

# IT資産管理関連タグの国際標準化動向

ソフトウェア資産管理評価認定協会

代表理事 高橋快昇

2016年6月10日

# 目次

- IT資産管理関連の国際標準化動向
- タグ関連規格の狙い
- ソフトウェア識別タグ
- 権利構造タグ
- 資源利用状況測定タグ
- まとめ

# ITAM関連の国際標準化動向

# IT資産管理国際規格 (ISO/IEC 19770) の動向

## ISO/IEC 19770-X シリーズ (ITAMの規格群)

概要

19770-5:2015 Overview & Vocabulary

無償ダウンロード可 <http://standards.iso.org/iso/19770/-5/>

プロセス

19770-1:201x (Rev. of 2012) Processes CD **初版JIS化済**

19770-8:201y

Guidelines for mapping of industry SAM practices with the 19770 family

19770-11:201y

Guidelines for the application of ISO/IEC 19770-1 for small organizations

情報構造  
(タグ)

**S/W Identification**

19770-2:2015 (Rev. of 2009)

S/W identification Tag **JIS化中**

**Entitlement**

19770-3:2016

S/W Entitlement Schema

19770-22:201x

Guide to cyber security

**Usage**

19770-4:201x CD

Resource utilization measurement

19770-7:201x Tag Management WD

凡例 :

出版済

開発中

計画中

Standard

Tech. Report

# タグ関連規格の狙い

# タグ標準化の狙い

ライセンス、セキュリティ、サービスでの徹底した高品質と省力化

## IT資産の特徴

- ✓ 頻繁に変更される
- ✓ 簡単に複製できる
- ✓ 利用場所と存在場所が異なっている
- ✓ 非常に数が多い



## ITを利用した自動化



- ✓ 同じ製品で異なる名称
- ✓ 同じ企業で異なる名称
- ✓ 情報のフォーマットバラバラ
- ✓ プラットフォーム毎にも異なる
- ✓ ツールで異なる結果

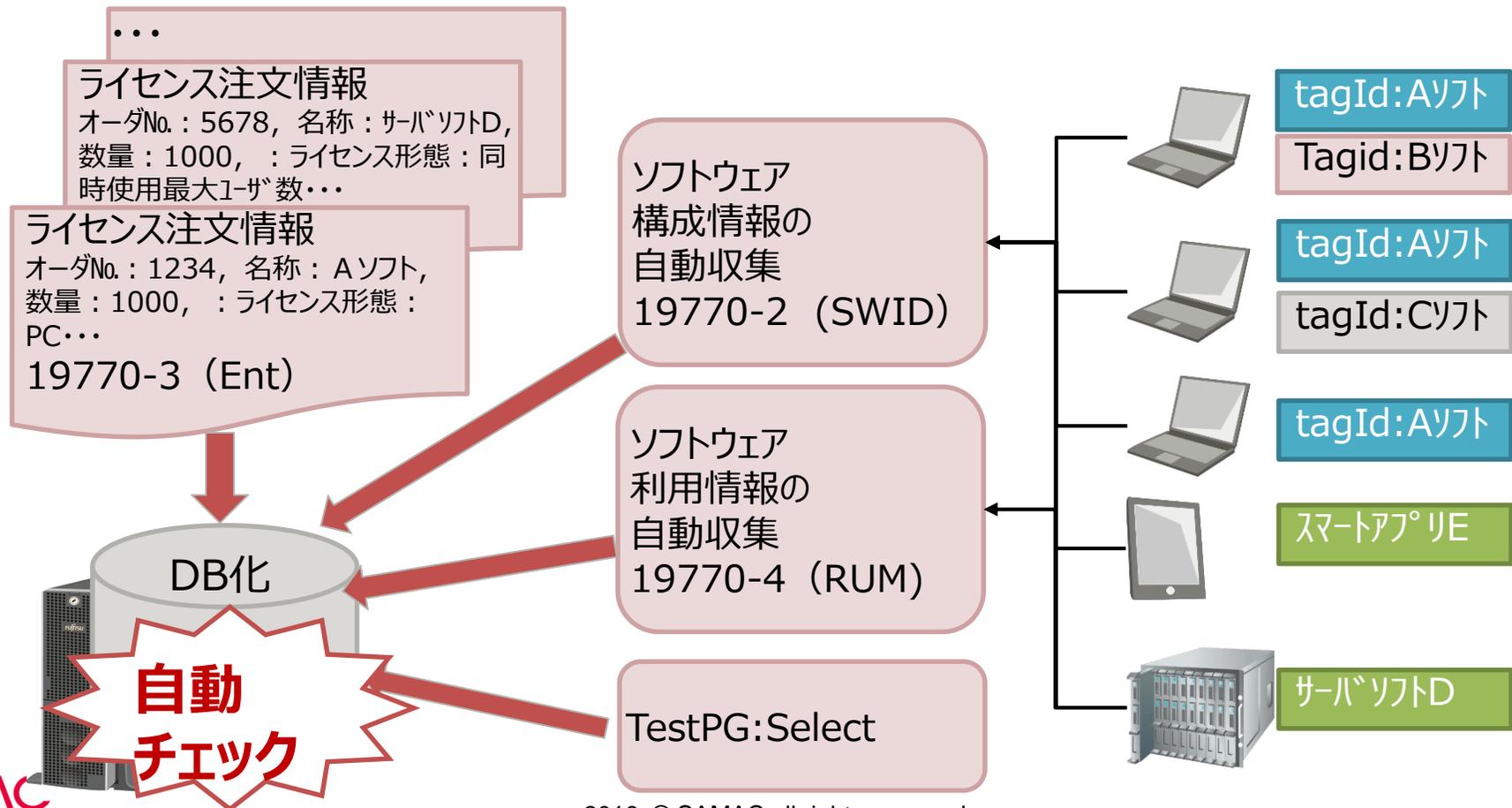
## 情報構造の標準化

(ISO/IEC 19770-2,-3,-4)



