# 【2】 パーミッションコントロール実証試験概要

「症例登録を踏まえた病院共通のコンピュータシステム開発とコストに関する研究」

# パーミッションコントロール実証実験概要

2007年1月19日

# 目次

# 1. パーミッションコントロールとは

## 1. 1. マルチコードプラットフォーム

「マルチコードプラットフォーム」とは、複数のコード体系／アーキテクチャが混在する環境下でやり取りされる情報の相互運用を可能とするためのプラットフォームであり、「ユビキタス時代の社会プラットフォーム」を支える技術の一つである。

現在普及しているバーコードに加え、情報量が増えて認識することが可能な、二次元バーコード、RFID、センサーなど今まで取れなかったような情報を把握することが出来、複数のシステムがつながるようになってきている。しかし、コード体系の標準化は進んでいるものの、デファクトなもの、独自のものが混在しており、技術の普及の妨げになっている。このような議論の中から

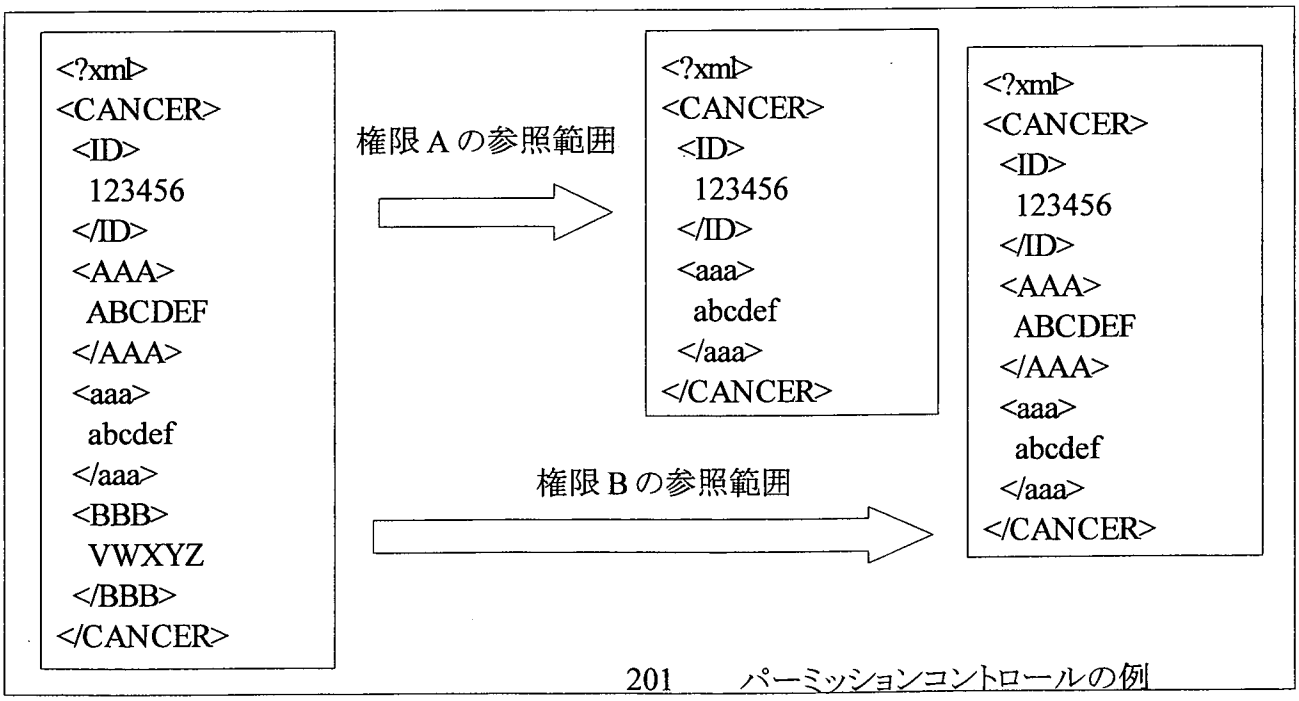
- 使うコード体系が複数存在しても大丈夫
- 使う(コードを記載する)技術が複数存在しても大丈夫

ということを世の中に示すために、経済産業省 平成18年度「エネルギー使用合理化電子タグシステム開発調査事業」として構築され、Tyzohコミュニティ(http://tyzoh.jp)よりオープンソースとして公開されている。

## 1. 2. パーミッションコントロール

「パーミッションコントロール」は平成18年度の実証実験の考察の中から、昨今の情報セキュリティの観点からプラットフォームに入ってくる"**データへの参照を誰に許可するかは、データの提供者の意思でコントロール可能にするべき**"という問題を解決するために、「マルチコードプラットフォーム」の拡張機能として開発中の技術である。

平成19年現在、プラットフォームに入ってくるデータのうちXML形式で投入されるものについて、XML文書の情報毎(element毎)に提供者がアクセス権限を設定できる機能を構築している。




