

# IMS LTI&Caliper ハツカソン

- LTI編 -

2017年 8月 22日  
株式会社 内田洋行

---

# アジェンダ

---

1. LTIの概要
2. サンプルプログラムの概要
3. サンプルプログラム解説・デモ (PHP)
4. サンプルプログラム解説・デモ (Java)

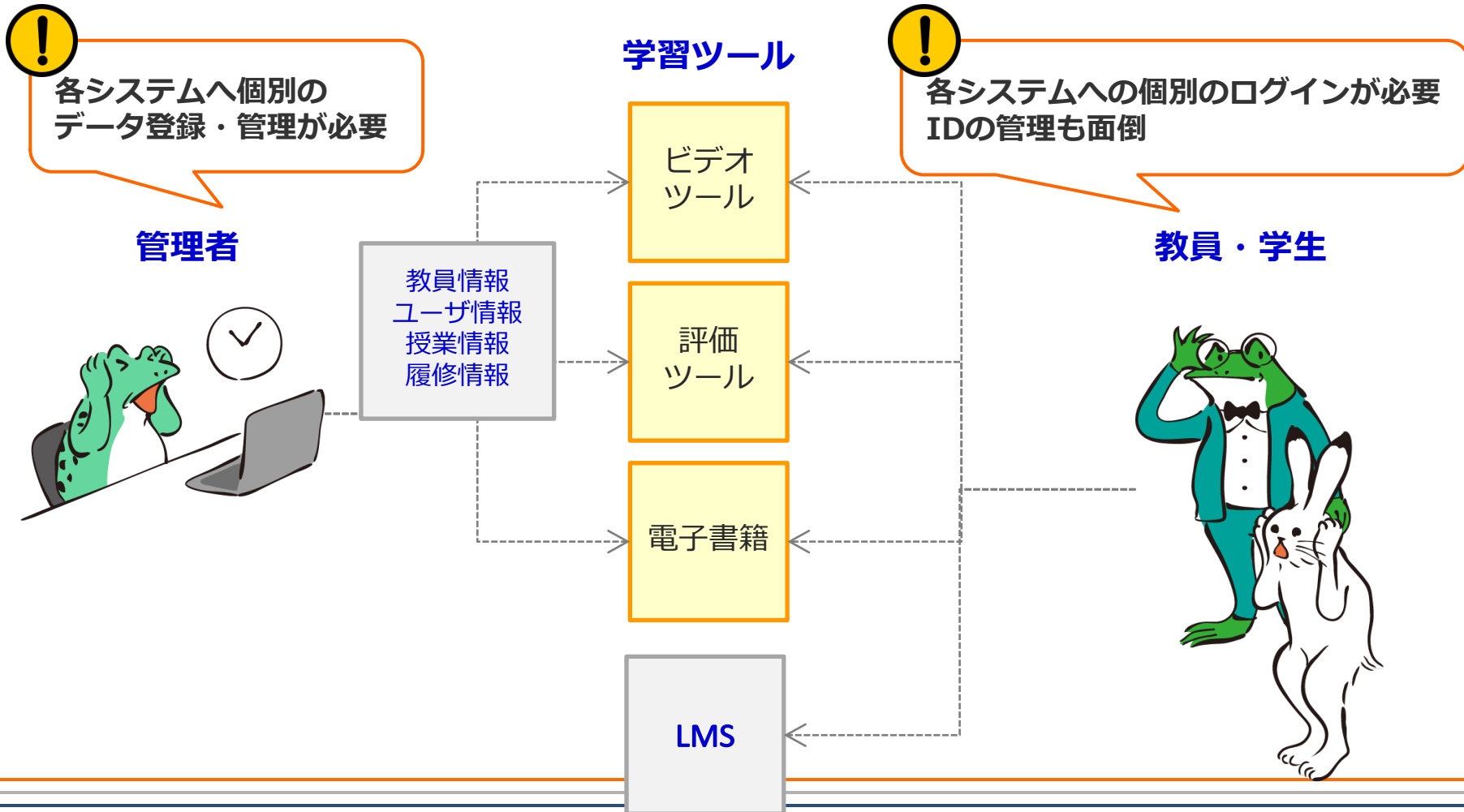
# LTIの概要 – 背景

より“濃い”授業を行うために、よいツールがあれば活用したい...



