

日本教育工学会 2016年度産学協同セミナー
2017年3月25日(土) 13:30~17:30 @内田洋行新川オフィス

教育ビックデータ・学習資源の共有の実現に向けて
教育工学の立場から国際標準規格の必要性を考える

IMS OneRoster

熊本大学 中野 裕司

所属・専任: 総合情報統括センター, 教授システム学専攻
兼任・兼担: 教授システム学研究センター(RCiS), eラーニング推進室,
情報電気電子工学専攻, 大学情報分析室,
大学教育統括管理運営機構

 Kumamoto University

 RCiS

教授システム学研究センター
Research Center for Instructional Systems

- 基盤として、SPA + Web API + 標準化
 - SPA (Single Page Application), RESTful Web API (JSON / JSONP)
- MS Enterpriseによる学籍情報、履修情報の共有
 - 大学ポータルを中心としたサービスとシステム間連携の必要性
 - 統合IDとSSO、科目情報&学生の基本・履修情報等の共有の主な流れ
 - IMS Enterpriseの例(学生、教職員の基本情報, 科目情報)
- IMS LISとOneRosterによる動的連携への発展の期待
 - Learning Information Services (LIS)
 - OneRoster
 - ◆ データ定義の例(組織、年度・学期、科目、クラス、ユーザ、成績等)
 - ◆ RESTインターフェース(抜粋)
 - ◆ Enterprise → OneRosterで何が変わった?

■ SPA (Single Page Application)

- SPAは、デスクトップアプリのように動作するWebアプリまたはWebサイトのこと。
- 最初にJavaScript, CSSも含めて全て必要なコードはロード済み。
(複数ページかあるように見えても、自分で書き換えたりしている)
- 必要なリソースは、その都度、動的にロードする (Ajax, Web API)。
- サーバ側でその都度ページを作らない
→ サーバ側の変更なしに修正等可能な場合も (HTML, CSS, JavaScriptのみ)
 - ◆ クライアントの変更 (WebAPI), WebAPIで新しいツールの開発が容易 (元)

■ RESTful Web API (JSON (LD) / JSONP)

- サーバとブラウザとのデータのやり取りのスタイル (Webサービスの一種)。
- 標準規格ではないが、スタイルが定義されている。
- JSONPの場合、ブラウザからクロスドメインなアクセス可。
 - ◆ マッシュアップ有利
- 将来、標準化に対応しやすい (と思う)

■ 標準化

- データ互換の確保 (学習ログ etc.)
- システム間連携 (同じ標準に対応したシステム間)

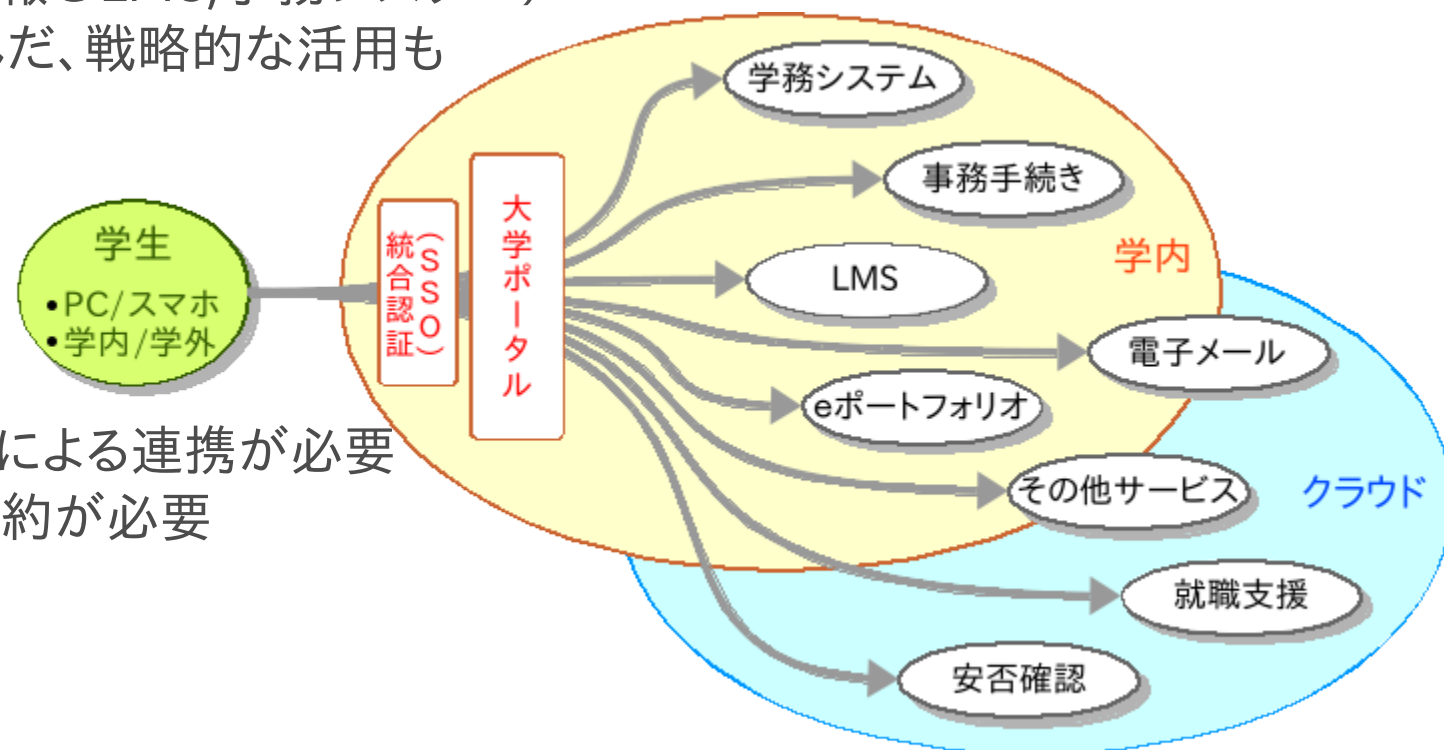
IMS Enterprizeによる 学籍情報、履修情報の共有

- 大学ポータルを中心としたサービスとシステム間連携の必要性
- 統合ID(熊大ID)とシングルサインオンの主な流れ
- 科目情報&学生の基本・履修情報等の共有の主な流れ
- IMS Enterprize
 - IMS Enterprize の例(学生、教職員の基本情報)
 - IMS Enterprize の例(科目情報)