201　パーミッションコントロールの例

# 2. パーミッションコントロール システム検証範囲

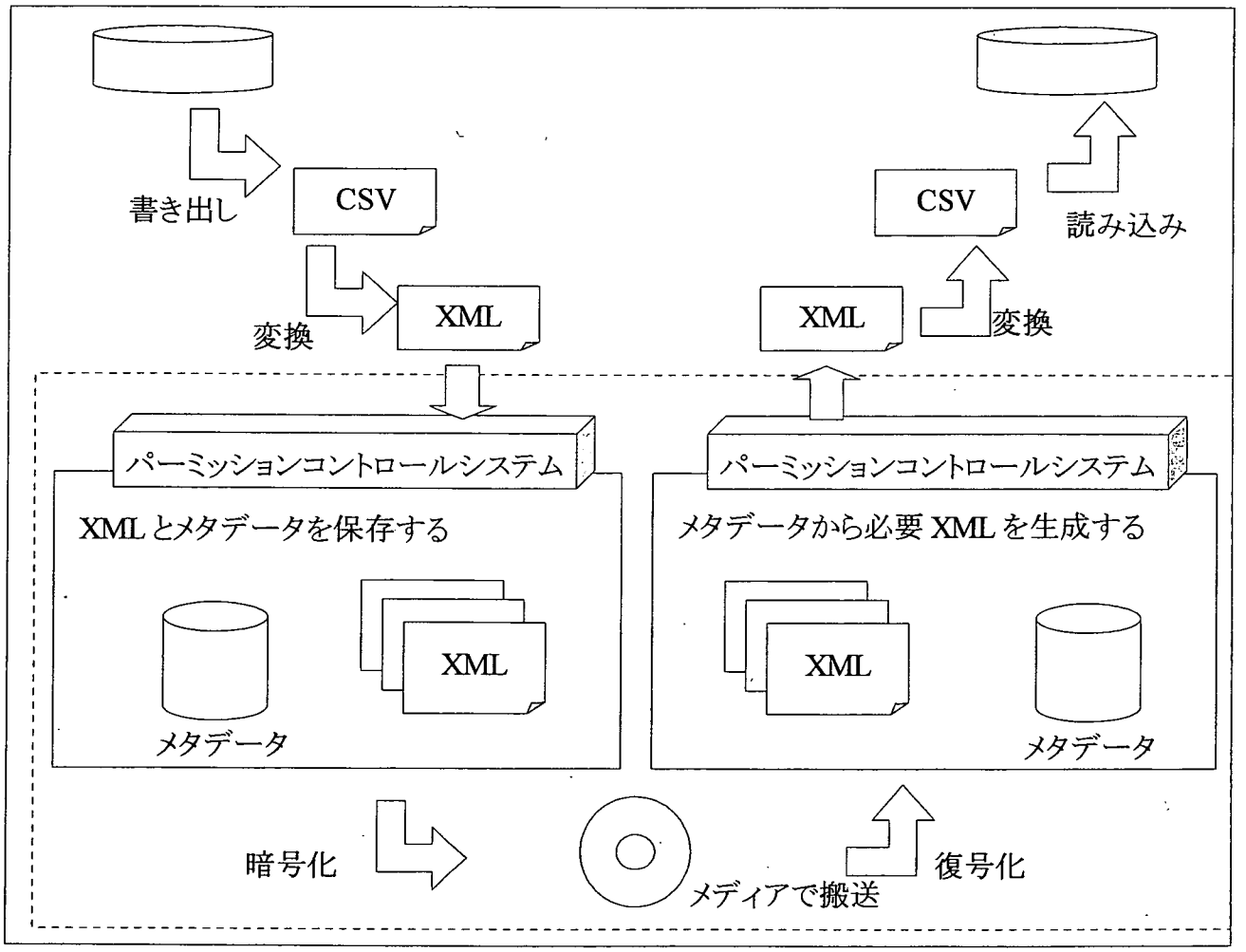
## 2. 1. がん臨床研究事業における検証条件

厚生労働省 がん臨床研究事業「症例登録を踏まえた病院共通のコンピュータシステム開発とコストに関する研究」においてパーミッションコントロールを以下の条件で検証する。

- HosCan-R システムより出力された CSV ファイルを 変換した XML ファイルをインプットとする。
- ユーザの管理に関しては今回の検証では、固定的なものとする。
- セキュリティの観点からデータの転送は行わず、CD(メディア)による搬送とする。