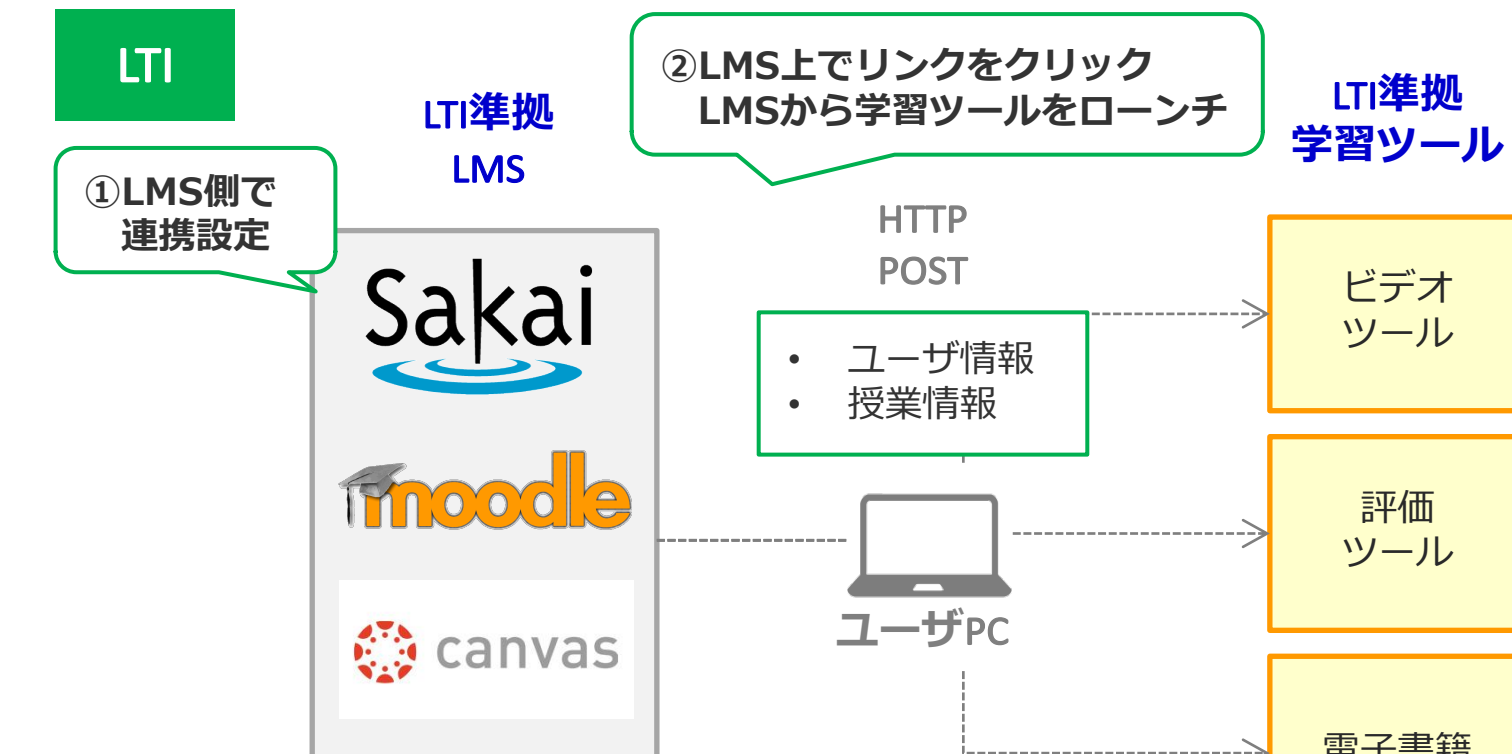
# LTIの概要 – 背景

よいツールがあっても、そのまま使おうとすると管理者にも利用者にも負担になってしまうことも...