大学ポータルを中心としたサービスとシステム間連携の必要性

■ 大学ポータル:

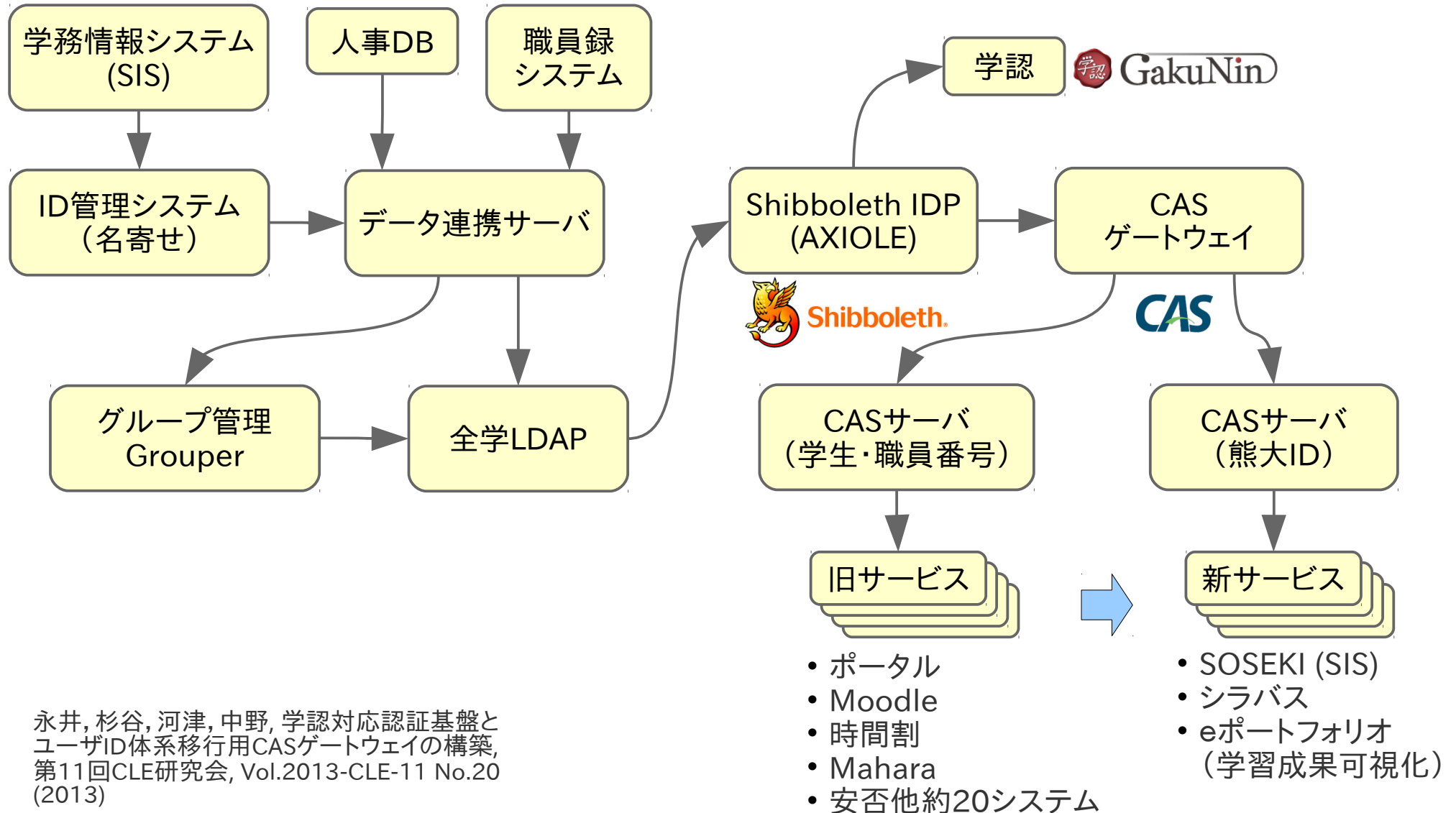
- 学生、教職員等のキャンパスライフをサポートする各種サービスへの入口
- SSOで認証され、個人に特化した機能やユーザ毎カスタマイズを提供
- 各種サービスの連携が必要
(例: 学籍,履修情報@LMS,学務システム)
- 卒業生を巻き込んだ、戦略的な活用も
(Kats編 2010)



