantic Tunnel と呼ぶ．

仲介者の機能を P2P ネットワークで実現することで，Semantic Tunnel にはスケーラブル・冗長化といった特徴も持たせることができる．また，新しい翻訳機能を持った Semantic Peer が Semantic P2P Network に参加すると，Semantic P2P Network に参加している全てのスマート環境で，新しい翻訳機能を利用することができる．

但し，この設計においてはセキュリティを考慮しない．セキュアに仲介者を構成するには，図 2 で示したように，接続するスマート環境を特定して，エンドポイントに仲介者を配置する．

Semantic Peer には，設定情報として，翻訳可能なプロトコルのリストを持たせる．設定情報は，例えば IPv4 と IPv6 の変換が可能，Jini と UPnP の変換が可能，といった情報である．これらの設定情報は Semantic P2P Network で共有される．これにより Semantic P2P Network は，必要に応じて最適な Semantic Peer を経由する Semantic Tunnel を構築し，通信・サービスの変換を実現する．

3.4 Secure Semantic Tunneling で実現する適応性

Secure Semantic Tunneling における Semantic とは，異種スマート環境のサービスを連携させるにあたって，必要なプロトコル変換を適応的に選択することを表す．例えば，IPv4 で構成されたスマート環境にあるサービスが，IPv6 で構成されたスマート環境にあるサービスと通信する場合，IPv4 と IPv6 の変換が必要であると判断され，IPv4/IPv6 変換機能を持った

Semantic Peer が変換を行う．また別の例では，TCP による通信を行う TV 会議システムと，UDP による通信を行う TV 会議システムをつなげる場合，Semantic Peer は互いの TV 会議システムが送受信するデータを，TCP 側には TCP として，UDP 側には UDP として返す．

また，Semantic Tunnel は Smart Peer に対して，トンネルの接続先のスマート環境を，同じネットワークのように見せることができる．連携する 2 つのスマート環境のサービスアーキテクチャが同じであれば，同一ネットワークに見せることで通信は確立できる．この場合，Semantic Tunnel は，サービスを操作するパケットだけでなく，サービスの発見パケットもトンネリングすればよい．このように，プロトコル変換がない場合には，同一のネットワークに見せる単純なトンネルとなり，そうでない場合はプロトコルを変換するトランスレータになる．接続方式を適応的に選択することも Secure Semantic Tunneling の特徴である．

3.5 実現方法

本章では，Secure Semantic Tunneling モデルについて述べた．以降の章では，Secure Semantic Tunneling を実現するための設計について述べる．

Secure Semantic Tunneling モデルではスマート環境を連携させるため，以下のシーケンスを想定する．また，各項に対応する章を示す．

1. 連携するサービスを検索し，検索結果としてサービスが属すスマート環境を特定する (5 章)
2. 接続先のスマート環境とのトンネルを作る (4 章)
3. 適切なプロトコルの変換を行い，サービスとの連携を行う (6 章)

4 スマート環境間の通信

本章では，異種スマート環境の連携において，特に通信の接続性について述べる．具体的には，スマート環境において外部ネットワークとの接続性を持たせるためのシステムを設計する．我々はこの機能を Sementic Peer に実装する．

4.1 要件

スマート環境は，プライベートネットワークに構築されることが多く，他のスマート環境とインターネットを越えて接続されることは，あまり考慮されていない．そのため，同種のスマート環境を接続する場合でも，VPN を用いて接続先のスマート環境を同じネットワークにあるように見せる，スマート環境にグローバルアドレスを与えて NAT の設定を行い，クライアントからアクセス可能とする，スマート環境を IPv6 のネットワークで構築する，などの，事前の設定が必要となる．