## 2. 2. がん臨床研究事業における検証範囲

厚生労働省 がん臨床研究事業「症例登録を踏まえた病院共通のコンピュータシステム開発とコストに関する研究」における検証範囲を図示する。

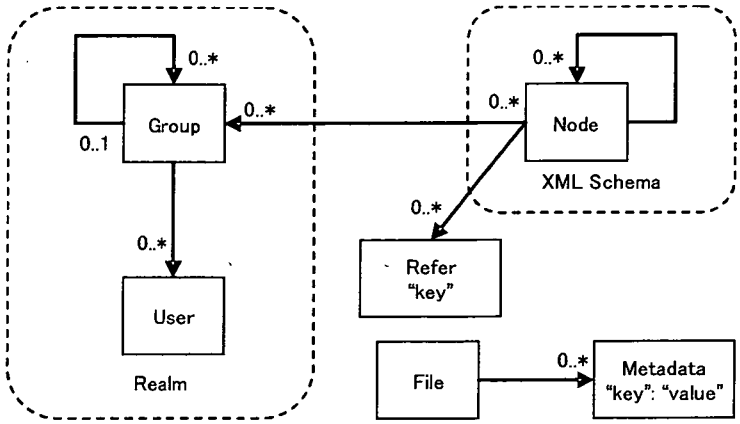


## 2.3.がん臨床研究事業における検証内容

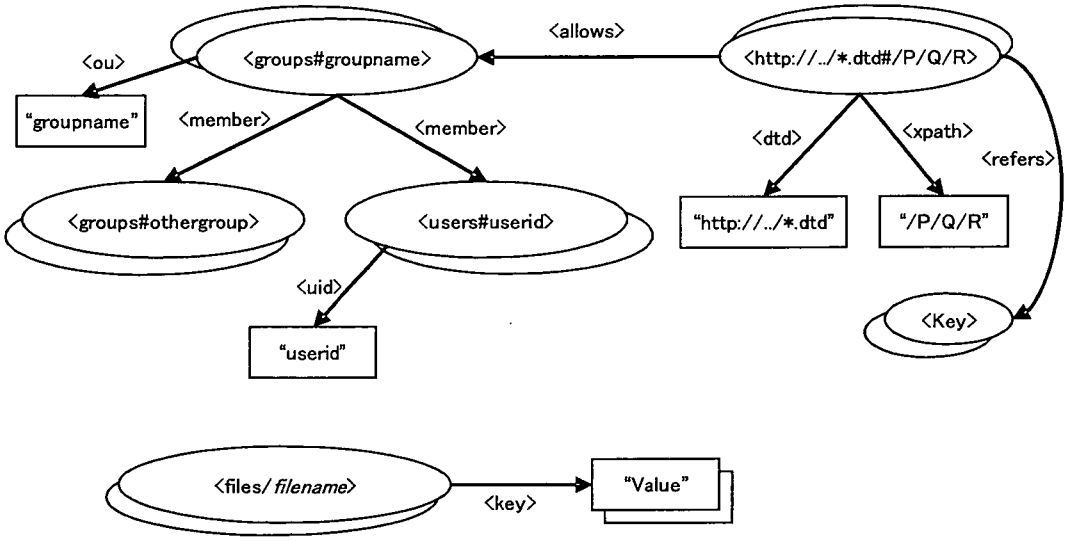
- XML データとメタデータをパーミッションコントロールシステムに入力、保存する。
- データすべてを暗号化し CD(メディア)で搬送する。
- データを複合化し、設定したパーミッションに従った XML データを出力する。

# 3. データモデル

## 3. 1. Class Diagram of XML ACL




## 3. 2. RDF Schema Diagram of XML ACL




## 3. 3. XML Schema について

Node は次のものにより、ユニークに識別される

- ファイル種別 (典型的には DTD URL)
- Node を示す、ルートからの、単純な XPATH ("/ROOT/DATA/NAME" など)

### 3. 3. 1. Realm について

特殊な Group として次のようなものを設ける。

- Refer("name"): File に付与された、メタデータ "name" を参照

### 3. 3. 2. ACL について

ひとつの Statement には、次のものが含まれる。

- Subject: XML Node 名、もしくは、File 名
- Group: Group 名 (UserID を直接指定しても良い。)

[Statement の例] "http://host/dir/schema.dtd#/root/data/personal" Refer("Creator")

- XML Schema と File は独立であることに注意。(特定の File の中のとある TAG について Statement を指定することは出来ない。)
- 同一の Subject について、複数の Statement が存在する場合、それには適用順序はない。

### 3. 3. 3. パーミッションの有無の計算アルゴリズム

- これから行いたい、User を決定 (引数などで渡す)
- ルートから深さ優先でタグを手繰る。
- マッチする Statement が見つからなかったら、それ以下のタグ(First Child)は手繰らず、次のタグ (Next Sibling or Parent)に移る。
- あるノードの Statement がひとつも付いていない場合、次のような仮定が行われる。
- ルートノードの場合は、Everyone が仮定される。
- 中間のノードの場合は、上位(Parent)のノードのパーミッションが仮定される。