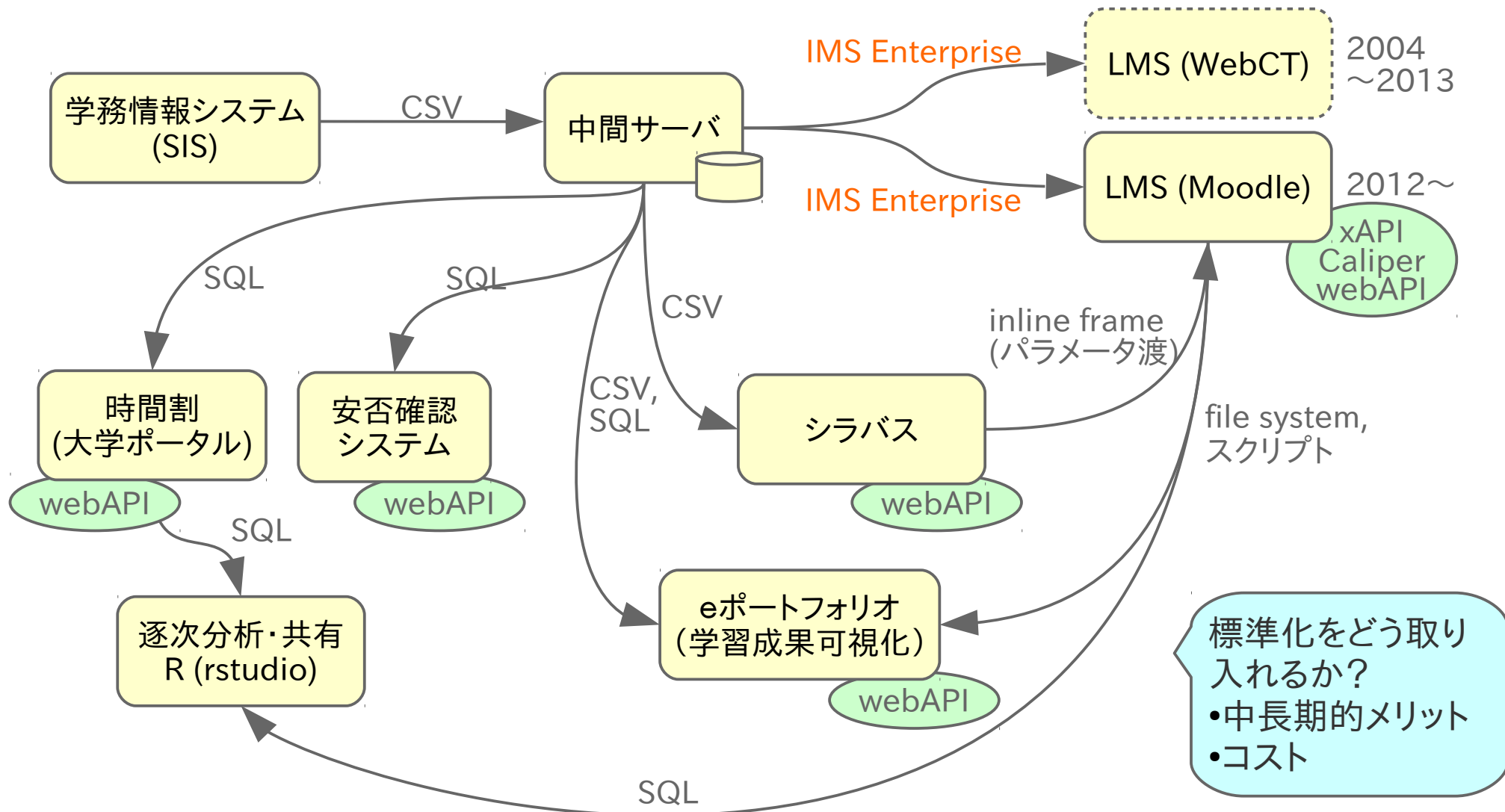
■ サービスの分散化

- 単一データソースによる連携が必要
- 分散したログの集約が必要

統合ID(熊大ID)とシングルサインオンの 主な流れ



科目情報&学生の基本・履修情報等の共有の主な流れ



標準化をどう取り入れるか?
 •中長期的メリット
 •コスト

- 科目情報、受講情報等を共有するための標準仕様
 - 2004年ごろ知ったこと: Web CT等のLMSは米では、科目や受講者の情報を学務情報システム(SIS)とこの仕様で簡単に連携できるらしい
 - 各種SIS(海外), WebCT, BlackBoard, Moodle, Sakai, Canvas 等が対応
- サポートする相互運用性
 - Personal Profile Data Maintenance (学生のプロフィール)
 - Group Management (科目、スケジュール等)
 - Enrollment Management (科目等グループへの登録者)
 - Final Result Processing (最終成績)
- OneRoster & LIS (Learning Information Services) へ発展?
 - <https://www.imsglobal.org/activity/onerosterlis>
 - まだ具体例は少ないが、動的かつ

IMS Enterprise の例 (学生、教職員の基本情報 1/2)

person.xml (17,547名登録, 約11MB)

```
<?xml version="1.0" encoding="UTF-8"?>
<enterprise>
  <properties>
    <datasource>WebCT</datasource>
    <datetime>2016-05-22T02:41:48JST</datetime>
  </properties>
  <person>
    <sourcedid>
      <source>kuLdap</source>
      <id>16XtXXXX</id>
    </sourcedid>
    <userid>16XtXXXX</userid>
    <name>
      <fn>熊大 太郎</fn>
      <n>
        <family>太郎</family>
        <given>熊大</given>
      </n>
    </name>
    <email>XXXX@XX.kumamoto-u.ac.jp</email>
```

中野, 喜多, 杉谷, 松葉, 右田, 武藏, 入口, 太田, 平, 辻, 島本, 木田, 宇佐川, 「WebCT、学務情報システムSOSEKI、教育用PCシステムのデータ同期」, 第2回WebCT研究会予稿集, pp.3-8 (2004-9)
<http://www.cc.kumamoto-u.ac.jp/sites/static/arcmit04/nakano.pdf>

1人目の登録開始

ユーザID

氏名

電子メール

IMS Enterprise の例 (学生、教職員の基本情報 2/2)

```
<extension>  
  <customcolumns xmlns="http://www.webct.com/IMS">  
    <column name="no">2016-tX-XXX</column>  
    <column name="role">0</column>  
    <column name="fulltime">0</column>  
    <column name="gakunen">1</column>  
    <column name="department">工学部情報電気電子工学科</column>  
  </customcolumns>  
</extension>
```

拡張項目

```
</person>
```

1人目の登録終了

```
<person>
```

2人目の登録開始

.....

.....

