
PHPおよびJavaによるCaliper実装

- IMS LTI & Caliper ハッカソン -



株式会社ネットラーニングホールディングス

技術本部長 李 在範

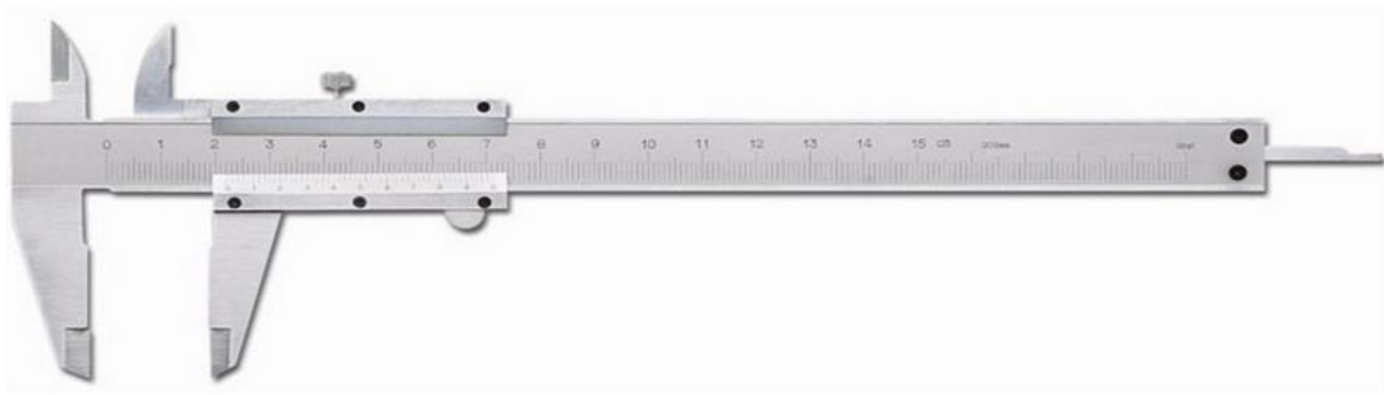
技術本部 杉山 秀則

NetLearning® Quality.

Caliper Analytics とは？

IMS GLOBALが策定を進める

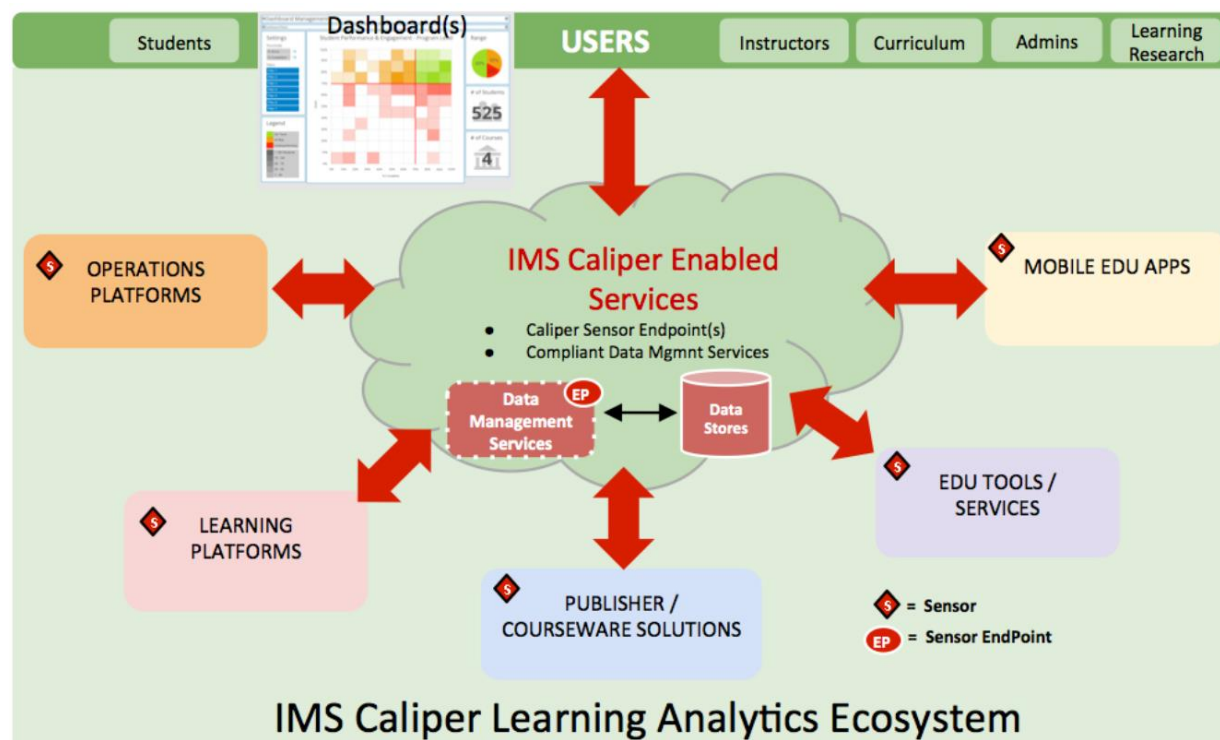
“オンライン学習測定フレームワーク”