## *3. 4. URI について*

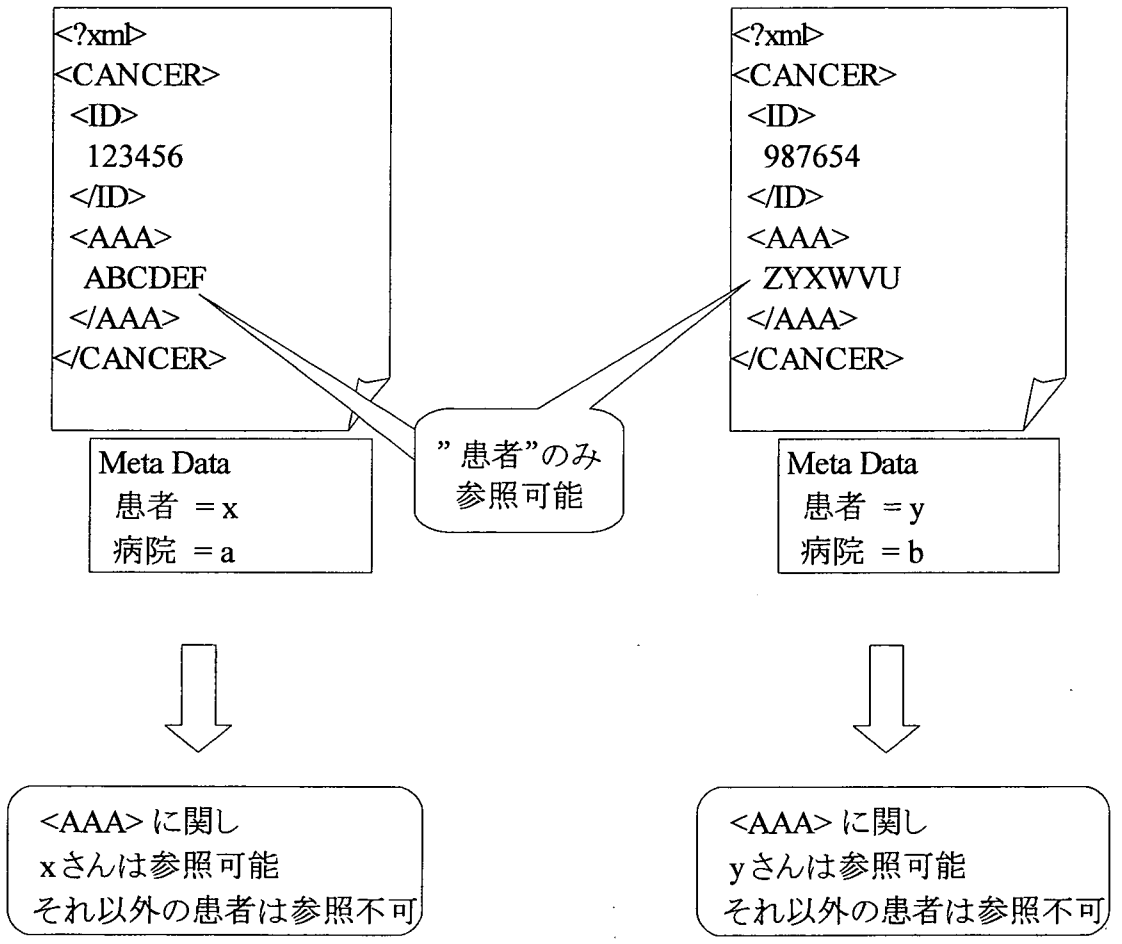
| URI 省略形 | 正式 URI |
|---|---|
| <ou> | <ftp://ftp.rfc-editor.org/in-notes/rfc1617.txt#attributes/ou> |
| <member> | <ftp://ftp.rfc-editor.org/in-notes/rfc2256.txt#attributes/member> |
| <uid> | <ftp://ftp.rfc-editor.org/in-notes/rfc2798.txt#attributes/uid> |
| <groups#*groupname*> | <ldap://www.pref.chiba.jp/ou=*groupname*,o=www.pref.chiba.jp> |
| <users#*userid*> | <ldap://www.pref.chiba.jp/uid=*userid*,o=www.pref.chiba.jp > |
| <allows> | <http://www.tyzoh.jp/mcode/permission/RDFterms#allows> |
| <dtd> | <http://www.tyzoh.jp/mcode/permission/RDFterms#dtd> |
| <xpath> | <http://www.tyzoh.jp/mcode/permission/RDFterms#xpath> |
| <refers> | <http://www.tyzoh.jp/mcode/permission/RDFterms#refers> |
| <*key*> | <http://www.ncc.go.jp/cancer/karte/RDFterms#*key*> |
| <files/*filename*> | <http://www.pref.chiba.jp/byouin/gan/files/*filename*> |