# そこでLTI

- LTI = Learning Tools Interoperability
- LMSと学習ツールのインターフェース



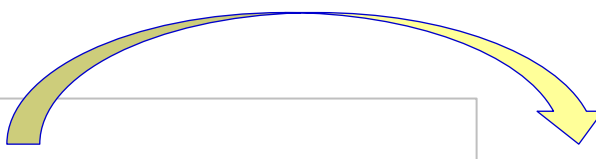
- 個別対応不要。簡単な設定のみで学習ツールと連携させられる
- ユーザはLMSにのみログイン。各学習ツールはLMS上のコンテンツとして見える

# LTIローンチで渡されるデータ

LTI準拠 LMS



LTI準拠 学習ツール



```
lti_message_type=basic-lti-launch-request
lti_version=LTI-1p0

user_id=0163da3b-f67f-4cf3-945f-92c8ed434e21
user_image=http://c22.media.hosei.ac.jp/direct/profile/...
roles=Instructor

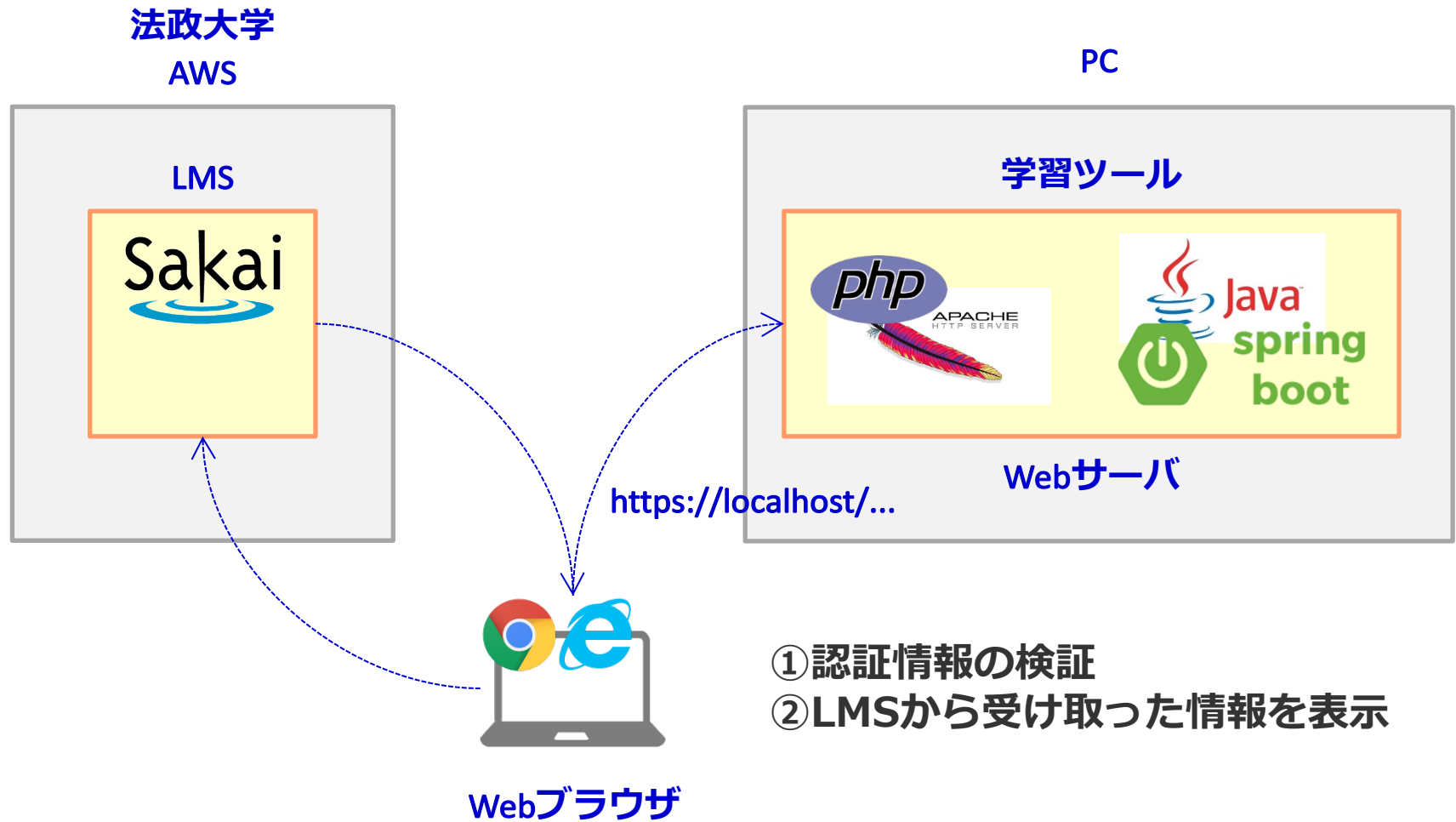
context_id=c3947b99-5613-4825-b5bb-d273884422b6
context_label=uchida
context_title=uchida
context_type=Group

resource_link_description=LTIbasic_kubo
resource_link_id=fcc2a280-0867-43dc-b463-a296e0d48b48
resource_link_title=LTIbasic_kubo

launch_presentation_css_url=http://c22.media.hosei.ac.jp/.../tool_base.css
launch_presentation_locale=ja_JP
launch_presentation_return_url=http://c22.media.hosei.ac.jp/imsblis/service/...
```