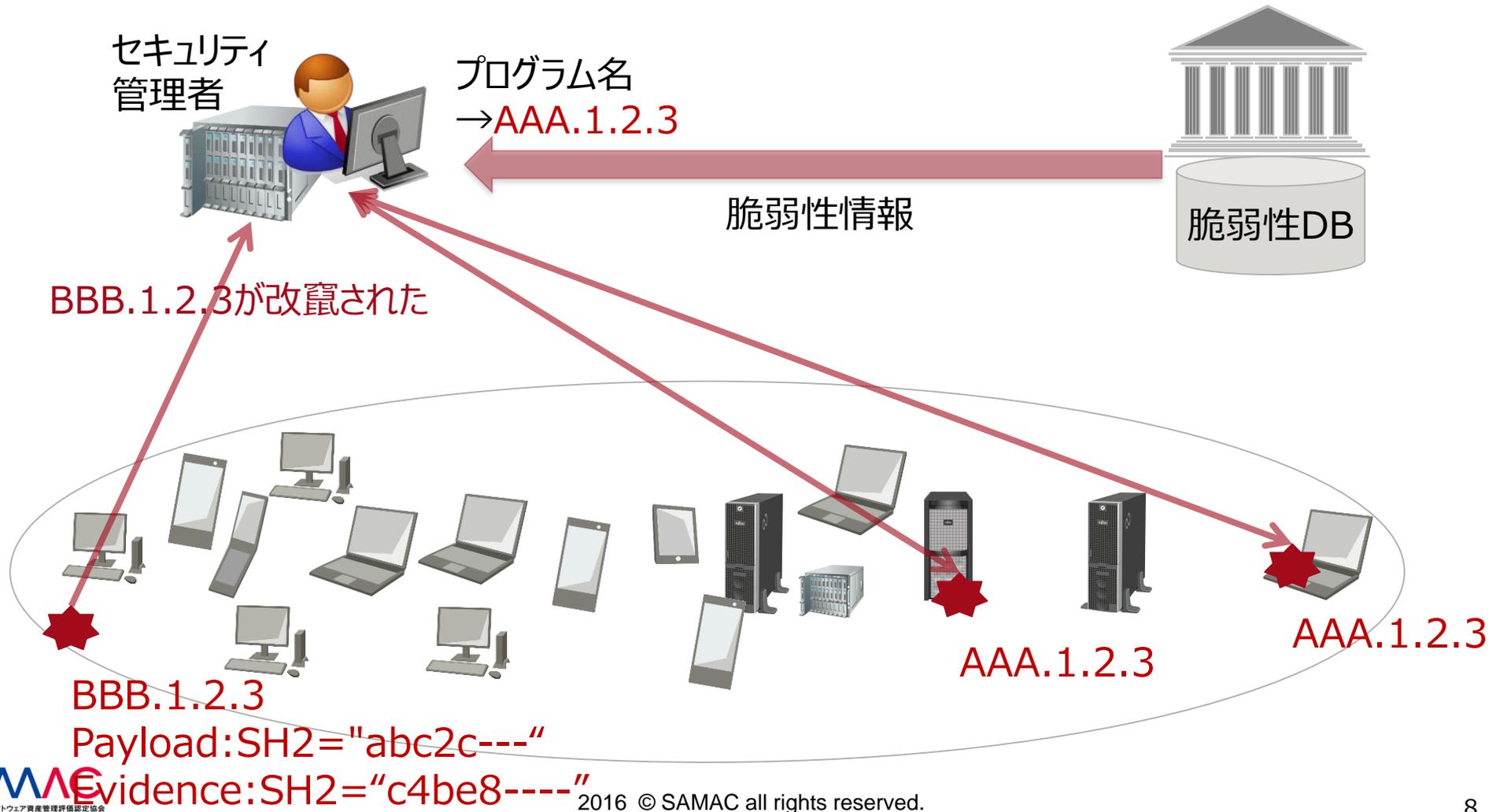
# ライセンスコンプライアンスでの例

- 権利情報は 1 9 7 7 0 - 3、ソフトウェア識別情報は 1 9 7 7 0 - 2、リソースの利用状況は 1 9 7 7 0 - 4 で情報構造が標準化される。
- 各種情報を集約し、自動チェックする。



# セキュリティでの例

世界中で報告される脆弱性情報から該当プログラムが直ちに見つかる。  
改竄されたプログラムが直ちに発見される。



# ソフトウェア識別タグ

## 19770-2 ソフトウェア識別タグの概要 (SWIDタグ : Software identification tagと呼ぶ)

ソフトウェア識別タグのXSDファイル :  
<http://standards.iso.org/iso/19770/-2/2015/schema.xsd>

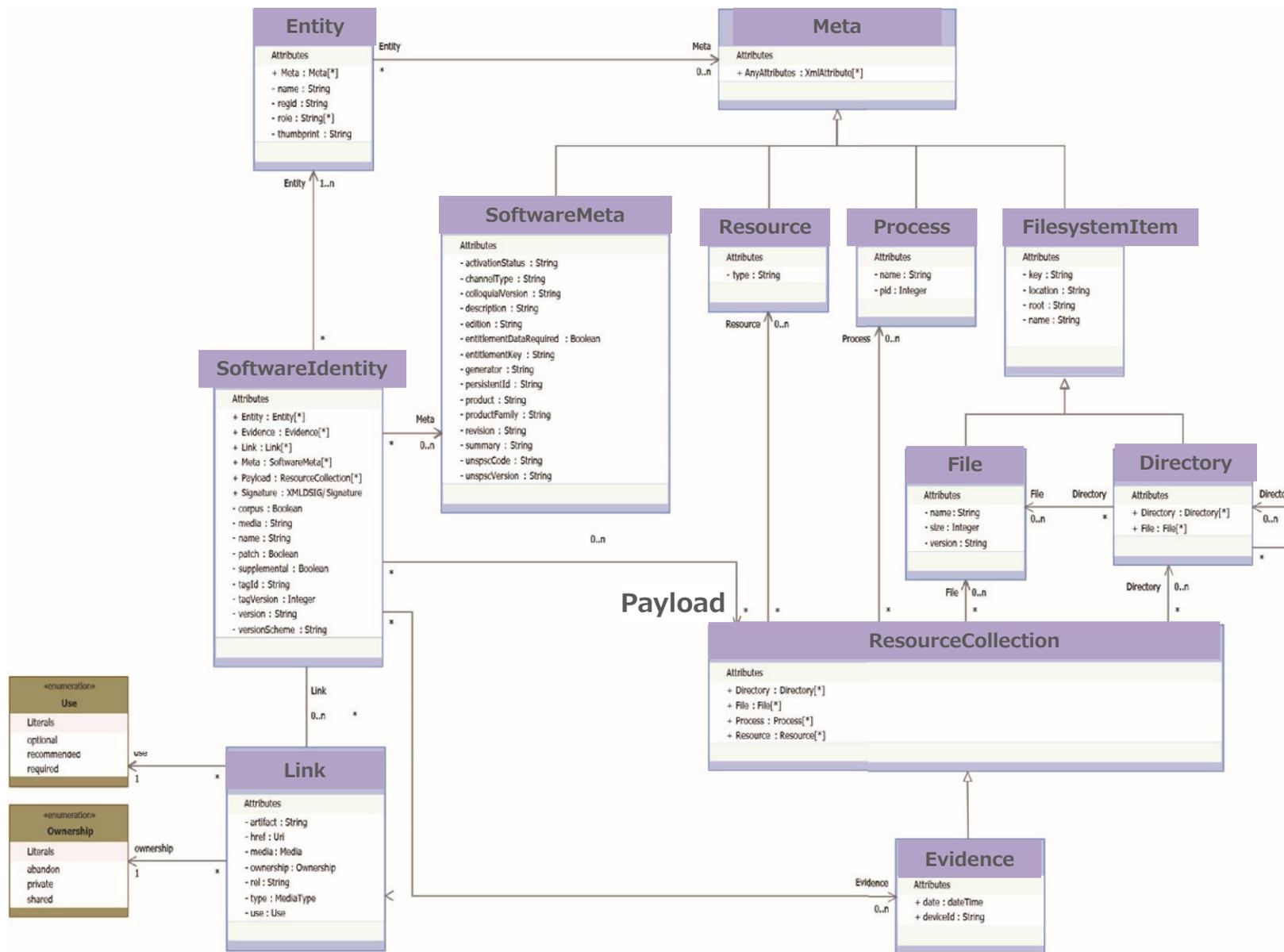
# ソフトウェア識別タグでなにが定義できるか？

組織名\_プロダクト名.swidtag



要素	説明		
ルート属性(SoftwareIdentity)	ソフトウェア識別についてのルート属性を記述する。		
子要素	<a href="#">組織情報(Entity)</a>	タグを生成した組織は必須, 他は選択	このSWIDタグに対して責任のある組織の情報を記述する。
	<a href="#">リンク情報(Link)</a>	選択	他のファイルの参照関係を記述する。関係するファイルやダウンロード元, 脆弱性データベース, 使用権なども定義できる。
	<a href="#">メタ情報(Meta)</a>	選択	このSWIDに関する任意の情報を記述する。
	<a href="#">ソフトウェアの本来情報(Payload)</a>	選択	インストールされるファイルについて本来の情報を記述する。
	<a href="#">ソフトウェアの実際情報(Evidence)</a>	選択	SWIDタグが見つからないソフトウェアのシステム検査結果を記述する。
	署名情報(Signature)	選択	このSWIDタグに対して責任のある組織の署名情報を記述する。