## 3. 5. メタデータについて

このパーミッションコントロールシステムではデータ投入時に参照権限を設定するが、この設定方法が複雑であると実用に向かない技術になってしまう。このため、メタデータによるコントロールを行うことにした。

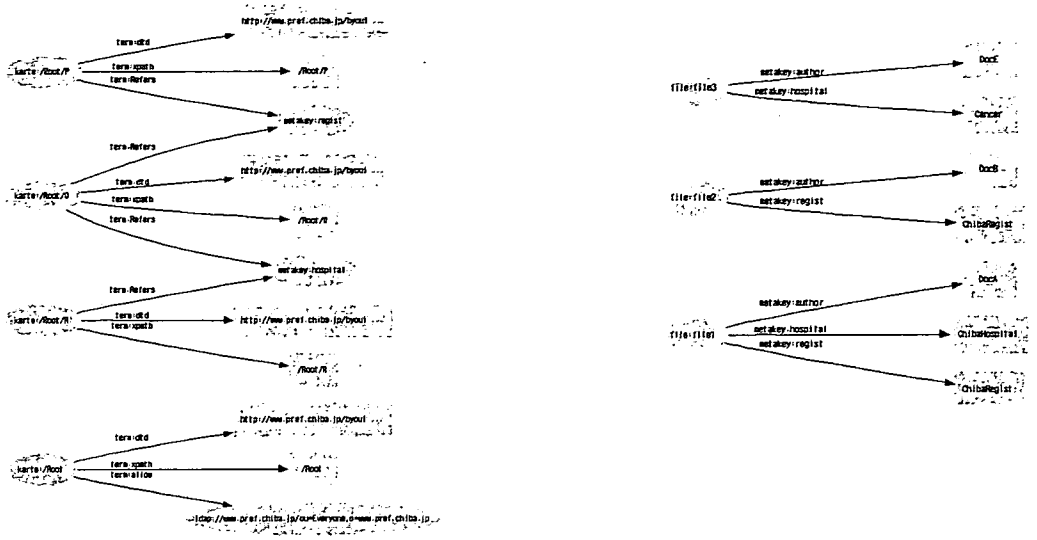
例えば、コントロール上は"患者"という指定が可能であるが、実際にはデータ(XML)毎にメタデータが付与されており、カルテAの"患者"はxさん、カルテBの"患者"はyさんというようなコントロールを行う。

この方式を使用することで、文書(XML)に対する複雑なパーミッションコントロール設定情報とメタデータを分離することが可能になり、個々の文書(XML)に対しての設定はメタデータとして持たせればよいことになる。