# サンプルプログラムの概要

- LTIのバージョンはv1.1 (Basic LTI) を使用



# PHP開発環境例（Windows）

---

- XAMPP (v.7.1.7)
  - <http://www.apachefriends.org/jp/index.html>
- NotePad++ (v.7.5)
  - <http://notepad-plus-plus.org/>



# PHP動かしか方・デモ

---

## <学習ツール: ローカルHTTPサーバー>

- 入手したサンプルコードをローカルのHTTPサーバーへコピー
  - htdocsに任意の名称でフォルダーを作成
- HTTPサーバーを起動

## <LMS: Sakai>

- Sakaiにサンプルの学習ツールを登録

# PHP動かし方・デモ

Sakai

マイワークスペース

uchida ▼

uchida > LTIサンプル (藤原)

access Viewに入る

ホーム

お知らせ

リソース

サイト情報

名簿

LTIサンプル (藤原)

キャンセル

保存した設定をクリア

**必須情報**

\*リモートツールの URL:

\*リモートツールのキー:

\*リモートツールのシークレット

**表示情報**

ボタンテキストをセット:  (ツールメニュー中のテキスト)

ツールタイトルをセット:  (ツール上部)

このツールのアイコンを選択してください:  ▼

**起動情報(任意)**

ローカルHTTPに配置したサンプルプログラムのURLを登録

OAuthのConsumer Key とSecret  
キーは「12345」  
シークレットは「secret」を入力

任意の名称を入力

# ソースコード解説 (PHP)

helloworld.php

有効なツールかどうかの結果が格納

```
1 <?php
2 error_reporting(E_ALL & ~E_NOTICE);
3 ini_set("display_errors", 1);
4
5 require_once 'ims-blti/blti.php';
6 $lti = new BLTI("secret", false, false);

session_start();
header('Content-Type: text/html; charset=utf-8');
?>
```

ライブラリを参照し、BLTIインスタンスを生成  
secretを登録

```
11
12 <!DOCTYPE html>
13 <html>
14 <head>
15 <meta charset="UTF-8" />
16 <title>Building Tools With The Learning Tools Operability Specification</title>
17 </head>
18
19 <body>
20 <?php
21 if ($lti->valid) {
22 <?>
23 <h2>Hello, World!</h2>
24 <p>We have implemented a basic LTI tool!</p>
25 <h3>A basic dump of POST parameters:</h3>
26 <pre>
27 <?php
28 foreach($_POST as $key => $value) {
29     print "$key=$value\n";
30 }
31 <?>
32 </pre>
33 <?php
34 } else {
35 <?>
36 <h2>This was not a valid LTI launch</h2>
37 <p>Error message: <?= $lti->message ?></p>
38 <?php
39 }
40 <?>
41 </body>
```

有効なツールである場合、  
POSTに格納されるパラメーターを出力

# Java 開発環境（推奨）

---

- Eclipse IDE for Java EE Developers
  - <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/oxygenr>
- JDK 1.8
  - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
    1. インストーラを実行してインストール
    2. 環境変数の追加
      - JAVA\_HOMEにJDK のパスを追加
        - 例：C:¥Program Files¥Java¥jdk1.8.0\_05
- Maven
  - <https://maven.apache.org/download.cgi>
    1. ダウンロードしたファイルを解凍し、任意の場所に配置
    2. 環境変数の追加
      - 環境変数PATHにMaven の binフォルダを追加
        - 例：C:¥apache-maven-3.3.3¥bin