.....

```
</person>
```

```
</enterprise>
```

IMS Enterprise の例 (科目情報 1/3)

section.xml (149,516科目, のべ1,811,201名, 約481MB, 8年度分)

```
<?xml version="1.0" encoding="UTF-8"?>
<enterprise>
  <properties>
    <datasource>WebCT</datasource>
    <datetime>2015-10-09T02:35:44JST</datetime>
  </properties>
```

.....

```
<group>
  <sourcedid>
    <source>WebCT</source>
    <id>2016-58-XXXXX</id>
  </sourcedid>
  <grouptype>
    <scheme>LEARNING_CONTEXT_V1</scheme>
    <typevalue level="90"/>
  </grouptype>
  <description>
    <short>情報基礎A</short>
    <long>情報基礎A 2016前期 教養教育</long>
  </description>
```

1科目目の登録開始

科目ID

科目名

IMS Enterprise の例 (科目情報 2/3)

```
<relationship relation="1">
  <sourcedid>
    <source>WebCT</source>
    <id>2016-58-0000a</id>
  </sourcedid>
  <label/>
</relationship>
<relationship relation="1">
  <sourcedid>
    <source>WebCT</source>
    <id>2016-00-0000a</id>
  </sourcedid>
  <label>Term</label>
</relationship>
</group>
<membership>
  <sourcedid>
    <source>WebCT</source>
    <id>2016-58-XXXXX</id>
  </sourcedid>
```

学部コード

年度・学期コード

1科目目の登録終了

メンバーの登録開始

科目ID

IMS Enterprise の例 (科目情報 3/3)

```
<member>  
  <sourcedid>  
    <source>kuLdap</source>  
    <id>YYYYYYYY</id>  
  </sourcedid>  
  <idtype>1</idtype>  
  <role roletype="03">  
    <status>1</status>  
  </role>  
</member>
```

1人目のメンバー

ユーザID

教員として

```
.....  
<member>  
  <sourcedid>  
    <source>kuLdap</source>  
    <id>16AtAAAAA</id>  
  </sourcedid>  
  <idtype>1</idtype>  
  <role roletype="01">  
    <status>1</status>  
  </role>  
</member>
```

2人目のメンバー

ユーザID

学生として

```
.....  
</membership>  
<group>  
.....  
</enterprise>
```

メンバーの登録開始

IMS Learning Information Services (LIS)とOneRosterによる 動的連携への発展の期待

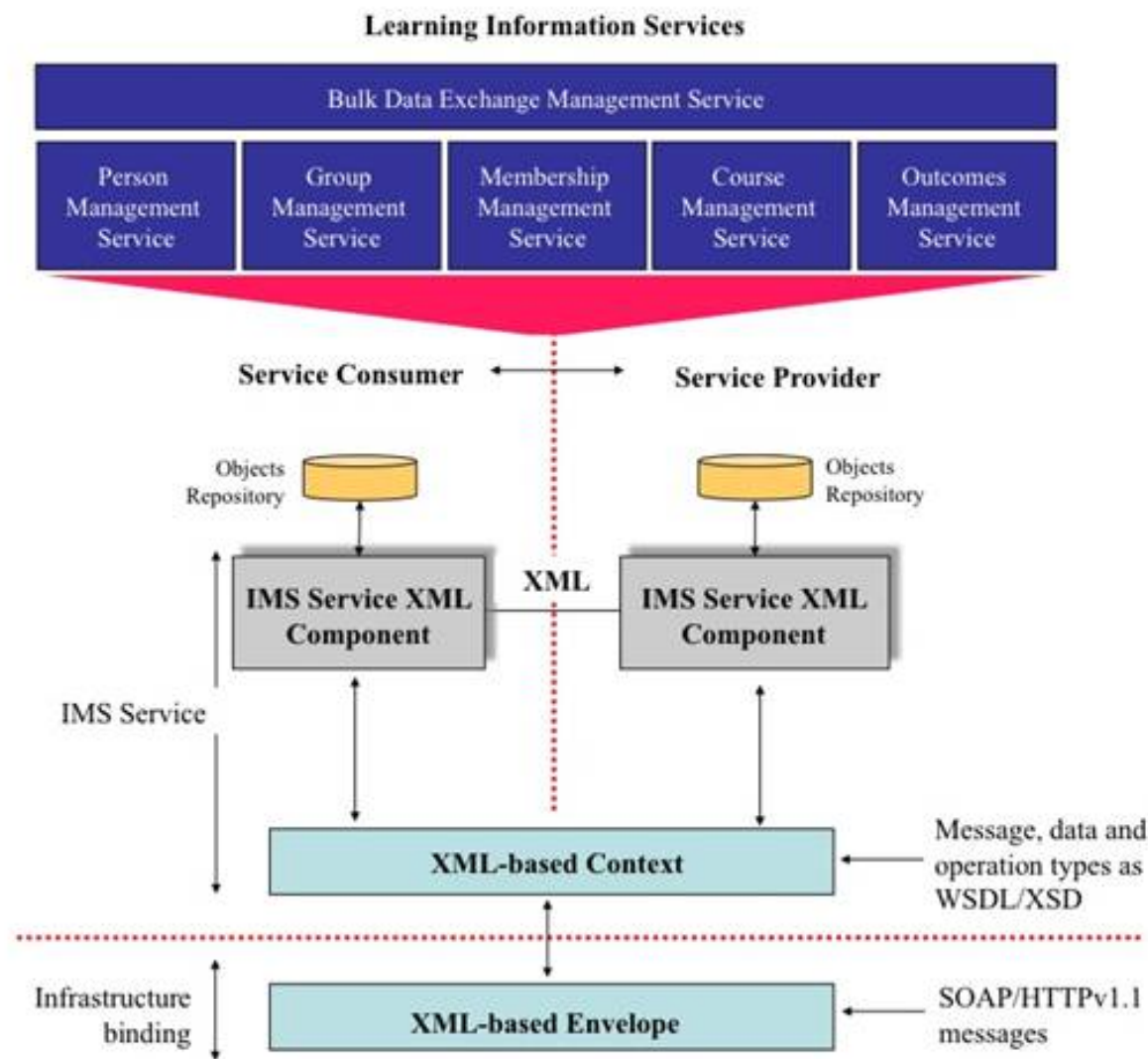
- Learning Information Services (LIS)
- OneRoster