これでは，必要に応じて任意のスマート環境と相互接続することは難しい．これらの問題を解決してスマート環境を相互接続するため，我々は次の要件を設定した．

1. プライベートネットワークに構築されたスマート環境であっても通信可能である

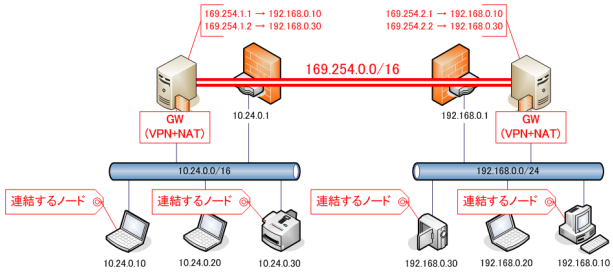

図 5: GW ノードによるスマート環境の接続

2. 1 の状況のスマート環境を通信可能にするにあたって，ネットワーク機器に特殊な設定を行わない
3. スマート環境を構成するホストに特殊なプログラム (VPN クライアントプログラムなど) をインストールしない

1 は，異なるプライベートネットワークにあるスマート環境同士を，どのように通信可能とするか，という要件である．更に 2 で，Ad-hoc にスマート環境を通信可能とすることを目指す．スマート環境間を通信可能とするにあたり，VPN の設定や，NAT の設定を行うことは，問題で挙げた事前準備にあたる．また 3 は，スマート環境を構成するホストが，プリンターやネットワークメディアプレイヤーなどの PC ではないデバイスであっても，他のスマート環境と通信可能とすることを想定している．

4.2 概要

上記の要件を満たすため，我々は Secure Semantic Tunneling の設計においてゲートウェイノード (以下，GW ノード) を導入する．GW ノードは，Semantic Peer と同一であるが，ここでは Semantic Peer の持つ機能のうち，特に通信の機能についてのみ論じるため，あえて GW ノードと呼ぶ．

GW ノードは，スマート環境において，外部と通信可能なノードである．GW ノードが他のスマート環境との通信を経由することで，外部との通信手段を持たないノードも，他のスマート環境と通信することが可能となる．Skype では，Skype をインストールしたホストしか他のネットワークのノードと通信することができない．この例で言えば，GW ノードは，Skype をインストールしていないノードでも他の Skype クライアントと通信させる機構である．

4.3 GW ノード

外部と通信させたいノードは，デフォルトゲートウェイを GW ノードに設定する．GW ノードは受け取ったパケットを，接続先の GW ノードに転送する．内部のノード (Smart Peer) から見て，GW ノードはデフォルトゲートウェイのプロキシとなる．これによりスマート環境内のノードは，異なるネットワークにあるスマート環境のノードと通信することができる．この構成を図 5 に示す．

この構成では，GW ノードがスマート環境間を透過的に見せるときに，2 つのネットワークのアドレス体

系を考慮する必要がない．つまり，接続先のスマート環境のホストに，ローカルのホストと重複したアドレスがあったとしても，多対多のアドレス変換を行うことで，アドレスの重複を解決する．

また，GW ノード間の通信は，適応的に選択される．グローバルアドレスを持たないスマート環境では，GW ノードは NAT 越えを行い，グローバルアドレスを持つ場合は，直接の通信を行う．また，接続する GW ノード同士が IPv6 で通信可能な場合，両者は IPv6 にて通信を行う．

4.4 名前解決

また，GW ノードには，DNS Proxy の機能を持たせる．これにより，DNS クエリの情報を利用することができるようになり，動的に GW ノードがトンネルの接続先を決定したり，GW ノードに動的にアドレス変換のテーブルにノードを登録する，ということが可能になる．

また，DNS による名前解決を拡張することができる．DNS(ホスト名) は既存のネットワークアプリケーションでも利用するため，5 章で述べるオブジェクトの検索システムを GW ノードに実装することで，既存のネットワークアプリケーションでも異種スマート環境内のホストの名前解決を行うことができる．

