資料３－１４

方式の比較の際の「評価軸」について

情報連携基盤技術 WG 構成員　崎村夏彦

ゲートウェイとアクセストークン方式の比較をしっかりしようとするならば、まずは適切な評価軸の整理から行わなければならないと思います。アドホックに、この点は、あの点は、とやることにはあまり意義を感じません。もしろ、これらの評価軸を定め、それらを要件化する事のほうが重要であると考えます。なぜなら、大まかにゲートウェイ方式、アクセストークン方式と言っても、その内容には色々あり、また技術の進歩も急速ですから、現時点で「○○の方式が良い」のような記述にすることは、将来の可能性の芽を積んでしまうからであります。

なお、方式の比較に関しては、法的側面と技術的側面について分類して軸の整理を行うべきであると考えます。

以下は、そうした軸の例です。（あくまでも例示ですので、全体をカバーしているものではありません。）

法的側面

- 国家による一元管理（情報の集中把握）に対する耐性
  - ある一機関や一システムを制御することによって一元管理ができるようになっているものは、複数を制御しなければいけないものに比べて危険です。制御対象が増えれば増えるほど安全になります。
  - また、一機関を制御することによって、他の機関に気取られることなく情報を収集できるようなポイントがあるものは、無いものに比べて非常に危険になります。
- アカウンタビリティ確保／監査の容易性
  - 監査の際の大きなポイントは、複数機関のログを付けあわせて、その整合性をチェックすることにあります。したがって、情報の出し手と受け手で独立してログを取れるようになっているものはそうでないものに対して優位性があることになります。

技術的側面

- スケーラビリティ

    - 処理が集中するボトルネックがあるものは、無いものに比べて不利になります。
    - 比較対象両方に処理が集中するポイントが有った場合、それぞれの負荷を比較する必要があります。
    - 拡張の用意さという意味のスケーラビリティの観点では、新たなサービス追加が既存のシステムに与えるインパクトが小さいほうが優位性があることになります。
- シングルポイント
    - あるサーバ／システムが落ちると、全体が止まってしまうようなものは、一部しか止まらないものに対して不利です。
- プライバシー向上策について
    - 独立した2つの情報漏えい事故によって取得された情報から、第三者がデータの付け合せを行って望まない自己像を生むことがどのくらい困難になっているか。例えば、仮名（リンクコード）方式のほうが共通コード方式よりも付け合せは困難（不可能では無い。属性の付け合せで付け合せも可能だろう）なので、有利となります。
    - 任意の2機関の結託による、望まない自己像の生成がどのくらい困難か。例えば、共通コード方式に比べて、仮名方式は有利です。（シーケンスアタックや、属性にヒント情報を与えるアタックなどが考えられますが、共通コード方式に比べて手数が増える分アタックコストが上がり、それだけ起きる確率が減少します。）
    - 必要最低限の情報しか連携しないようにすることの容易性。例：例えば、個人ごとに必要な属性項目や、属性項目を評価した結果（例：20 歳以上か）のみを連携できるようになっているものは、そうでないものに対して有利です。
- サービスの移転の容易性／囲い込み耐性
    - ある製品を使っていたときに、他の製品に乗り換えるのがどのくらい容易か。（データの取り出しの容易性、データの標準化度合いなど。）これが容易であれば、ベンダーによる囲い込みに対する耐性がある程度確保されるため、競争による費用の軽減が期待されます。
    - 同じプロトコルを使ったオープンソース実装があるもののほうが無いものよりも有利です。
    - オープンプロトコルによらないものは囲い込みを行われるリスクが高いので避けるべきです。
- （デジュール／デファクト）標準化度

    - 標準化されているものはされていないものに比べて、多くの人のレビューを受けているか、多くの製品が提供されるか（＝費用が安くなる、囲い込み耐性が上がる）などの優位性を持っています。
- 技術トレンドとの合致性
    - 技術は日進月歩です。技術トレンドを見極めて、合致させていかないと、ガラパゴス化したシステムに囲い込まれてしまいます。したがって、その時点で技術トレンドに乗っているものはそうでないものに対して有利です。
    - 最近は、コンシューマ向け技術が後からエンタープライズに適用されるというのがトレンドになってきています。この点では、コンシューマ向けサービスに広く実装されているもののほうが有利ということになります。
- サービス追加の容易性
    - 疎結合なものは密結合なものに比べて、既存システムに対する影響が少ないので、導入時の費用が安くなります。したがって、密結合なシステムに比べて優位です。
- 接続前テストについて
    - 対向でひとつのテストシステムに対してテストすれば、全体に対してテストが行われたのと同じになるような、十分に標準化されたシステムは、そうでないものに対して優位です。
    - 疎結合なシステムは、追加部分のみのテストで事足りるので、密結合なシステムに対して優位です。
- 調達の容易さ
    - 商用オフザシェルフ（COTS）は、カスタムシステムに比べて費用面でも導入期間の面でも優位です。
- プロトコルのシンプルさ、エレガントさ
    - プロトコルは常に競争にさらされていますが、シンプルなプロトコルは利用が広がりやすいので生き残る可能性が高いです。
    - なお、ここで言うシンプルさとは、非技術者向けの見え方ではなく、技術者向けのものです。オープンソースで人気のあるプロトコルはそれだけシンプルであると言えましょう。
- オープンソース実装の有無
    - 同じプロトコルを採用しているオープンソース実装があるもののほうが、安心して採用できます。