# SWIDタグのUMLダイアグラム (参考)



# SWIDタグ 例 (Fabrikam Writer 2013)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SoftwareIdentity
  xmlns="http://standards.iso.org/iso/19770/-2/2014/schema.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:SHA256="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"
  name="Fabrikam Writer 2013"
  tagId="{GUID}FabrikamWriter-2013"
  version="12.1.1"
  versionScheme="multipartnumeric">
  <Entity name="Fabrikam Corporation" regid="fabrikam.com" role="softwareCreator
  licenser tagCreator" />
  <!-- also requires the C++ runtime -->
  <Link rel="requires" href="swidpath:/SoftwareIdentity/Meta[@persistentId='EAB2F277-
  0920-4F8C-B466-6FE0CA7D59DF' ] ownership="shared" />
  <Payload>
    <File name="writer.exe" size="1024000" location="../bin/"
    SHA256:hash="c4128c2dc663ae7a6b6bc6787594057396e6b3f569cd50fd5ddb4d1bbafd2b6a" />
    <File name="writer.dll" size="524012" location="../bin/"
    SHA256:hash="96e6b3f569cd50fd5ddb4d1bbafd2b6ac4128c2dc663ae7a6b6bc67875940573" />
  </Payload>
</SoftwareIdentity>
```

XML宣言

19770-2デフォルト名前空間

W3C XMLスキーマ名前空間

デジタル承認名前空間

ハッシュ名前空間

タグの名前

判り易くするために記述しているが、16バイトのGUID (例えば、2df9de35-0aff-4a86-ace6f7dddd1ade4c) を記述する。

persistentIdが'EAB2---'であるSWIDタグが示しているプロダクト

C++が共用されている

writer.dllの正しいサイズ、位置、Hash情報

# プレインストールソフトウェアのSWIDタグ 例

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<SoftwareIdentity
```

```
  xmlns="http://standards.iso.org/iso/19770/-2/2014/schema.xsd"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
```

```
  xmlns:SHA256="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"
```

```
  name="Fabrikam Writer 2013"
```

```
  tagId="{GUID}FabrikamWriter-2013"
```

```
  version="12.1.1"
```

```
  versionScheme="multipartnumeric"
```

```
  corpus="true" >
```

このSWIDタグが示しているプロダクトがプレインストール製品

```
  <Entity name="Fabrikam Corporation" regid="fabrikam.com" role="softwareCreator  
  licensor tagCreator" />
```

```
<!-- also requires the C++ runtime -->
```

```
  <Link rel="requires" href="swidpath:/SoftwareIdentity/Meta[persistentId='EAB2F277-  
0920-4F8C-B466-6FE0CA7D59DF']" ownership="shared" />
```

```
<Payload>
```

```
  <File name="writer.exe" size="1024000" location="../bin/"
```

```
    SHA256:hash="c4128c2dc663ae7a6b6bc6787594057396e6b3f569cd50fd5ddb4d1bbafd2b6a" />
```

```
  <File name="writer.dll" size="524012" location="../bin/"
```

```
    SHA256:hash="96e6b3f569cd50fd5ddb4d1bbafd2b6ac4128c2dc663ae7a6b6bc67875940573" />
```

```
</Payload>
```

```
</SoftwareIdentity>
```

# パッチのSWIDタグ 例

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<SoftwareIdentity
```

```
  xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
```

```
  xmlns:SHA256="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"
```

```
  name="Fabrikam Writer 2013 FP13-4"
```

```
  tagId="{GUID}Fabrikam2013-FP13-4"
```

```
  version="1.0"
```

```
  versionScheme="multipartnumeric"
```

```
  patch="true">
```

このSWIDタグが示している  
ソフトウェアがパッチ

このソフトウェア(パッチ)の継承元のSWIDタグのtagIdが  
{GUID}FabrikamWriter2013であること

```
<!-- this software is created, licensed and tagged by the same org -->
```

```
<Entity name="Fabrikam Corporation" regid="fabrikam.com" role="softwareCreator  
licensor tagCreator" />
```

```
<Link rel="ancestor" href="swid: {GUID}FabrikamWriter-2013" />
```

```
<Payload>
```

```
<File name="writer.dll" size="550012" location="../bin/"
```

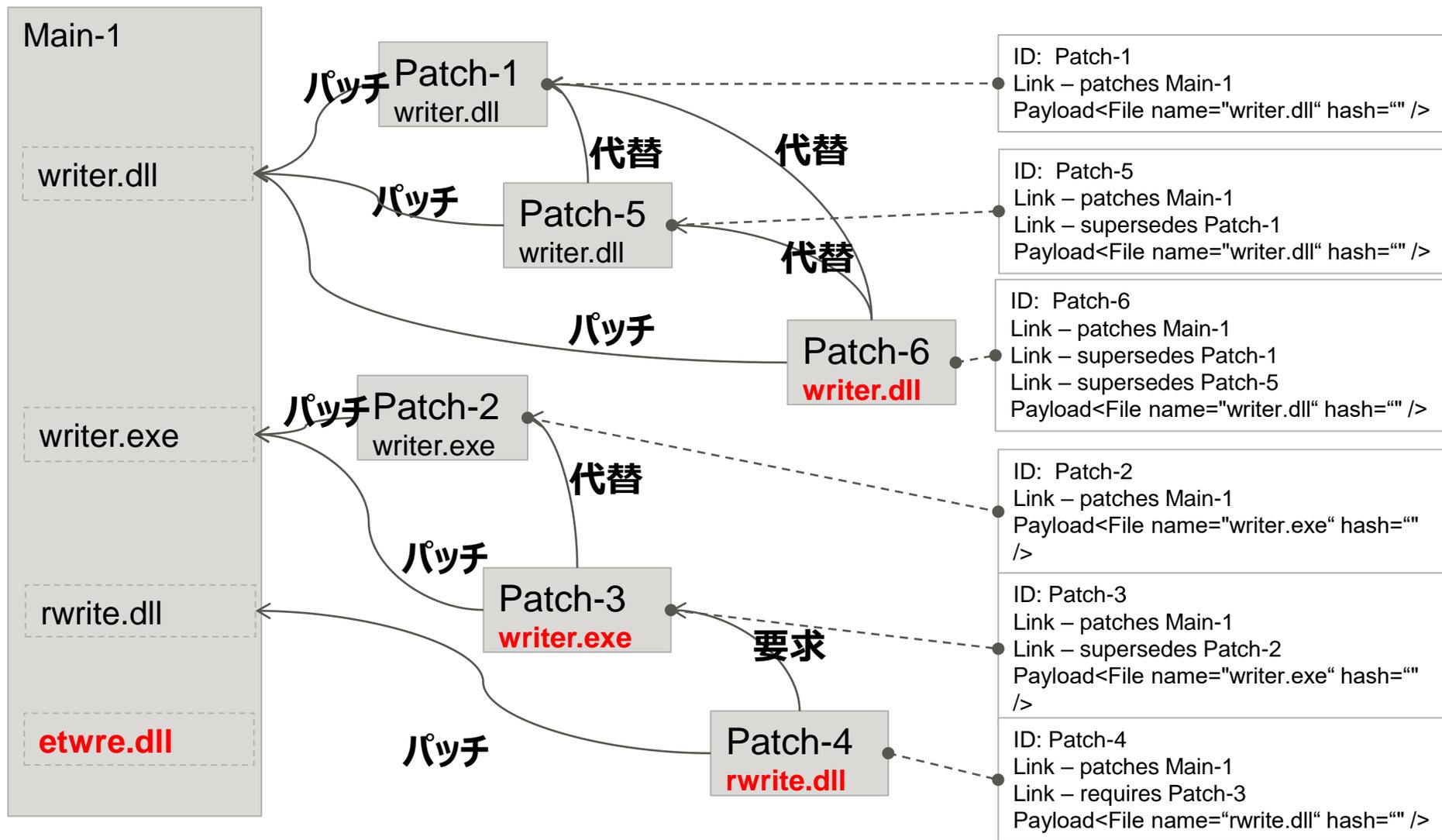
```
  SHA256:hash="db4d1bbafd2b6ac41296e6b3f569cd50fd5d8c2dc663ae7a6b6bc67875940573" />
```