Learning Information Services (LIS)

<https://www.imslobal.org/lis/lisv2p0p1/LISSpecPrimerv2p0p1.html>

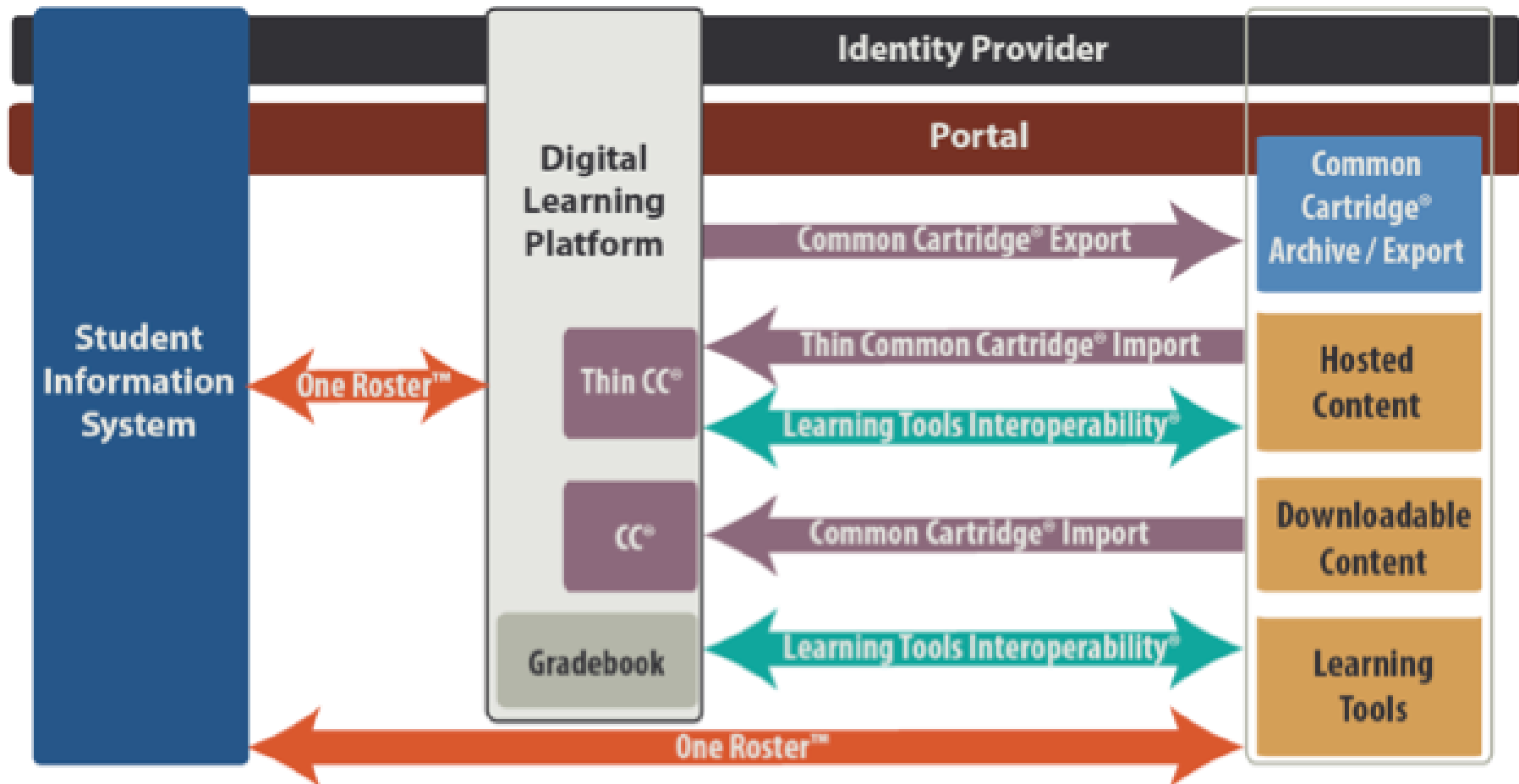
- 学修の文脈で、人々、グループ、メンバー、科目、成績についての情報交換を管理
- LIS規格 v2.0は、Enterprise Services規格 v1.0を置き換える
- Web Services基盤を利用
- 最新版

- Date Issued: 30 September 2013
- Latest version:
<http://www.imslobal.org/lis/>
↓↓↓ (LINK) ↓↓↓
OneRoster and Learning Information Services
<https://www.imslobal.org/activity/onerosterlis>



<https://www.imslobal.org/lis/lisv2p0p1/LISSpecPrimerv2p0p1.html>
より引用

- Learning Information Services (LIS) のサブセット
- DLP、SIS、Learning Tool、Hosted Contentsを繋ぐ重要な役割。