なお、こうした軸出しは、インターネット上でオープンに行うのが有効であると考えますので、ご検討いただければと存じます。

2 方式の比較の例

軸の整理がきっちり出来る前の段階での比較には疑問が残りますし、方式の中でもより詳細な個別の方式に依存する（特に技術的側面については）ので、現段階での比較にどれだけ意味があるかは不明です。以下は上記を適用した場合のひとつの例として、いくつかの項目について記載しています。

法的側面

	ゲートウェイ方式	アクセストークン方式
国家による一元管理（情報の集中把握）に対する耐性	ゲートウェイを掌握するだけで、他の組織に気取られることなく、全情報を取得することができます。したがって、国家による一元管理に対する耐性は低いと考えられます。	全機関を掌握するか、あるいは、集約サービスを新たに作って、そこに各機関からの情報を流し込む必要があります。しかし、これは秘密裏に行うことは極めてこんなんです。したがって、ゲートウェイ方式よりは耐性が高いと思われます。
アカウンタビリティ確保／監査の容易性	ゲートウェイでのログの他に、各機関でのログを取りそれらを付けあわせることで整合性のチェックができます。ただし、ゲートウェイの中で情報が抜かれている可能性を排除するためのログを複数機関で整合性チェックすることはできないので、その点がやや弱いと言えるでしょう。一方、情報の出し手と受け手が結託しても、ゲートウェイが結託していなければ、ログの整合性が取れなくなるというメリットはあります。（が、その場合、それ以前に、情報連携基盤を通さずに連携するでしょうが。）	データの出し手と受け手で必ずペアになってログが残るので、整合性のチェックができます。ゲートウェイが存在しないので、中抜きの危険はなく、その点ではアカウンタビリティの確保が行い易いです。

技術的側面

	ゲートウェイ方式	アクセストークン方式
スケーラビリティ	処理がゲートウェイに集中するので不利です。 密結合なので、新規サービスの追加という意味でも不利です。	ゲートウェイが無いので有利です。 疎結合なので、新規サービスの追加という意味でも有利です。
シングルポイント	ゲートウェイが落ちると、すべての通信が止まります。	アクセストークンを発行するサーバが落ちても、過去に発行した有効なトークンを使って連携を続行できます。
プライバシーの向上策について	仮名方式をとれば、情報漏洩による望まない自己像の形成はある程度困難になります。 内部の 2 機関の結託については、シーケンスアタックや、ステータスアタックが可能です。後者については、ゲートウェイで補足できるのではという考えもありますが、データが暗号化してある前提のもとではゲートウェイでは検出できません。（暗号化していない場合、国家管理の問題が出ます。） 必要最低限の情報連携という側面は、採用される具体的なプロトコル依存です。	仮名方式をとれば、情報漏洩による望まない自己像の形成はある程度困難になります。 内部の 2 機関の結託については、シーケンスアタックや、ステータスアタックが可能です。 必要最低限の情報連携という側面は、採用される具体的なプロトコル依存です。
サービスの移転の容易性／囲い込み耐性	筆者が不勉強なため、ゲートウェイモデルの情報連携の国際標準がわかりませんので、評価できません。	SAML/OAuth などの国際／デファクト標準を利用し、データフォーマット（スキーマ）も標準化されたものを使えば、囲い込み耐性は高いと言えます。（もとより、採用する具体的技術によります。） オープンソース実装も多数あります。
（デジュール	同上	SAML/OAuth など、多数の標準化

／デファクト）標準化度		されたオプションが存在します。実装も多数存在します。
技術トレンドとの合致性	インターネット時代においてはゲートウェイ方式はやや不利なようです。	最近では、OAuth ベースのプロトコルの勢いが非常に盛んです。
サービス追加の容易性	新たなサービスを追加した際に、必ずゲートウェイを通過することになるので、既存サービスに対する影響が心配されるため、慎重な運用が求められます。	疎結合なので、既存サービスに対する影響は最小限に留められます。そのため、サービスの追加は容易です。
接続前テストについて	具体的にどのようなプロトコルになるかに依存するので確定的には言えませんが、ゲートウェイとの間での試験をやれば足りるように感じられます。 但し、新規サービス追加の場合には、影響度テストが再度必要になります。	試験用のサービスに対するテストを行えば足りるので、ゲートウェイと同じと思われます。標準化されたものは、テストサービス自体も供給されていたりするので便利です。 新規サービスの追加も、原則追加部分だけのテストで事足ります。 （これがダメな前提だと、そもそもインターネットが機能しなくなります。）
調達の容易さ	標準化度合いがわからないので、評価できません。	標準化されているので、製品が多数あります。また、オープンソースもありますので、調達は比較的容易であると考えられます。商用オフザシェルフ（COTS）は、カスタムシステムに比べて費用面でも導入期間の面でも優位です。
シンプルさ・エレガントさ	内容がわからないので評価できません。	プロトコルによります。
オープンソースの有無	同上	どのプロトコルにも複数存在します。