但し，実装にあたっては，OS の DNS をキャッシュする機構が，本設計に影響することが考えられる．この問題については，実装を行った上で評価したい．

4.5 セキュリティ

GW ノードは，スマート環境間のすべての通信を仲介するため，セキュリティに関する設定は GW ノードに対して行うだけでよい．特に，5 章で述べる，スマート環境内のオブジェクトの指定を利用することで，特定のオブジェクトに対する詳細なアクセス権限の設定が可能である．

4.6 スマート環境以外からの接続

外部からスマート環境に接続するシナリオとして，ユーザが外出先のホテルから，携帯端末を用いて家のスマート環境に繋げ，家のファイルサーバのコンテンツを，ホテルにあるネットワークメディアプレーヤーで再生する，というシナリオが考えられる．このシナリオでは，ホテルには予め GW ノードはなく，また，ホテルのネットワーク構成を変更することができない．

この場合，携帯端末が GW ノードになることで，家のネットワークと繋げることができる．ただし，これは携帯端末が繋がるだけである．

ホテルのネットワークにあるネットワークメディアプレーヤーなどのサービスを利用したい場合では，ネットワークメディアプレーヤーのデフォルトゲートウェイは変えられない．この場合は，ネットワークメディアプレーヤーからみて，携帯端末が一対多のアドレス変換を行う．また，DNS Proxy も利用できないため，GW ノードの IP アドレスを通信相手として指定するか (NAT)，サービスの発見を行うサービスアーキテクチャの場合，発見のパケットを GW ノードが経由することで解決できると考えられる．

また，上記の問題は，ホテルのネットワークが IPv6 で構成される場合，複数のデフォルトゲートウェイを設定できるため，解決することができる．

今後は，GW ノードによる構成を実装した上で，現状の構成だけでは対応しづらいネットワーク環境について検討する．

4.7 状況検知

4.6 で述べたように，ユーザは携帯端末を持ってスマート環境を移動することが考えられる．

本研究では，このような状況において，現在そのホストがどのようなネットワーク・状態にあるかを検出する機構を開発する．この機構は，ホストに常時稼動させておき，例えば，自宅からオフィスに移動したとき，ネットワークが変わったことを自動的に検知させる．この機構には次の二つの目的がある．

第一の目的は，ユーザのいる場所を自動的に判断させることである．ユーザが携帯端末を持ってある場所に行った際，携帯端末の状況と場所名を対応付けて登録しておく．これには，無線 LAN の SSID[8][13] やデフォルトゲートウェイの MAC アドレスが利用できる．こうすることで，次回ユーザが同じ場所に来たときに，そこがどのような場所だったかを自動で判断することができる．

これを発展させることで，ネットワークの情報から場所名という抽象的な情報を導き出すことができ，Semantic P2P Network で異なる表現の場所名を共有することで，セマンティックな場所の検索が行えるようにする．

第二の目的は，トンネルの接続先や最適な接続方法を決定することにある．Skype は，自身が実行されているホストが，どのようなネットワーク・状態なのかを検知する．この状況検知を利用して，4.3 で述べた，GW ノードによる適応的なトンネル構成を行う．

5 スマート環境のオブジェクトの検索システム

本章では，外部から，スマート環境内部のユーザ，サービスを，どのように検索・指定するかについて述べる．本章で検討する検索機構は，Semantic P2P Network の構成と，そこでやりとりするデータとして実装される．

5.1 要件

本研究では，スマート環境は要求に応じて Ad-hoc に相互接続されることを想定している．そのため，4 章で述べたスマート環境間のトンネルは，常時接続されているわけではない．そのため，どのスマート環境と接続すればよいかを決定するために，事前にユーザ，場所，サービス (以下，オブジェクト) を検索する必要がある．本システムの概要を図 6 に示す．

検索の結果として，オブジェクトを一意にする識別子を受け取る．この識別子は他のスマート環境にあるオブジェクトを識別できなくてはならない．Ad-hoc に他のスマート環境に接続するため，この識別子に IP アドレスは利用できない．よって，少なくともホスト名のような方法でオブジェクトを指定する必要がある．