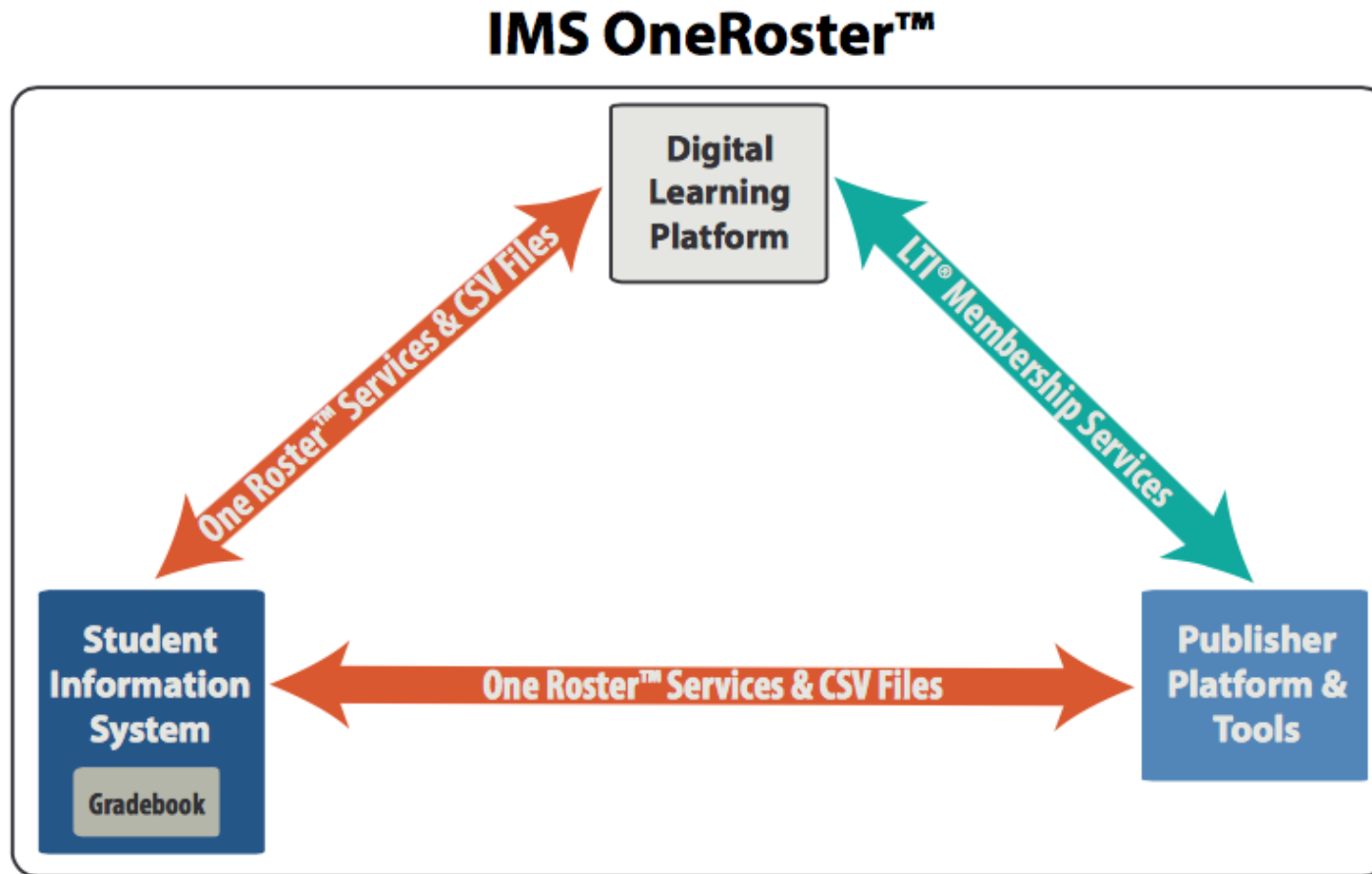


<https://www.imslobal.org/activity/onerosterlis> より引用

OneRoster (2/2)

<https://www.imsglobal.org/oneroster-rfp-guidance-checklist>

- Restful APIであるOne Roster Services が CSV で、SISとDLPや他のツールとデータ共有。



<https://www.imsglobal.org/oneroster-rfp-guidance-checklist> より引用

OneRosterのデータ定義の例 (組織, JSON)

■ 組織

```
{
  "org": {
    "sourcedId": "<sourcedId>"
    "status": "active | inactive | tobedeleted"
    "dateLastModified": "<最終変更日>"
    "name": "<組織名>"
    "type": "school | local | state | national"
    "identifier": "<組織の名称>"

    "metadata": {
      "ncesId": "0808120938"
    }

    "parent": {
      "href": "<親組織のURI>"
      "sourcedId": "<親組織のsourcedid>"
      "type": "org"
    }
  },
```

```
  "children": [
    {
      "href": "<子組織のURI>"
      "sourcedId": "<子組織のsourcedid>"
      "type": "org"
    },
    ...
    {
      "href": "<子組織のURI>"
      "sourcedId": "<子組織のsourcedid>"
      "type": "org"
    }
  ]
}
```

OneRosterのデータ定義の例 (年度・学期(academicSession), JSON)

■ 年度・学期

```
{
  "academicSession": {
    "sourcedId": "年度・学期(term)の  
sourcedId>"
    "status": "active | inactive | tobedeleted"
    "dateLastModified": "<最終変更日>"
    "title": "<学期名>"
    "startDate": "<年度・学期の開始日>"
    "endDate": "<年度・学期の終了日>"
    "type": "term"

    "parent": {
      "href": "<親年度・学期のURI>"
      "sourcedId": "<親年度・学期のsourcedId>"
      "type": "academicSession"
    }
  }
}
```

```
  "children": [{
    "href": "<1番目の子年度・学期のURI>"
    "sourcedId": "<1番目の子//のsourcedId>"
    "type": "academicSession"
  },
  {2番目の子年度・学期},
  {...}
]
```

OneRosterのデータ定義の例 (科目 (course), JSON)

■ 科目 (course)

```
{
  "course" : {
    "sourcedId" : "<sourcedId>"
    "status" : "active | inactive | tobedeleted"
    "dateLastModified" : "<最終更新日>"
    "metadata" : {
      "duration" : "<時間数>"
    }
    "title" : "<科目名>"
    "schoolYear" : {
      "href" : "<科目の所属年度のURI>"
      "sourcedId" : "<科目の所属年度の
sourcedId>"
      "type" : "academicSession"
    }
  }
}
```

```
  "courseCode" : "<科目コード>"
  "grade" : "<対象学年>"
  "subjects" : ["1番目の学科?", "2番目の学
科?" .. ]
// 例: ["chemistry", "physics"]
//      ["music", "drama, "poetry"]

  "org" : {
    "href" : "<所属組織>"
    "sourcedId" : "<所属組織のsourcedId>"
    "type" : "org"
  }
}
```