# 4. データサンプル

file:///C:/DOCUME~1/kuwaki/LOCALS~1/Temp/B2Temp/Attach/02.gif



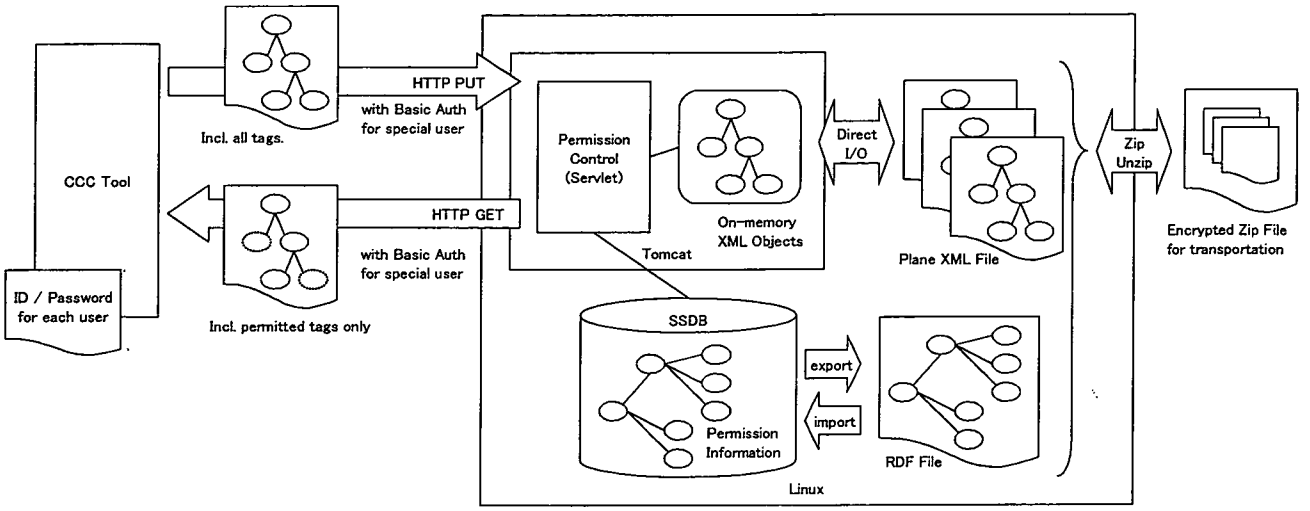

RDF例

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rfc1617="ftp://ftp.rfc-editor.org/in-notes/rfc1617.txt#attributes/"
 xmlns:rfc2256="ftp://ftp.rfc-editor.org/in-notes/rfc2256.txt#attributes/"
 xmlns:rfc2798="ftp://ftp.rfc-editor.org/in-notes/rfc2798.txt#attributes/"
 xmlns:term="http://www.tyzoh.jp/rncode/permission/RDFterms#"
 xmlns:karte="http://www.pref.chiba.jp/byouin/gan/karte.dtd#"
 xmlns:metakey="http://www.ncc.go.jp/cancer/karte/RDFterms#"
 xmlns:file="http://www.pref.chiba.jp/byouin/gan/files/"
>
 <rdf:Description rdf:about="karte:/Root">
  <term:dtd>http://www.pref.chiba.jp/byouin/gan/karte.dtd</term:dtd>
  <term:xpath>/Root</term:xpath>
  <term:allow rdf:resource="ldap://www.pref.chiba.jp/ou=Everyone,o=www.pref.chiba.jp"/>
 </rdf:Description>
 <rdf:Description rdf:about="karte:/Root/P">
  <term:dtd>http://www.pref.chiba.jp/byouin/gan/karte.dtd</term:dtd>
  <term:xpath>/Root/P</term:xpath>
  <term:Refers rdf:resource="metakey:regist"/>
 </rdf:Description>
 <rdf:Description rdf:about="karte:/Root/Q">
  <term:dtd>http://www.pref.chiba.jp/byouin/gan/karte.dtd</term:dtd>
  <term:xpath>/Root/Q</term:xpath>
  <term:Refers rdf:resource="metakey:regist"/>
  <term:Refers rdf:resource="metakey:hospital"/>
 </rdf:Description>
 <rdf:Description rdf:about="karte:/Root/R">
  <term:dtd>http://www.pref.chiba.jp/byouin/gan/karte.dtd</term:dtd>
  <term:xpath>/Root/R</term:xpath>
  <term:Refers rdf:resource="metakey:hospital"/>
 </rdf:Description>
 <rdf:Description rdf:about="ldap://www.pref.chiba.jp/ou=Everyone,o=www.pref.chiba.jp">
  <rfc2256:member>
   <rdf:Description rdf:about="ldap://www.pref.chiba.jp/ou=ChibaRegist,o=www.pref.chiba.jp">
    <rfc1617:ou>ChibaRegist</rfc1617:ou>
    <rfc2256:member>
     <rdf:Description rdf:about="ldap://www.pref.chiba.jp/uid=DocA,o=www.pref.chiba.jp">
      <rfc2798:uid>DocA</rfc2798:uid>
     </rdf:Description>
    </rfc2256:member>
    <rfc2256:member>
     <rdf:Description rdf:about="ldap://www.pref.chiba.jp/uid=DocB,o=www.pref.chiba.jp">
      <rfc2798:uid>DocB</rfc2798:uid>
     </rdf:Description>
    </rfc2256:member>
   </rdf:Description>
  </rfc2256:member>
  <rfc2256:member>
   <rdf:Description rdf:about="ldap://www.pref.chiba.jp/ou=ChibaHospital,o=www.pref.chiba.jp">
    <rfc1617:ou>ChibaHospital</rfc1617:ou>
    <rfc2256:member rdf:resource="ldap://www.pref.chiba.jp/uid=DocB,o=www.pref.chiba.jp"/>
    <rfc2256:member>
     <rdf:Description rdf:about="ldap://www.pref.chiba.jp/uid=DocC,o=www.pref.chiba.jp">
      <rfc2798:uid>DocC</rfc2798:uid>
     </rdf:Description>
    </rfc2256:member>
   </rdf:Description>
  </rfc2256:member>
```