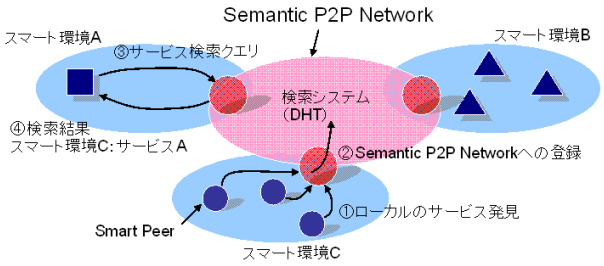

図 6: スマート環境のオブジェクトの検索システム

更に，ユーザが多くのスマート環境・サービスを扱うことを考えると，明確にホスト名がわからない状況でも，オブジェクトを特定できることが望ましい．

オブジェクトの検索を行う場合，ユーザ，場所，サービスは，それぞれ異なる特徴を持つ．まず，ユーザは移動し，様々なスマート環境に存在し得る．サービスは，様々なサービスアーキテクチャ，アプリケーションで定義されるため，その記述形式が異なる．場所 (スマート環境) はユーザ・サービスを含み，場所が決まれば，ユーザ・サービスが見つけられるようにしたい．これにより，例えばユーザがいる場所にあるサービスを発見する，というクエリを実行することができる．

本章で検討する要件を整理し，以下に示す．

1. ユーザ，場所，サービスを検索するためのシステムを，GW ノード（Semantic Peer）同士のネットワーク (Semantic P2P Network) で実現する
2. 一般的な名前解決 (DNS，SIP など) との親和性を持たせ，これらを利用したサービスアーキテクチャ，アプリケーションが動作できるようにする
3. ユーザ，場所，サービスの関係を定義する．検索結果として，ユーザが居る場所のサービスが見つかるようにする
4. ユーザ，場所，サービスを抽象的なクエリで検索できる
5. ユーザは移動したり，サービスは追加されることがあるため，存在通知の仕組みを持たせる

5.2 アプリケーション

本章で検討する検索システムにより，次のようなアプリケーションを実現することができる．

一般的な TV 会議システムは，接続先の IP アドレスを指定して TV 会議を開始する．本検索システムでは，ユーザを検索することで，そのユーザが現在いるスマート環境が特定できる．TV 会議システムがユーザのラップトップ PC ではなく，スマート環境にある TV 会議システムであっても，ユーザのいる場所のサービスが分かることで，TV 会議を行うことができる．これにより，ユーザがラップトップ PC を持ち歩く必要がなく，複数のスマート環境を移動しても，そのときに居る場所の TV 会議システムに接続される．また，ユーザの近くにいる人の PC に TV 会議システムがインストールされていれば，その PC と TV 会議を行うことも可能である．

このように，接続先にどのようなシステムがあり，ネットワーク環境や IP アドレスがどうなっているかを気にすることなく，ユーザを指定してサービスを実行することが可能になる．

5.3 検索システム

オブジェクトの検索システムは，Semantic P2P Network で実現する．

Semantic Peer はスマート環境における Smart Peer でもある．そのため，Smart Peer が実装しているサービスアーキテクチャによって，スマート環境にどのようなサービスがあるかを知ることができる [7]．また，スマート環境の場所情報は，Semantic Peer にのみ設定すればよい．

Semantic P2P Network では，DHT を利用して，ユーザ，場所，サービスの情報を共有するとともに，検索の機構を提供する [7]．

5.4 クエリ

ユーザ，場所，サービスの関係を考慮したクエリには，Intentional Naming System[9] や ASAMA[10] のような属性ベースの検索技術を利用することができる．

本研究では，これに加えて，検索対象がスマート環境であること，ユーザとサービスが動的であることを考慮する．動的とは，ユーザの場合，ユーザがスマート環境を移動することを指す．サービスの場合，多くのサービスアーキテクチャでは，IP アドレスに依存せず，サービス自身が存在通知を行うことを指す．