OneRosterのデータ定義の例 (クラス (class), JSON)

- クラス（科目が複数クラスに分かれている場合）

```
{
  "class" : {
    "sourcedId" : "<クラスのsourcedId>"
    "status" : "active | inactive | tobedeleted"
    "dateLastModified" : "<最終変更日>"
    "title" : "<クラス名>"
    "classCode" : "<クラスコード(組番号)>"
    "classType" : "homeroom | scheduled"
    "location" : "<場所>"
    "grade" : "<学年>"
    "subjects" : [<"1番目の学科", "2番目" ..>]

    "course" : {
      "href" : "<科目のURI>"
      "sourcedId" : "<科目のsourcedId>"
      "type" : "course"
    }
  }
}
```

```
  "school" : {
    "href" : "<学部のURI>"
    "sourcedId" : "<学部のsourcedId>"
    "type" : "org"
  }

  "terms" : [{
    "href" : "<1番目の学期のURI>"
    "sourcedId" : "<1番目の学期のsourcedId>"
    "type" : "academicSession"
  }, {2番目の学期}, ..
  ]
}
```

OneRosterのデータ定義の例 (ユーザ(user 学生・教員等), JSON)

■ ユーザ(学生・教員等)

```
{
  "user" : {
    "sourcedId" : "<sourcedid>"
    "status" : "active | inactive | tobedeleted"
    "dateLastModified" : "<最終更新日>"
    "username" : "<ユーザ名>"
    "userId" : "<ユーザID>"
    // active directory / lti user id / some other id
    "givenName" : "<名>"
    "familyName" : "<姓>"
    "role" : "teacher | student | parent | guardian
| relative | aide | administrator"
    "identifier" : "<学生番号、職員番号等>"
    "email" : "<電子メール>"
    "sms" : "<SMS>"
    "phone" : "<電話番号>"

    "agents" : [
      {
        "href" : "1番目の関係者(親等)のURI"
        "sourcedId" : "//sourcedid"
        "type" : "user"
      }, {2番目の..} ...]
  }
}
```

```
  "demographics" : {
    "href" : "<人的特徴(demographics data)の
URI>" // birthdate, sex, asian, ..
    "sourcedId" : "<上記のsourcedid>"
    "type" : "demographics"
  }

  "orgs" : [ {
    "href" : "<1番目の所属組織>"
    "sourcedId" : "<//のsourcedid>"
    "type" : "org"
  }, {2番目の所属組織}, ... ]
}
```

OneRosterのデータ定義の例 (評価項目? (Line Item), JSON)

■ 評価項目?のカテゴリ (Line Item Categories)

```
{
  "category" : {
    "sourcedId" : "<カテゴリのsourcedId>"
    "status" : "active | inactive | tobedeleted"
    "dateLastModified" : "<最終更新日>"
    "title" : "<カテゴリ名>"
  }
}
```

■ 評価項目? (Line Item)

```
{
  "lineitem" : {
    "sourcedId" : "<sourcedId>"
    "status" : "active | inactive | tobedeleted"
    "dateLastModified" : "<最終更新日>"
    "title" : "<名>"
    "description" : "<に関する説明>"
    "assignDate" : "<登録日>"
    "dueDate" : "<締切日>"
    "category" : {
      "href" : "<カテゴリのURI>"
      "sourcedId" : "<//のsourcedId>"
      "type" : "category"
    }
  }
}
```

```
"class" : {
  "href" : "<クラスのURI>"
  "sourcedId" : "<クラスのsourcedId>"
  "type" : "class"
}
```

```
"gradingPeriod" : {
  "href" : "<評価期間>"
  "sourcedId" : "<//のsourcedId>"
  "type" : "academicSession"
}
```

```
"resultValue" : {
  "min" : "<評価値の最小値(実数)>"
  "max" : "<評価値の最大値(実数)>"
}
}
```

OneRosterのデータ定義の例 (成績 (Result), JSON)

■ 成績 (Result)

```
{
  "result" : {
    "sourcedId" : "<sourcedid>"
    "status" : "active | inactive | tobedeleted"
    "dateLastModified" : "<最終更新日>"
    "lineitem" : {
      "href" : "<評価項目?(lineitem)のURI>"
      "sourcedId" : "<//のsourcedId>"
      "type" : "lineitem"
    }
  }

  "student" : {
    "href" : "学生のURI"
    "sourcedId" : "<//のsourcedId>"
    "type" : "user"
  }

  "score" : "<成績スコア>"
  "resultstatus" : "not submitted | submitted |
partially graded | fully graded | exempt"
  "date" : "<最終更新日>"
  "comment" : "<コメント>"
}
```


RESTインターフェース (抜粋)

[https://hostname/learningdata/v1/*\[?+\]](https://hostname/learningdata/v1/*[?+])

Service Call	Endpoint(*)	Endpoint GET(read)
getOrgs	/orgs	全組織を返す
getOrg	/orgs/{id}	(source) idで指定された組織
getCourse	/courses/{id}	idで指定された科目情報
getUser	/users/{id}	idで指定されたユーザ情報
getStudents	/students	全学生情報
getStudentsForClassInSchool	/schools/{id}/classes/{id}/students	指定学部の指定クラスの全学生
getClassesForCourse	/courses/{id}/classes	指定されたコースの全クラス
getResultsForClass	/classes/{class_id}/results	指定されたクラスの全成績
getResultsForLineItemForClass	/classes/{class_id}/lineitems/{li_id}/results	指定クラスの指定評価項目の全学生の成績
getResultsForStudentsForClass	/classes/{id}/students/{id}/results	指定クラスの指定学生の全成績