```
</Payload>
```

パッチ適用後のwriter.dllの正しいサイズ、位置、Hash情報

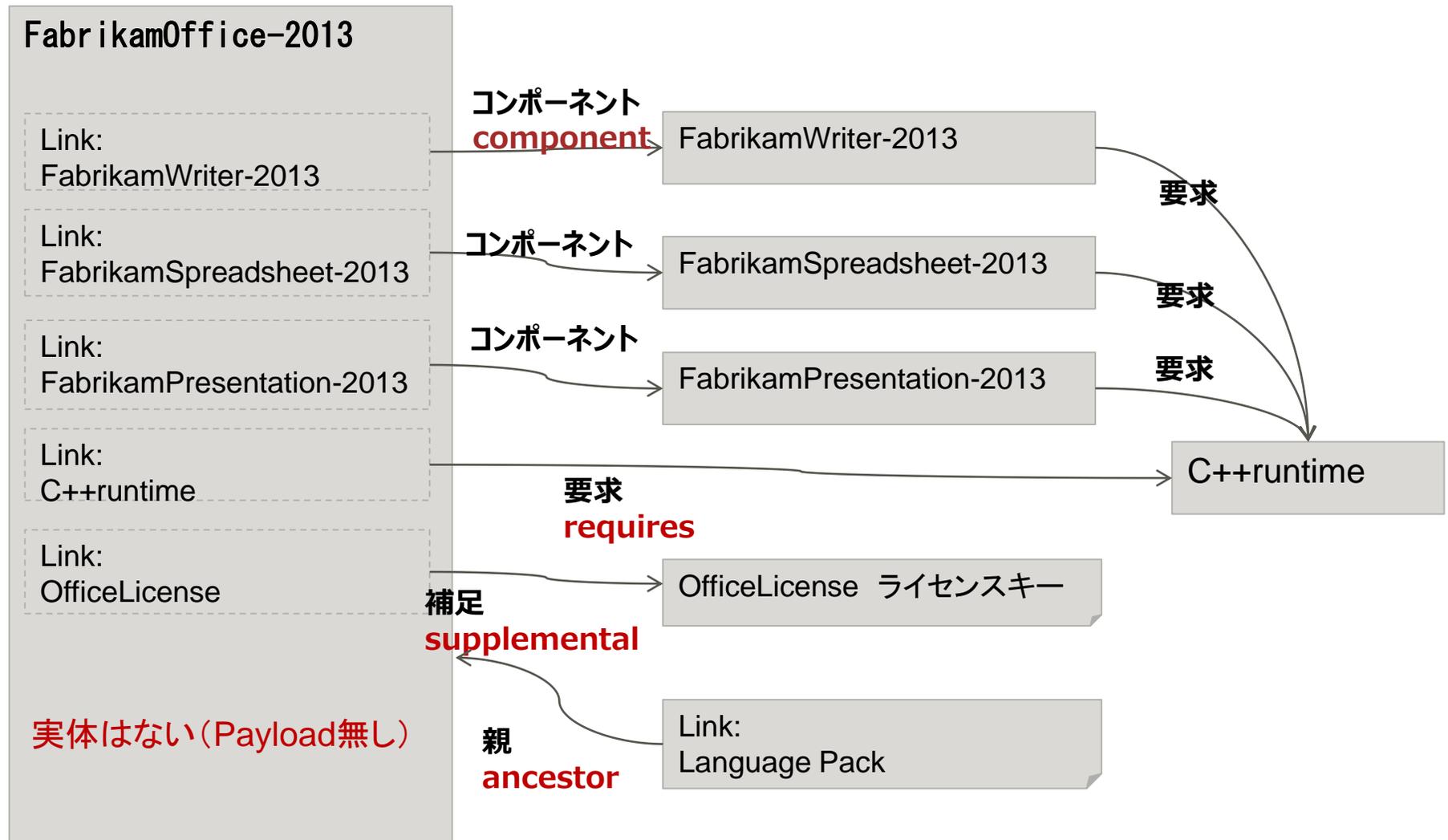
```
</SoftwareIdentity>
```

# パッチの管理



- ID: Patch-1  
Link – patches Main-1  
Payload<File name="writer.dll" hash="" />
- ID: Patch-5  
Link – patches Main-1  
Link – supersedes Patch-1  
Payload<File name="writer.dll" hash="" />
- ID: Patch-6  
Link – patches Main-1  
Link – supersedes Patch-1  
Link – supersedes Patch-5  
Payload<File name="writer.dll" hash="" />
- ID: Patch-2  
Link – patches Main-1  
Payload<File name="writer.exe" hash="" />
- ID: Patch-3  
Link – patches Main-1  
Link – supersedes Patch-2  
Payload<File name="writer.exe" hash="" />
- ID: Patch-4  
Link – patches Main-1  
Link – requires Patch-3  
Payload<File name="rwrite.dll" hash="" />

# スウィート製品SWIDタグのリンク関係



□ : タグファイル ( ?.swidtag )、

□ : 補足タグファイル ( ?.swidtag )、

← : リンク関係

# スウィート製品のSWIDタグ 例

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<SoftwareIdentity
```

```
  xmlns="http://standards.iso.org/iso/19770/="
  xmlns:xsi="http://www.w3.org/2001/XMLSchema="
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
```

```
  name="Fabrikam Office 2013"
  tagId="{GUID}FabrikamOffice-2013"
  version="12.1.1"
  versionScheme="multipartnumeric">
```

名前が“Fabrikam Office 2013”で  
tagIdが“={GUID}FabrikamOffice-2013”  
のSWIDタグ。

SWIDタグのtagIdが{GUID}Fabrikam2013  
のソフトウェアはFabricam Office 2013のコンポーネント  
で推奨されている。

```
<Entity name="Fabrikam Corporation" regid="fabrikam.com" role="softwareCreator licensor
tagCreator" />
```

このコンポーネントはオプション

```
<Link rel="component" href="swid: {GUID}FabrikamWriter-2013" use="recommended" />
```

```
<Link rel="component" href="swid: {GUID}FabrikamSpreadsheet-2013" use="recommended" />
```

```
<Link rel="component" href="swid: {GUID}FabrikamPresentation-2013" use="optional" />
```

```
<!-- also require
```

同じディレクトリに“FabrikamOfficeLicense.swidtag”の補足タグがある

```
<Link rel="requires" href="swidpath:/softwareidentity/meta[epersistencid= EAB21277-0920-
4F8C-B466-6FE0CA7D59DF' 1" ownership="shared" />
```

```
<Link rel="supplemental" href="./FabrikamOfficeLicense.swidtag" />
```

```
<Meta entitlementDataRequired="true" />
```

```
<Meta SKU="FabrikamOfficeStandard-2013-Retail" />
```

```
</SoftwareIdentity>
```