また，ユーザ，場所，サービスを抽象的なクエリで検索できるように，オントロジーによる検索を検討する．

5.5 オブジェクトの識別子

クエリの結果は，場所 (スマート環境) を表す識別子と，ユーザ，サービスを特定できる識別子になる．

場所の識別子は，Semantic Peer がトンネルを接続しなければならない場所 (スマート環境) を表す．

Semantic Peer がトンネルを構築すれば，ユーザやサービスを表す識別子は，サービスやアプリケーションのクライアントにわたされる．そのため，これらの識別子は，一般的に利用されている表現形式が望ましい．但し，特殊なアプリケーションやサービスアーキテクチャでは，独自のユーザアカウントやサービス記述を利用するため，この限りではない．

よって Secure Semantic Tunneling では，ユーザの識別に SIP，サービスの識別に URL を使用することを検討する．URL はスキーム，ホスト名，ポート，パスで構成されることから，サービスを指定するのに最低限の情報を持っていると言える．

また，4.4 で述べたように，DNS Proxy を利用することで，ホスト名を利用して適応的なトンネルを構築することも可能になると考えられる．一方で，ホスト名は OS のキャッシュの問題があり，ユーザやサービスの IP アドレスが頻繁に変わる状況で正しく機能するか，評価をする必要がある．

5.6 ユーザの特定

ユーザの識別子に関しては，P2P SIP[11] が研究されており，SIP の機構を Semantic P2P Network へ導入することが可能であると判断できる．P2P SIP では，DHT を SIP の Location Service に利用する．これは仲介機能を分散化させた Semantic P2P Network のモデルと親和性が高い．

一方，Semantic P2P Network では，ユーザが移動すること，ユーザは場所及び近くのサービスとの関連を持つことから，P2P SIP を拡張する必要があると考えている．また，P2P SIP ではユーザ名の認証の問題があり，これについては Semantic P2P Network でも解決する必要がある．

6 異種サービスアーキテクチャの連携

本章では，異種のサービスアーキテクチャを連携させるためのプロトコル変換，及び最適な連携方式について述べる．ここで実現される機構は，Semantic Peer の翻訳機能に実装される．

6.1 プロトコル変換

我々は，異種のサービスアーキテクチャを翻訳するにあたって，そのプロトコルはコントロール用とデータ用に大別できると考えている．

コントロール用のプロトコルは，サービスを操作するためのプロトコルにあたる．コントロール用のプロトコルは，United Spaces，cogma，UPnP，Jini，SOAP など多種であるが，一方で多くのサービスアーキテクチャでは，RPC の形式をとる．つまり定型的なプロトコル群であり，そのため，異種のプロトコル間での変換は比較的容易であると考えられる．

データ用のプロトコルは，サービス同士でやりとりされるデータの記述形式にあたる．データの記述形式には，RPC の引数や戻り値で利用されるようなプリミティブ型や，複雑なデータを表現するための構造体，動画や音声などのマルチメディアデータに大別できる．ここで，前者二つは，コントロール用のプロトコルと同じく定型的であり，変換可能であると考えられる．一方，マルチメディアデータに関しては，標準規格を利用することが多く，プロトコル変換を行うことは少ない，と考えられる．

我々は，同じデータフォーマットを利用しているが，操作方法だけが違うアプリケーションが多くある，と予想している．

また，多くのサービスアーキテクチャでは，イベントがサポートされている．スマート環境では，イベントの通知は重要な機能である．イベントの記述形式の変換や，スマート環境の外へのイベント通知について検討を行う．

今後は，これらの仮説を検証するため，様々なアプリケーションのプロトコルを調査する．また，異種のサービスアーキテクチャを連携させる，様々なプロトコル変換を実装し，翻訳に関する問題の明確化と，最適な設計を検討していく．実装においては，中間言語を用意するのではなく，互いの言語に翻訳し合うことをポイントにおく．