```
  <rfc2256:member>
   <rdf:Description
 rdf:about="ldap://www.pref.chiba.jp/ou=FukuokaRegist,o=www.pref.chiba.jp">
    <rfc1617:ou>FukuokaRegist</rfc1617:ou>
    <rfc2256:member rdf:resource="ldap://www.pref.chiba.jp/uid=DocC,o=www.pref.chiba.jp"/>
    <rfc2256:member>
     <rdf:Description rdf:about="ldap://www.pref.chiba.jp/uid=DocD,o=www.pref.chiba.jp">
      <rfc2798:uid>DocD</rfc2798:uid>
     </rdf:Description>
    </rfc2256:member>
   </rdf:Description>
  </rfc2256:member>
  <rfc2256:member>
   <rdf:Description
 rdf:about="ldap://www.pref.chiba.jp/ou=FukuokaHospital,o=www.pref.chiba.jp">
    <rfc1617:ou>FukuokaHospital</rfc1617:ou>
    <rfc2256:member rdf:resource="ldap://www.pref.chiba.jp/uid=DocD,o=www.pref.chiba.jp"/>
    <rfc2256:member>
     <rdf:Description rdf:about="ldap://www.pref.chiba.jp/uid=DocE,o=www.pref.chiba.jp">
      <rfc2798:uid>DocE</rfc2798:uid>
     </rdf:Description>
    </rfc2256:member>
   </rdf:Description>
  </rfc2256:member>
  <rfc2256:member>
   <rdf:Description rdf:about="ldap://www.pref.chiba.jp/ou=Cancer,o=www.pref.chiba.jp">
    <rfc1617:ou>Cancer</rfc1617:ou>
    <rfc2256:member>
     <rdf:Description rdf:about="ldap://www.pref.chiba.jp/uid=DocF,o=www.pref.chiba.jp">
      <rfc2798:uid>DocF</rfc2798:uid>
     </rdf:Description>
    </rfc2256:member>
   </rdf:Description>
  </rfc2256:member>