アクティベーションキーが要求されている

# アクティベーション補足SWIDタグ 例

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<SoftwareIdentity
```

```
xmlns= "http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
```

```
xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns:ds= "http://www.w3.org/2000/09/xmldsig#"
```

```
name="Fabrikam Office 2013"  
tagId="{GUID}FabrikamOffice-2013"  
version="12.1.1"  
versionScheme="multipartnumeric"  
supplemental="true">
```

名前が"Fabrikam Office 2013"で  
tagIdが "{GUID}FabrikamOffice-2013" の  
補足タグ (supplemental="true")

```
<Entity name="Fabrikam Corporation " regid="fabrikam.com " role="softwareCreator  
licensor tagCreator" />
```

```
<!-- Product Key for this product-->
```

```
<Meta entitlementKey="IDFKK-A9123-PG123-SFA97-CCADV" />
```

```
</SoftwareIdentity>
```

アクティベーションキーが  
"IDFKK-A9123-PG123-SFA97-CCADV"

# 言語の補足SWIDタグ 例

```
<?xml version="1.0" encoding="UTF-8"
<SoftwareIdentity
```

```
xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
```

```
name="Fabrikam Office 2013 Language Pack - French"
tagId="{GUID}FabrikamOffice-2013-French"
version="12.1.1"
versionScheme="multipartnumeric"
supplemental="true">
```

```
<Link rel="parent" href="swid:{GUID}FabrikamOffice-2013" />
```

```
<Entity name="Fabrikam Corporation" regid="fabrikam.com" role="softwareCreator
licensor tagCreator" />
```

```
</SoftwareIdentity>
```

名前が” Fabrikam Office 2013 Language Pack - French”で  
tagIdが””{GUID}FabrikamOffice-2013-French”の  
補足タグ(supplemental=“true”)

Link先が親でそのSWIDタグのtagIdが  
” {GUID}FabrikamOffice-2013”であることを示す

# SWIDタグ導入の課題

- 誰がSWIDタグを導入するのか？
- SWIDタグの認証は？

## 権利構造タグ

19770-3 権利スキーマの概要  
権利スキーマのXSDファイル：

<http://standards.iso.org/iso/19770/-3/2015/schema.xsd>

# ENTで何が定義できるのか？

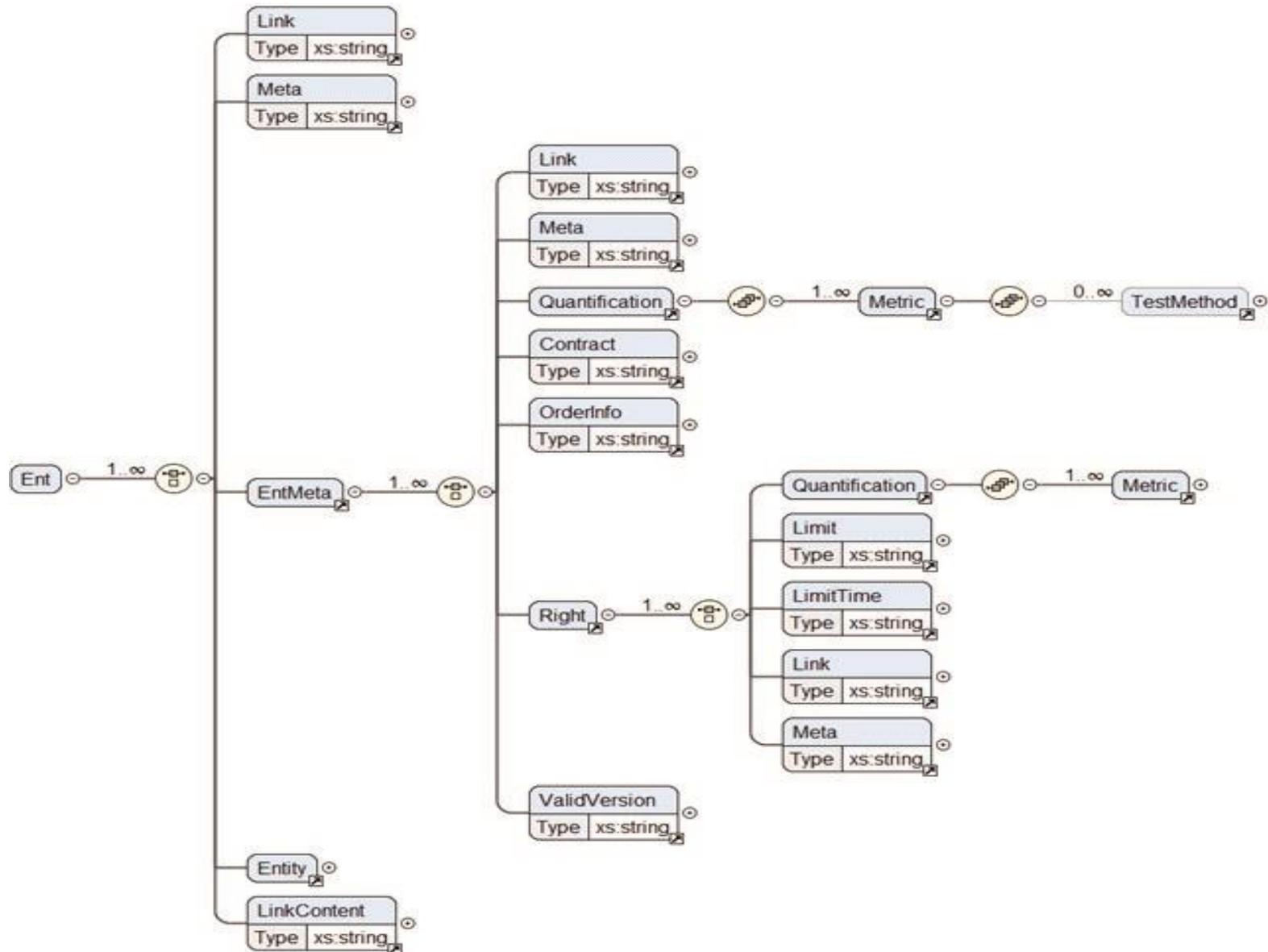
注：権利（Entitlement）を定義したファイルをEntと呼ぶ

組織名\_プロダクト名\_entId.ent



要素		説明	
<a href="#">ルート属性(Ent)</a>		権利についてのルート属性を記述する。	
子要素	<a href="#">組織情報(Entity)</a>	EntCreatorは必須, 他は選択	Entが定義している権利に関連する組織の情報を記述する。
	<a href="#">契約メタ情報(EntMeta)</a>	必須	契約に関する情報
	<a href="#">リンク情報(Link)</a>	選択	他のファイルの参照関係を記述する。関係するファイルやダウンロード元, 脆弱性データベース, 使用権なども定義できる。
	<a href="#">メタ情報(Meta)</a>	選択 (1個のみ)	この契約に関する任意の情報を記述する。
	<a href="#">署名情報(Signature)</a>	選択	このEntに対して責任のある組織の署名情報を記述する

# EntタグのUMLダイアグラム (参考)



# ライセンスの定義の例

顧客 A がfabrikam社のサーバライセンスを100購入した。インストールライセンスが100、CALライセンスは1 CPU単位に5 CALある。

fabrikam.com\_server2014\_AAA.ent

ライセンス数とそのカウント方法