6.2 サービスアーキテクチャの仲介方式

コントロール用のプロトコルでは，RPC 以外の部分で，サービスアーキテクチャが持つ特徴が問題となる場合がある．サービスアーキテクチャは，それぞれ異なる特徴を持ち，そのため，単純なプロトコルの変換を行うだけではうまく連携することができない場合が考えられる．

例えば，ステートフルな通信を行うサービスアーキテクチャを，ステートレスなサービスアーキテクチャから実行するとき，単純なプロトコル変換だけでは，連携は難しい．United Spaces のようなセキュリティを考慮したサービスアーキテクチャや，cogma のようにソフトウェア移送を行うサービスアーキテクチャを，ごくシンプルなサービスアーキテクチャから実行するときも同様である．

我々は，このようなサービスアーキテクチャを連携させるためには，トランスレータ方式ではなく，プロキシ方式が適している，と考えている．プロキシは，サービスアーキテクチャのクライアント実装であり，より特化した実装が可能で，特殊なサービスアーキテクチャのプロトコルを隠蔽することができる．

このように，異種のサービスアーキテクチャを連携させる場合には，仲介する機能として，トランスレータ方式と，プロキシ方式の 2 つの方法がある．似通ったサービスアーキテクチャを連携させる場合にはトランスレータ方式，異質なサービスアーキテクチャを連携させる場合にはプロキシ方式が適している．

例えば，スマート環境を相互接続して，Bluetooth マイクとサービスを連携する場合を考える．ある Bluetooth 機器に接続させる Bluetooth マイクを検索する，という場合であれば，Bluetooth のプロトコルは同じであるため，Bluetooth プロトコルを IP に変換してトンネルの中を通せば，Bluetooth マイクを発見・接続できる．一方，あるサービス (United Spaces や cogma，UPnP など) に接続したいマイクを検索して，Bluetooth マイクが見つかった，という場合は，より上位のレイヤで処理すべきであり，Bluetooth へのプロキシがあるほうが望ましい．

異種のサービスアーキテクチャの仲介を行うには，上記の 2 つの方式が考えられ，それらはサービスアーキテクチャの差異によって，適応的に選択されるべきである．今後，我々は，様々な異種サービスアーキテクチャの連携を実装し，最適な仲介方式について検討する．このときの実装は，汎用的な連携ではなく，もっともシンプルな連携方式を選択していくことで，上記の仮説を検証する．

7 適応的な異種スマート環境の連携

本章では，異種スマート環境を，動的かつ効率的に連携させる，つまり適応的な，連携手法について述べる．4 章から 6 章では，異種スマート環境を接続・連携させることを主目的としていた．本章では，それに加えて，動的に連携させること，異種スマート環境の接続をスケールさせること，に注目する．

7.1 適応的な仲介方法の選択

異なるスマート環境に属する Smart Peer 同士を連携させる方法はケースバイケースであるが，4 章から

6 章の実装を行うことで，あるケースではどのような仲介が最適であるかを我々は把握している．

我々は，各レイヤの差異を一つずつ順番に解決していく方法はシンプルではなく，オーバーヘッドが大きくなると予測している．

同種のサービスアーキテクチャを連携させる場合には，トンネルを構築して同じネットワークにいるように見せることが，最もシンプルである．異種のサービスアーキテクチャを連携させる場合には，6.2 で述べたように，トランスレータ方式とプロキシ方式が選択可能である．また，このようにそれぞれのスマート環境と，サービスアーキテクチャの違いを判別することで，翻訳を行うべきレイヤを選択することができる．

今後は，Smart Peer がどのような連携をしようとしているかを判断し，最適な仲介方法を選択するためのシステムを検討する．

7.2 Semantic P2P Network

スマート環境が多数存在し，それらが相互に連携する場合でも，仲介者は仲介機能を果たさなくてはならない．3 章で述べたように，仲介機能を Semantic P2P Network で実現する手法を検討する．具体的には，4 章から 6 章で実装する，仲介の機能を分散化させる．