パラメータ (+)	説明
(規定値limit=100&offset=0) limit=10 limit=10&offset=10	Pagination 最初の10データ 次の10データ
sort=familyName&orderBy=asc sort=score&orderBy=desc	Sorting 姓の昇順 成績のよい順
givenName='Yuji' (givenName=%3D%27Yuji%27) dateLastModified>'2015-01-01' (.ed%3E%272015%3D01-01%27)	Filtering 名がYuji (encoded) 更新日が..より後 (encoded)
fields=givenName,familyName	Field Selection 姓名のみ

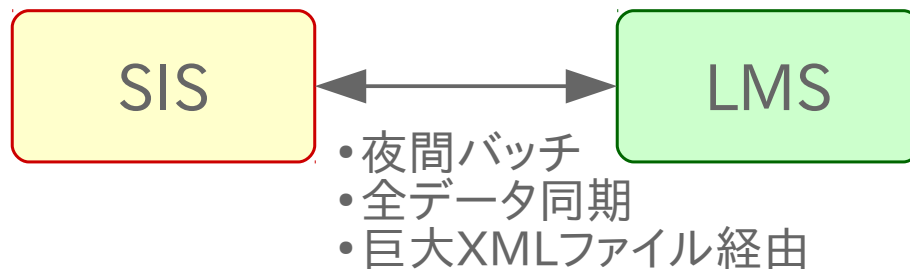
■ セキュリティ

- SSL
- OAuth1.0

IMS LTI同様、2-legged OAuth authorization

Enterprise → OneRoster で何が変わった？

■ Enterprise

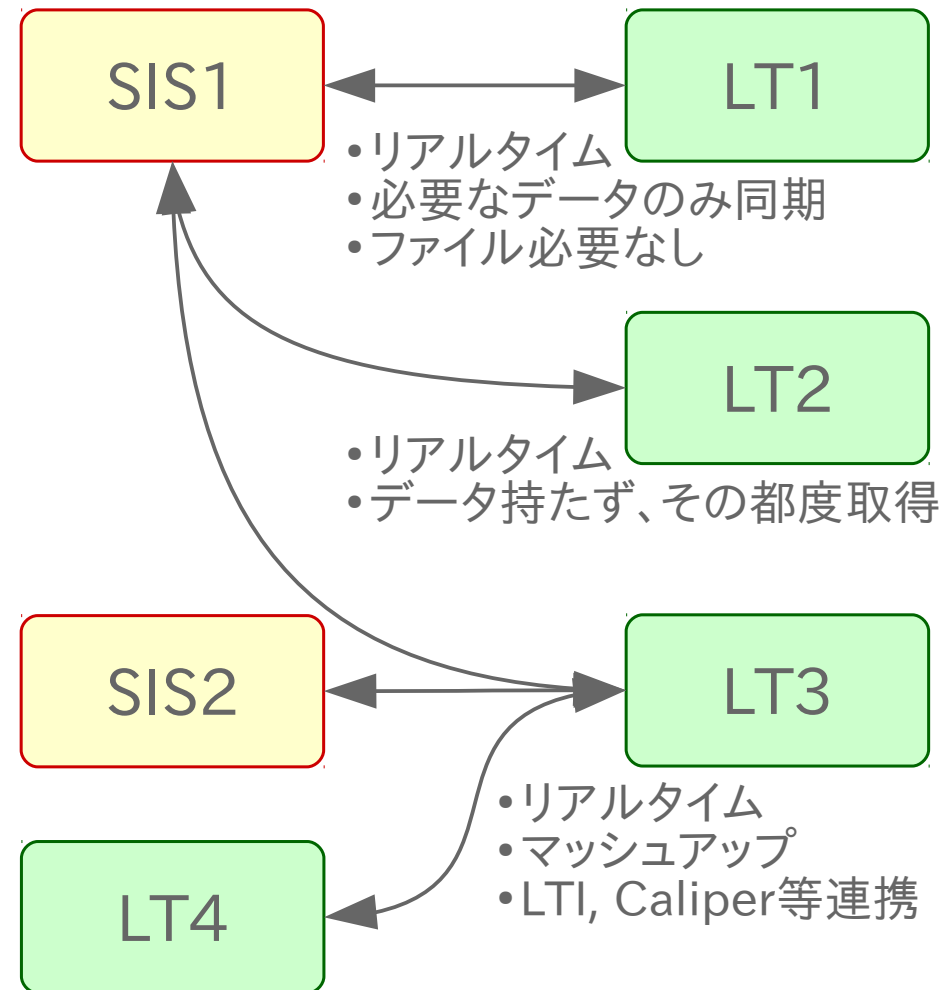


■ キーポイント

- 標準規格であること。
- WebAPI (REST) 化と認証連携で、
 - ◆ ファイルを介さず
 - ◆ 必要なデータのみ
 - ◆ マッシュアップ可能になった。

■ 実装コストとの兼ね合いか

■ One Roster



- 基盤として、SPA + Web API + 標準化
 - SPA (Single Page Application), RESTful Web API (JSON / JSONP)
- IMS Enterpriseによる学籍情報、履修情報の共有
 - 大学ポータルを中心としたサービスとシステム間連携の必要性
 - 統合IDとSSO、科目情報&学生の基本・履修情報等の共有の主な流れ
 - IMS Enterpriseの例 (学生、教職員の基本情報, 科目情報)
- IMS LISとOneRosterによる動的連携への発展の期待
 - Learning Information Services (LIS)
 - OneRoster
 - ◆ データ定義の例 (組織、年度・学期、科目、クラス、ユーザ、成績等)
 - ◆ RESTインターフェース (抜粋)
 - ◆ Enterprise → OneRosterで何が変わった?