```
<ent entId="AAA" entType="Initial" entCreationDate="2014-10-13T15:09:10+00:00">
  <Entity name="Fabrikam" regid="fabrikam.com" role="softwareCreator entCreator"/>
  <Entity name="顧客A" regid="顧客A.com" role="entitledEntity" alias="本社"/>
  <EntMeta products="server2014" entitlementType="License" >
```

```
  <Quantification quantity="100" >
```

```
    <Metric metricName="per Server" metricType="Device">
```

```
      <TestMethod testName="For SWIDTAG"/>
```

19770-2でのカウント方法

```
      <TestScript="tagId={GUID} server2014 tagCreatorRegid=fabrikam.com"/>
```

```
    </Metric>    </Quantification>
```

```
  <Right rightId="1" rightName="installLicense" >
```

```
    <LimitTime isPerpetual="true"/>
```

サーバライセンスの権利

```
  </Right>
```

```
  <Right rightId="2" rightName="CAL" >
```

サーバのCALライセンスの権利

```
    <LimitTime isPerpetual="true"/>
```

```
    <Quantification quantity="5" />
```

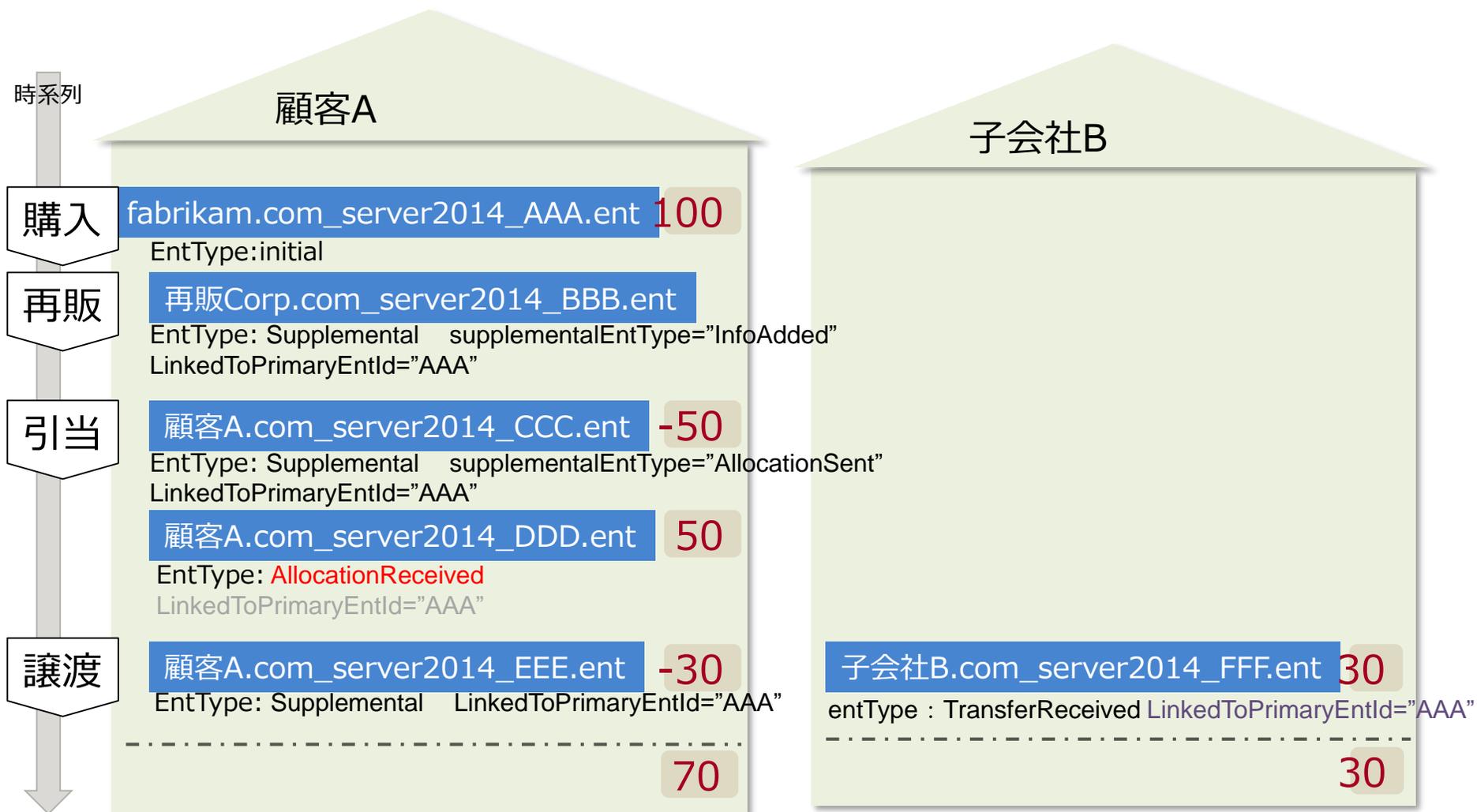
一つのサーバで5CALが許される

```
  </Right>
```

```
</EntMeta>
```

...

# 契約ファイル (Ent) のライフサイクル



# 再販業者による情報付与の例

再販業者（Ex. 再販Corp）が、Fabrikam社が発行したライセンスに情報を付加する。

fabrikam.com\_server2014\_AAA.ent

```
<ent entId="AAA" entType="Initial" entCreationDate="2014-10-13T15:09:10+00:00">
```

...

Entファイルのリンク関係

再販Corp.com\_server2014\_BBB.ent

```
<ent entId="BBB" entType="Supplemental" supplementalEntType="InfoAdded"
  linkedToPrimaryEntId="AAA"
  entCreationDate="2014-10-13T15:09:10+00:00"
  <Entity name="再販Corp" regid="再販Corp.com" role="entCreator"/>
```

...

Entタイプが補足

補足Entのタイプが追加情報

# 権利引当の例

ユーザ (Ex.顧客A)が、他の部門(事業部A)にライセンスを50引当てる。(ライセンスの管理を任せる)

fabrikam.com\_server2014\_AAA.ent

```
<ent entId="AAA" entityType="Initial" entCreationDate="2014-10-13">
  <Entity name="Fabrikam" regid="fabrikam.com" role="softwareCreator entCreator"/>
  ...
```

顧客A.com\_server2014\_CCC.ent

Entファイルのリンク関係

```
<ent entId="CCC" entityType="Supplemental" supplementalEntityType="AllocationSent"
  LinkedToPrimaryEntId="AAA" entCreationDate="2015-05-13">
  <Entity name="顧客A" regid="顧客A.com" role="entCreator" alias="本社"/>
  <EntMeta
    <Right rightId="1" rightName="installLicense" >
      <LimitTime isPerpetual="true"/> </Right>
    <Quantification quantity="-50">
    ...
```

引き当てるEnt情報

Entタイプが補足

50ライセンスの引当

顧客A.com\_server2014\_DDD.ent

```
<ent entId="DDD" entityType="AllocationReceived"
  LinkedToPrimaryEntId="AAA" entCreationDate="2015-05-13">
  <Entity name="顧客A" regid="顧客A.com" role="entCreator" alias="A事業部"/>
  <EntMeta
    <Right rightId="1" rightName="installLicense" >
      <LimitTime isPerpetual="true"/> </Right>
    <Quantification quantity="50">
```

引き当てられたEnt情報

# 権利譲渡の例

顧客A社は、fabrikam社から購入したライセンス100のうち30を子会社Bに譲渡

fabrikam.com\_server2014\_AAA.ent