 </rdf:Description>
 <rdf:Description rdf:about="file:file1">
  <metakey:author>DocA</metakey:author>
  <metakey:hospital>ChibaHospital</metakey:hospital>
  <metakey:regist>ChibaRegist</metakey:regist>
 </rdf:Description>
 <rdf:Description rdf:about="file:file2">
  <metakey:author>DocB</metakey:author>
  <metakey:regist>ChibaRegist</metakey:regist>
 </rdf:Description>
 <rdf:Description rdf:about="file:file3">
  <metakey:author>DocE</metakey:author>
  <metakey:hospital>Cancer</metakey:hospital>
 </rdf:Description>
</rdf:RDF>
```

# 5. パーミッションコントロール仕様(検証版)




## *5. 1. 外部仕様(HTTP によるXML ファイル操作)*

### 5. 1. 1. ユーザ認証

- 上記操作を行う際は、固定の特殊ユーザによる Basic 認証が必須。
- 特殊ユーザの ID と Password は、Servlet 側と CCC Tool 側で、あらかじめ共有しておく。

### 5. 1. 2. リスト

GET http://.../PermitCtrl

- ファイル名とそのメタデータが一行ずつ “filename?key1=val1&key2=val2 …” という形式で返す。

### 5. 1. 3. 追加

PUT http://.../PermitCtrl/*filename?key1=val1&key2=val2 ...*

- *filename* にファイル名を指定する。
- HTTP body に XML 追加したいファイルの内容を(エンコードせず)そのまま含める。
- メタデータを「キー=値」形式でパラメータ指定する。(複数指定可)
- 同一の *filename* が存在する場合、上書きされる。(メタデータも)
- 追加に関するパーミッションコントロールは Servlet では行わない。(CCC Tool 側で「追加」操作が出来るユーザを限定する。)

### 5. 1. 4. 削除

DELETE http://.../PermitCtrl/*filename*

- *filename* にファイル名を指定する。

- 削除に関するパーミッションコントロールは Servlet では行わない。(CCC Tool 側で「追加」操作が出来るユーザを限定する。)

#### 5. 1. 5. 取得

GET http://.../PermitCtrl/*filename*?user=*username*

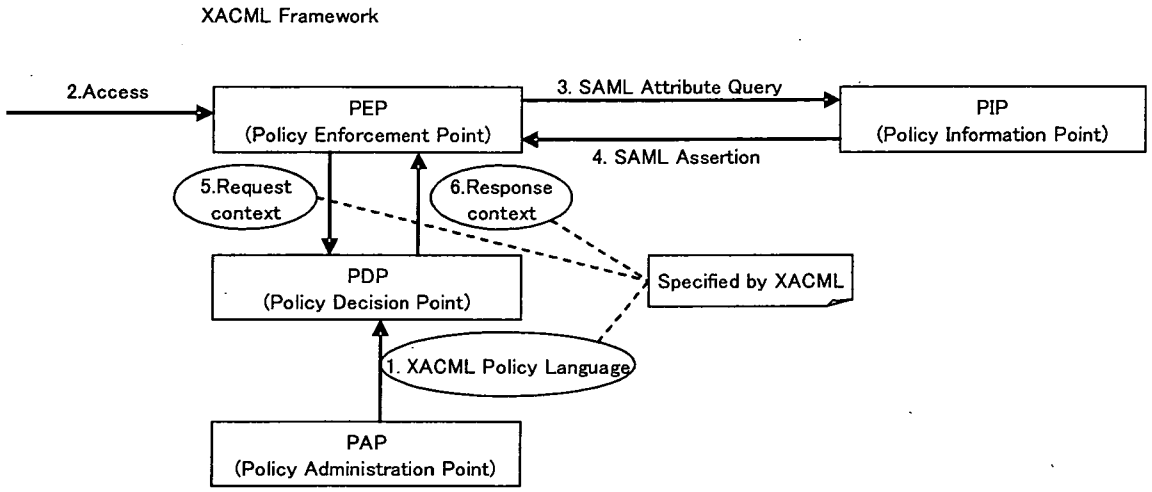
- パラメータ "user=" で実ユーザ ID を指定する。(Basic 認証で使用される特殊ユーザ ID ではないことに注意すること。)
- Permission が無いタグは空タグ (属性も無い。ただし、タグ自体は空タグとして残っていることに注意)

### *5. 2. パーミッションコントロール(検証版)のポイント*

- パーミッションに関係するデータは SSDB で持つ。
- パーミッションコントロールと XML データ蓄積に関係するプログラムは、すべて Linux 上に実装する。
- XML File は構造を持った状態で入出力/蓄積する。(フラットではない)
- XML データは DOM 形式などで直接メモリに持つ。(XML DB は使用しない)
- XML データの追加/取得は HTTP REST (Representational State Transfer)方式。
- XACML のフレームワークは流用するが、API は XACML にこだわらない。
- パーミッション情報は RDF ファイルを記述することで設定する。

# 6. (参考)XACMLについて

## 6. 1. データフロー




1. PAPは、PDPにポリシと呼ばれるルールの集合体を適用する。ポリシはXMLで記述される。このスキーマはXACMLで定められている。
2. あるリソースに対して、アクセスが発生する。
3. PEPは「誰がアクセスしてきたか」の情報(subjectのID)を元に、アクセス制御に必要な情報(属性)をPIPに問い合わせる。このプロトコルは任意である※。
4. PIPはリポジトリを参照し、「アクセス者の属性」「対象のリソースの属性」などをPEPに渡す。
5. PEPはこの情報を「要求Context」としてまとめ、PDPに渡す。
   要求ContextはXMLで記述される。このスキーマはXACMLで定められている。
6. PDPは1.で適用されているポリシと5.で渡された要求Contextを比較し、そのアクセスを認可してよいか採決を行う。
   採決結果(許可・不許可・判定不能など)を、「応答Context」としてまとめ、PEPに渡す。
   応答ContextはXMLで記述される。このスキーマはXACMLで定められている。
7. PEPは実際にアクセス制御を行う。

## 6. 2. 各コンポーネントの役割

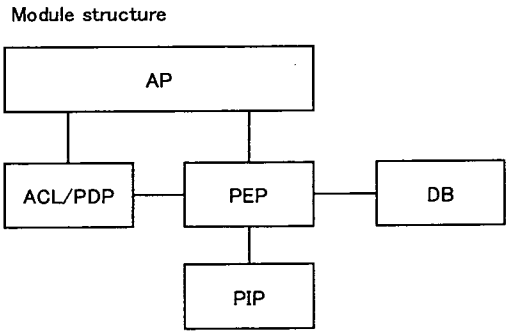
- PDP Policy Decision Point(ポリシ決定点)
  ポリシ(制御するためのルールの集合)をリクエストに対して適用し、そのアクセスの可否を決定し、それを「応答Context」として投げる。
- PAP Policy Administration Point
  リポジトリなどからポリシを呼び出し、ポリシを記述したXMLをPDPに渡す。
- PEP Policy Enforcement Point(ポリシ実行点)
  PIPよりアクセス制御に必要な情報を取得する。
  取得した情報を「要求Context」にまとめ、PDPに投げる。
  PDPが投げた「応答Context」元に、実際にアクセス制御を行う。
- PIP Policy Information Point
  アクセスしてきた人(Subject)の属性(役職、所属グループ等)、アクセス対象のリソースの属性(リソース管理者、公開許可など)、アクセス内容(読む、書きこむ、削除する等)の情報を、リポジトリから取得し、PEPに伝える。

## 6. 3. ポリシおよびポリシ適用の例

アクセス者のグループが Team1 で、「メンバ評定.xml」にアクセス要求があったとき、

1. 役職が Project Manager ならば、読取(read)と更新(write) を認可する。
2. 役職が Member であるならば、読取(read) のみができる。
3. どちらにも該当しないなら、アクセスは認めない。

これらを XACML のポリシ記述方式に従って記述しておき、PDP で実際のアクセス情報と比較し、認可するかしないかを判定する。PEP は PDP が出した判定結果に従い、実際のアクセス制御を行う。但し、この制御の方法は XACML では定められていない。



厚生労働省科学研究費補助金（がん臨床研究事業）による

「症例登録を踏まえた病院共通のコンピュータシステム開発と

コストに関する研究」

平成19年度報告書 （平成20年3月）

編 集　主任研究者 竜 崇正

発 行　千葉県がんセンター

千葉市中央区仁戸名町 666-2

TEL 043-264-5431

FAX 043-262-8680