# ソースコード解説 (Java)

## ① 認証情報の検証

### MockKeyService.java

```
6 @Service
7 public class MockKeyService implements LtiKeySecretService {
8
9     @Override
10    public String getSecretForKey(String key) {
11        return "secret";
12    }
13
14 }
```

本来であれば、渡されたキーに対応するSecretを  
DBや認証サーバに問い合わせ取得し、戻り値で渡す  
※今回はサンプルのため"secret"で固定

# ソースコード解説 (Java)

## ②LMSから受け取った情報を表示 (成功／失敗のみ)

### LtiController.java

```
@Lti
@RequestMapping(value="/launch", method=RequestMethod.POST)
public String ltiEntry(HttpServletRequest request, LtiVerificationResult result) {
    if(!result.isSuccess()) {
        return "error";
    }
    LtiLaunch launch = result.getLtiLaunchResult();
    System.out.println("=== LtiLaunch ===");
    System.out.println("context_id: "+launch.getContextId());
    System.out.println("resource_link_id: "+launch.getResourceLinkId());
    System.out.println("user_id: "+launch.getUser().getId());
    System.out.println("user_role: "+launch.getUser().getRoles());

    System.out.println("=== HttpServletRequest ===");
    //sort
    Set<Entry<String, String[]>> entrySet = request.getParameterMap().entrySet();
    List<Entry<String, String[]>> l = new ArrayList<Entry<String,String[]>>(entrySet);
    Collections.sort(l, (e1, e2)->e1.getKey().compareTo(e2.getKey()));

    for(Entry<String, String[]> entry : l) {
        System.out.println(entry.getKey()+" : "+Arrays.toString(entry.getValue()));
    }
    return "success";
}
```

認証情報の検証結果

認証に失敗していた場合、画面に“error”と表示

LTIローンチリクエストで渡された  
各種パラメータ

とりあえずコンソール(標準出力)に出力

認証に成功していた場合、画面に“success”と表示

# ソースコード解説 (Java)

## ②LMSから受け取った情報を表示 (画面つき・各パラメータ表示)

### LtiWebController.java

```
38      LtiLaunch launch = result.getLtiLaunchResult();
39
40      Map<String, String> ltiParams = new HashMap<String,String>();
41      ltiParams.put("context_id", launch.getContextId());
42      ltiParams.put("resource_link_id", launch.getResourceLinkId());
43      ltiParams.put("user_id", launch.getUser().getId());
44      List<String> roles = launch.getUser().getRoles();
45      model.addAttribute("ltiParams", ltiParams);
46      model.addAttribute("roles", roles);
47
```

LTIローンチリクエストで渡された  
各種パラメータを  
Web画面に渡す

```
58
59      model.addAttribute("reqParams", l);
60
61      System.out.println("== HttpServletRequest ==");
62      for(Entry<String, String[]> entry : l) {
63          System.out.println(entry.getKey()+" : "+Arrays.toString(entry.getValue()));
64      }
65
66      return "index";
67  }
```

Web画面(index.html)を表示

# Java動かしか方・デモ

1. Windowsであればコマンドプロンプト, Macであればターミナルでサンプルプロジェクトの直下に移動
2. サンプルプログラムの実行パッケージを作成

```
$ mvn package
```

3. targetディレクトリの中に, サンプルプログラムの実行パッケージが  
できている

```
$ ls ./target/  
classes                      maven-archiver  
demo-0.0.1-SNAPSHOT.jar      maven-status  
demo-0.0.1-SNAPSHOT.jar.original  surefire-reports  
generated-sources             test-classes  
generated-test-sources
```



# Java動かしか方・デモ

---

4. サンプルプログラム実行パッケージを実行し, 学習ツール (Spring Bootサーバ) を起動

```
$ java -jar target/demo-0.0.1-SNAPSHOT.jar
```

5. ブラウザでSakaiからアクセス, 動作を確認
  - LtiController>ltiEntry: <http://localhost:8080/launch>
  - LtiWebController>ltiEntryWeb: <http://localhost:8080/launchweb>
6. Ctrl +C でサーバを終了

# ハッカソン おすすめ基本編

- まずSakaiからLTIで学習ツールを呼び出し
- 学習ツールにCaliperを組み込み、学習イベントを送信してみる