```
<ent entId="AAA" entType="Initial" entCreationDate="2014-10-13">
  <Entity name="Fabrikam" regid="fabrikam.com" role="softwareCreator entCreator"/>
  ...
```

顧客A.com\_server2014\_EEE.ent

Entファイルのリンク関係

```
<ent entId="EEE" entType="Supplemental" supplementalEntType="TransferSent"
  LinkedToPrimaryEntId="AAA" entCreationDate="2015-05-13">
  <Entity name="顧客A" regid="顧客A.com" role="entCreator" alias="本社"/>
  <EntMeta
    <Right rightId="1" rightName="installLicense" >
      <LimitTime isPerpetual="true"/> </Right>
    <Quantification quantity="-30">
  ...
```

譲渡するEnt

Entタイプが補足

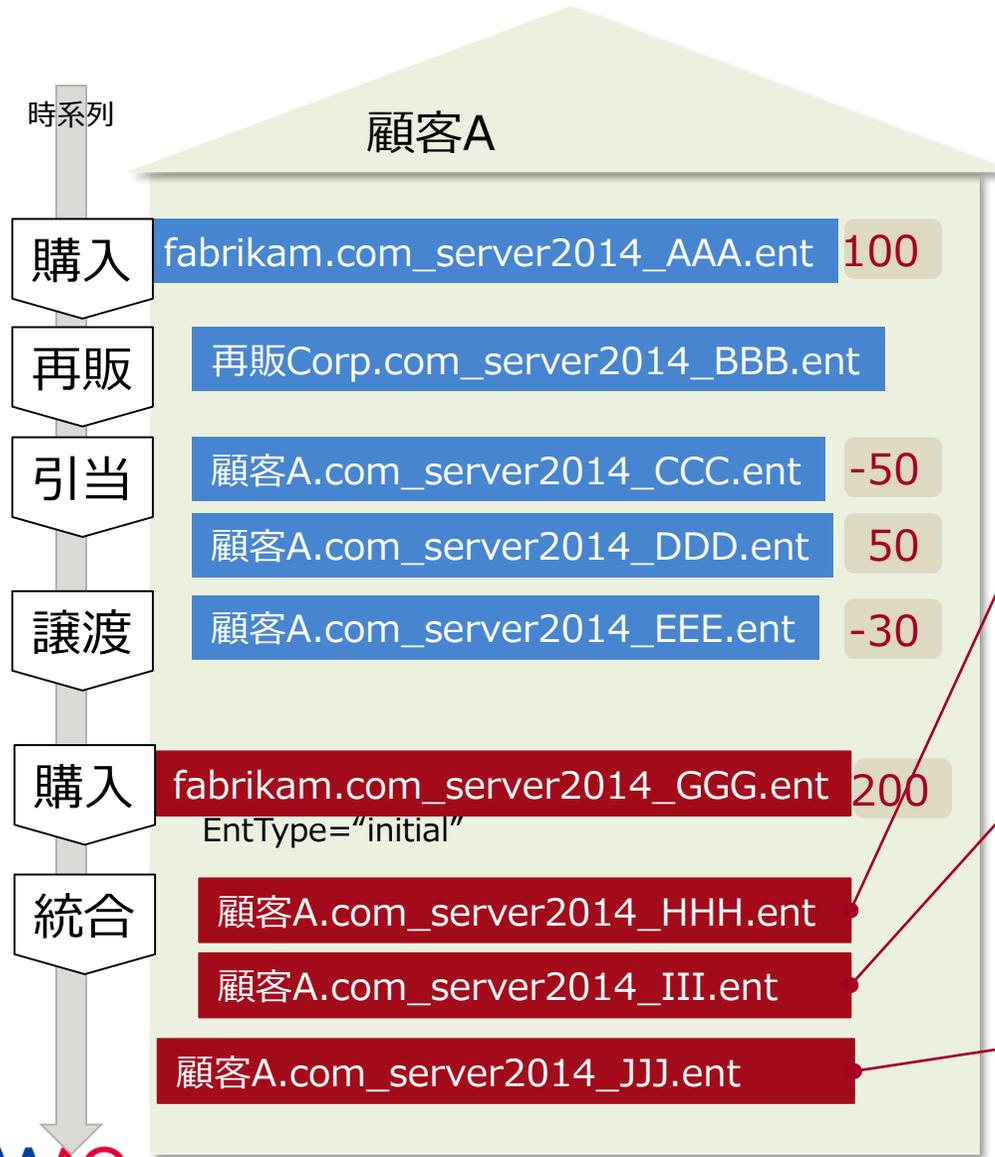
30ライセンスの譲渡

子会社B.com\_server2014\_FFF.ent

```
<ent entId="FFF" entType="TransferReceived"
  LinkedToPrimaryEntId="AAA" entCreationDate="2015-05-13">
  <Entity name="子会社B" regid="子会社B.com" role="entCreator"/>
  <EntMeta
    <Right rightId="1" rightName="installLicense" >
      <LimitTime isPerpetual="true"/> </Right>
    <Quantification quantity="30">
```

譲渡を受ける組織で作成されるEnt

# 権利統合の例



- entId="AAA"が統合されたことを示す
- entId="GGG"が統合されたことを示す
- 統合されたEntであることを示す
- 統合したentIdを全てリストする

```
<ent
  entId="HHH"
  entType="Supplemental"
  supplementalEntType="ConsolidationPart"
  linkedToPrimaryEntId="AAA"
  ...
```

```
<ent
  entId="III"
  entType="Supplemental"
  supplementalEntType="ConsolidationPart"
  linkedToPrimaryEntId="GGG"
  ...
```

統合後のライセンス数