5 章では，ユーザ，場所，サービスの検索システムの分散化を達成する．スマート環境の通信の接続性 (4 章) と P2P ネットワークによる検索システム (5 章) を連結することで，オブジェクトの検索結果を受けてから，通信のトンネルを確立する，シーケンスが実現できる．

6 章では，プロトコル変換と翻訳の方式について述べた．これらの翻訳機能をモジュール化し，Semantic P2P Network に配置する．Semantic Peer には翻訳可能なプロトコルを設定し，5 章の検索システムに登録することで，各 Semantic Peer の持つ翻訳モジュールを，Semantic P2P Network で利用可能にする．

8 まとめ

本稿では，異種スマート環境の連携を行うための，Secure Semantic Tunneling のコンセプトについて述べ，それを実現するための設計として P2P ネットワークを採用することの利点について述べた．また，Secure Semantic Tunneling を実現するための設計について，問題を分割して述べた．

我々は，本稿で述べた Secure Semantic Tunneling モデルによって，異種スマート環境間の差異を乗り越えてセキュアな相互接続・サービス連携を可能にする基盤技術が開発できるものと考えている．

今後は，本稿で述べた設計を実装し，Secure Semantic Tunneling モデルと仮説を評価する．具体的には，United Spaces と cogma による異種サービスアーキテクチャの接続を行う．また，各種レイヤ，プロトコルの変換を実現する様々なパターンを実装し，Secure Semantic Tunneling を実際に利用可能にしていく．

参考文献

[1] Yu Enokibori, Nobuhiko Nishio “A Secure Extension for Realizing Multi-Institutional Ubiquitous Service System”, UCS2004 (2004)

[2] 河口 信夫，春原 雅志，“WebCodget : Web サーバに移動して動作する組込み機器向け移動ソフトウェア” 情報処理学会ユビキタスコンピューティング研究会/映像情報メディア学会技術報告, 2005-UBI-2(8) (2005)

[3] Nobuo Kawaguchi “VPcogma: A Light-Weight Cooperative Middleware for Ubiquitous Embedded Devices” International Workshop on Software Architectures for Self-Organization with Pervasive2005 (2005)

[4] Diana Smetters, Dirk Balfanz, Glenn Durfee, Trevor Smith, KyungHee Lee ”Instant Matchmaking: Simple, Secure Virtual Extensions to Ubiquitous Computing Environments” UbiComp2006 (2006)

[5] 中野悦史，河口信夫，西尾信彦 “IPv6 Everywhere: IPv6 接続のための適応的トンネル構成機構” IC2006 (2006)

[6] Jin Nakazawa, Hideyuki Tokuda, W. Keith Edwards, Umakishore Ramachandran “A Bridging Framework for Universal Interoperability in Pervasive Systems” ICDCS’06 (2006)

[7] Jacob Woltersdorf, Nobuhiko Nishio “A Peer-to-Peer Overlay Network for Cooperating, Distributed Smart Environments” (2006)

[8] 伊藤誠悟, 河口信夫, “Locky.jp：無線 LAN を用いた位置情報・測位ポータル”, 情報処理学会研究会報告 (第 34 回モバイルコンピューティングとユビキタス通信研究会), 2005-MBL-034, pp.24-31 (2005)

[9] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, Jeremy Lilley “The Design and Implementation of an Intentional Naming System (INS)“ SOSP 1999 (1999)

[10] 永田 智大, 西尾 信彦, 徳田 英幸 “ASAMA: 適応的なサービス利用管理機構” 情報処理学会 論文誌, Vol.42 (6) 2001 年 6 月 pp.1557-1569 (2001)

[11] D. A. Bryan and C. Jennings, ”A P2P Approach to SIP Registration and Resource Location”, Internet-Draft draft-bryan-sipping-p2p-01 (2005)

[12] Skype http://skype.com/

[13] Place Lab http://www.placelab.org/