```
<ent entId="JJJ" entType="Consolidation"
  consolidationEntId="AAA"
  consolidationEntId="GGG"
  <Quantification quantity="270"
  ... (統合した全ての契約情報を転記する)
```

# ENTタグ導入の課題

- ソフトウェア開発者とライセンス販売業者間のルールは？
- ライセンス販売業者と利用者間のルールは？
- ライセンス販売時のENTタグの認証は？

# 資源利用状況測定タグ

**19770-4:資源利用状況測定の概要**  
**(RUM : Resource Utilization Measurement)**  
**<http://standards.iso.org/iso/19770/-4/2015/schema.xsd>**

# RUMで何が定義できるのか？

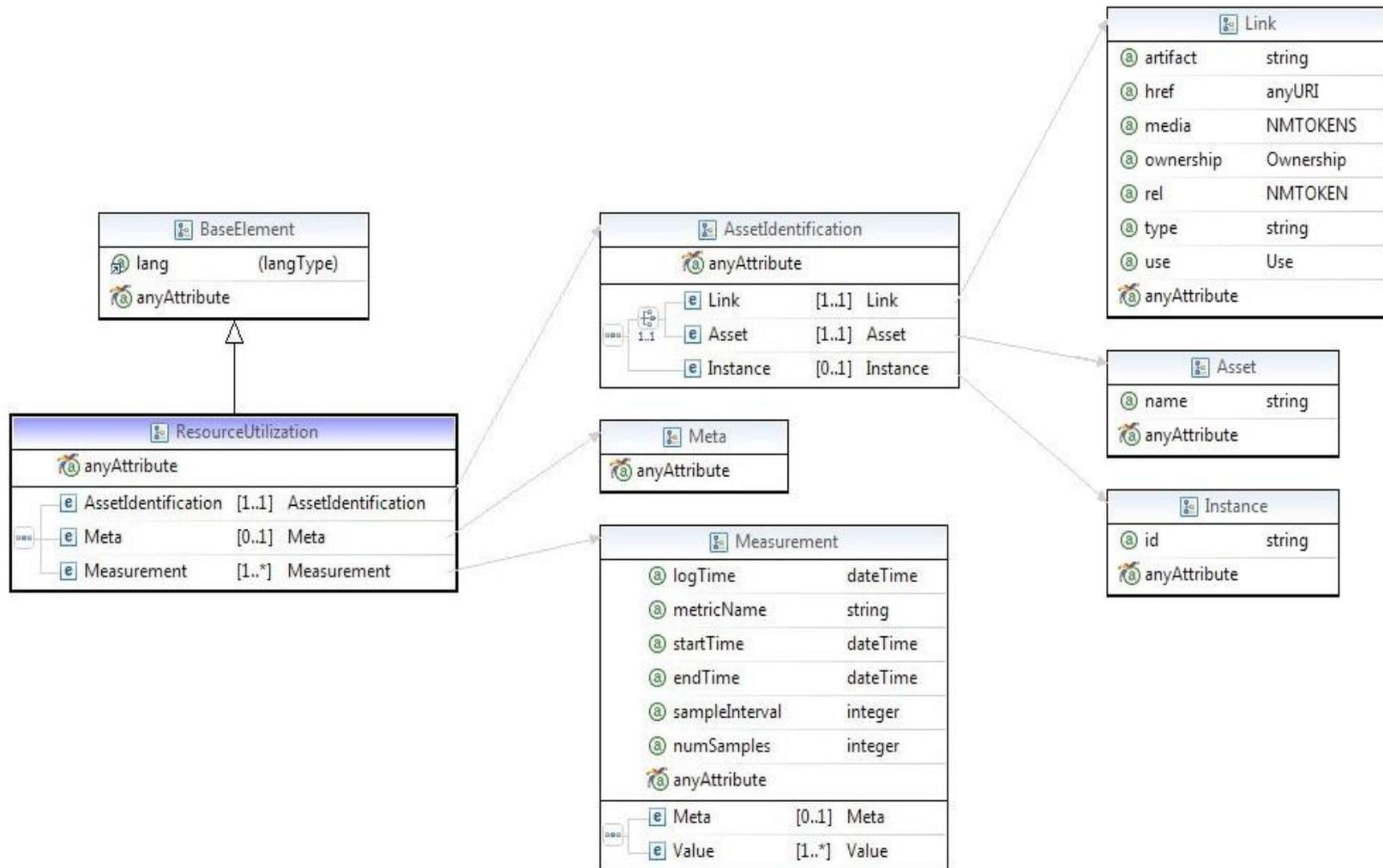
注：資源利用状況測定を定義したファイルをRumと呼ぶ

SWIDタグファイル名.ログファイル名.数字.rum



要素			説明
ルート属性(ResourceUtilization)			権利についてのルート属性を記述する。
子要素	資産識別子 (AssetIdentification)	必須	IT資産の識別子
	メタ情報 (Meta)	選択（1個のみ）	このRUMに関する任意の情報を記述する。
	測定情報 (Measurement)	選択	測定データ

# RUMのUMLダイアグラム (参考)



# RUMの例

## Fabrikam Webserver の利用状況測定の例

<?xml version="1.0" encoding="UTF-8"?> XML宣言

<ResourceUtilization

xmlns="http://standards.iso.org/iso/19770/-4/2015/schema.xsd"  
xmlns:xml="http://www.w3.org/XML/1998/namespace"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://standards.iso.org/iso/19770/-4/2015/schema.xsd schema.xsd ">

名前空間定義

<AssetIdentification>  
  <Link href="swid:FabrikamWebserver-2.2" rel="asset" />  
</AssetIdentification>

測定対象の資産

<Measurement logTime="2014-11-25T09:31:10+06:00"  
  startTime="2014-11-24T09:30:10+06:00"  endTime="2014-11-25T09:30:10+06:00"  
  metricName="AUTHORIZED\_USER">  
    <Meta subtype="EMPLOYEE" />  
    <Value type="number">12</Value>  </Measurement>

測定ログ

計量者の名前

<Measurement logTime="2014-11-26T09:31:10+06:00"  
  startTime="2014-11-25T09:30:10+06:00"  endTime="2014-11-26T09:30:10+06:00"  
  metricName="AUTHORIZED\_USER">  
    <Meta subtype="EMPLOYEE" />  
    <Value type="number">11</Value>  </Measurement>

測定対象を示す任意の記述

値：12人が利用

そのSWIDで利用状況の測定を行う

</ResourceUtilization>

Link測定対象とRUMとの関係を定義する  
"asset":SWIDタグ  
"metric":このRUMの定義  
"supplemental":このRUMに対する追加  
情報  
計量者の名前

# RUMタグ導入の課題

- ソフトウェア開発者とライセンス販売業者間のルールは？
- ライセンス販売業者と利用者間のルールは？
- ライセンス販売時のENTタグの認証は？

# まとめ

# 参考：米国の動向

## TagVault.org (IEEE-ISTOの非営利下部組織)

設立時のボード会員：

CA、Microsoft、ModusLink、Symantec

主要メンバー：

HP, EMC<sup>2</sup>, IBM、Anglepoint, Eracent, Gothaer Systems GmbH, iQuate, MITRE, Scalable Software, Open iT, US ARMY, 国防総省 (DoD), 国土安全保障省(DHS), 連邦政府調達局(GSA), US NAVY, 国立標準技術研究所(NIST)

アライアンスパートナー：

Agnitio Advisors, Cicala and Associates, CyberPack Ventures, IAITAM, IEEE-IS

## SWIDタグ適用ソフトウェア開発者：

Microsoft, Symantec, Adobe, IBM (約 300 product /月) など

## ツールベンダー：

CA Technologies (IT Client and Asset Manager), Hewlett Packard - DDMI, Microsoft - SCCM, Symantec - Altiris, Aspera, Asset Metrics, Eracent, Express Metrix, Flexera, iQuate, Magnicomp, Software Management.org

Caphyon's Advanced Installer, Flexera Software's InstallShield, Flexera Software's InstallAnywhere, Open Source - Windows Installer XML Toolset (WiX) など

# 参考：SAMAC辞書項目とSWIDタグ項目の関係

SAMAC辞書項目	メーカー名	インストール名称	エイリアス	エディション	バージョン	ソフトウェア種別
SAMAC辞書データの例	Microsoft	Microsoft Office Professional Plus 2013	Office	Professional Plus	2013	有償ソフトウェア



SWIDタグ	Entity要素の name	ルート要素の name	Meta要素の product	Meta要素の edition	Meta要素の colloquial Version	ルート要素でパッチ、バンドル Meta要素で有/無償その他任意の種別が定義可能

グローバルで一意的な識別子（GUIDを推奨）も利用可

VersionSchemeで定義した形式のVersionもルート要素で定義可能

# まとめ

- インベントリやライセンスの情報構造の標準化が使えるレベルに進展してきた
- ソフトウェアの管理単位（SWIDタグ）は、ISMSやITSMSの元となる
- ライセンスの管理単位（ENT）は、権利の条件をSWIDタグやリソースの利用状況を規定するRUMなどを使用して省力化が可能なこと
- ソフトウェアベンダーがプロダクト開発と同じレベルでタグに対応することが重要
- ソフトウェア開発者とライセンス販売業者間の取り決めが重要
- IT資産管理、セキュリティ管理のツールベンダーも積極的に標準化された情報構造を活用することが重要（むしろ先行すべきか？）
- 認証の方法

The logo for SAMAC consists of the letters S, A, M, M, A, and C. The 'S' and 'C' are red, while the two 'M's are blue. The letters are stylized and connected at the base.

一般社団法人 ソフトウェア資産管理評価認定協会