- 学習の測定方法の標準
- 様々な学習環境からデータ（学習履歴）を収集して分析等を実現

Caliper Analyticsの価値

学習履歴を標準化するより、様々な学習環境からデータを収集して分析することが可能となる



学習データ（学習履歴）に基づいた教育内容の改善を実現

Caliper Analyticsの対応状況

- 認証取得

—18件

NetLearning®

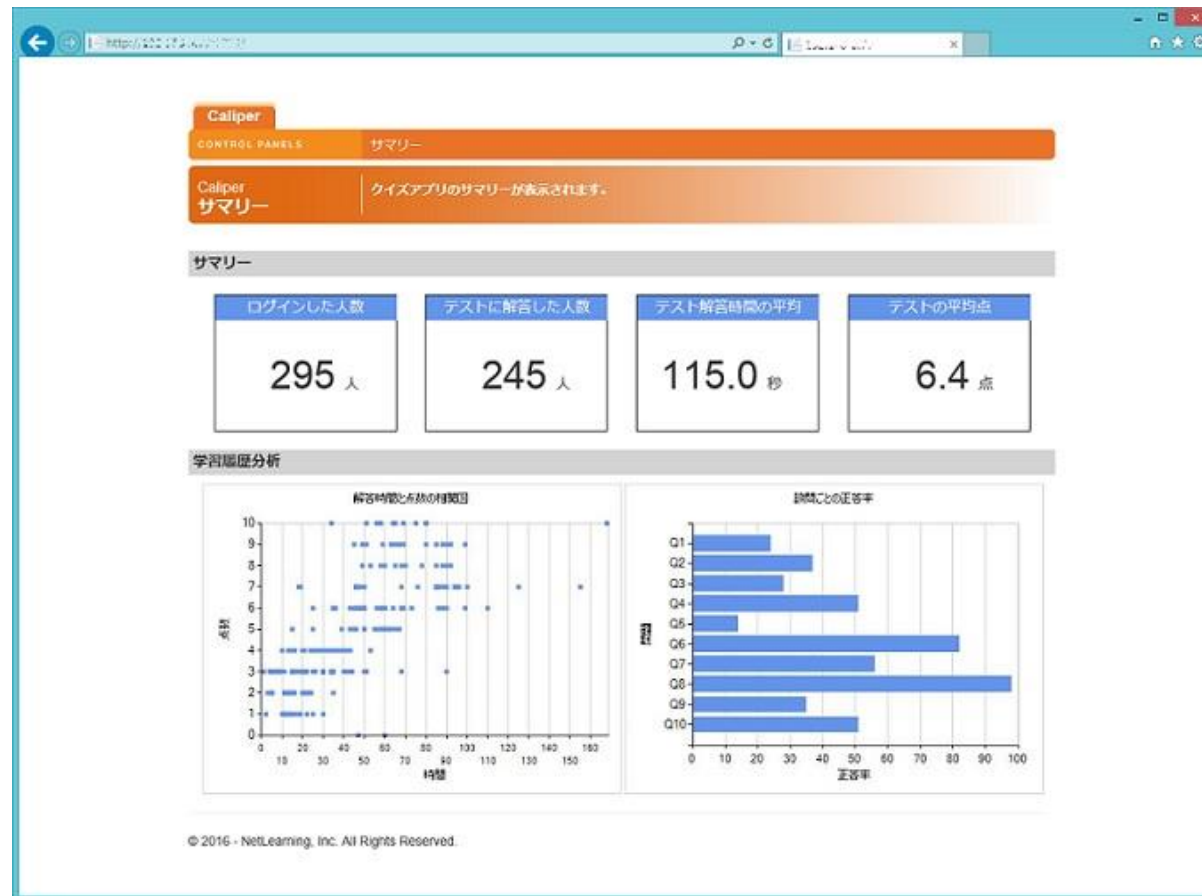
Blackboard Learn™



D2L

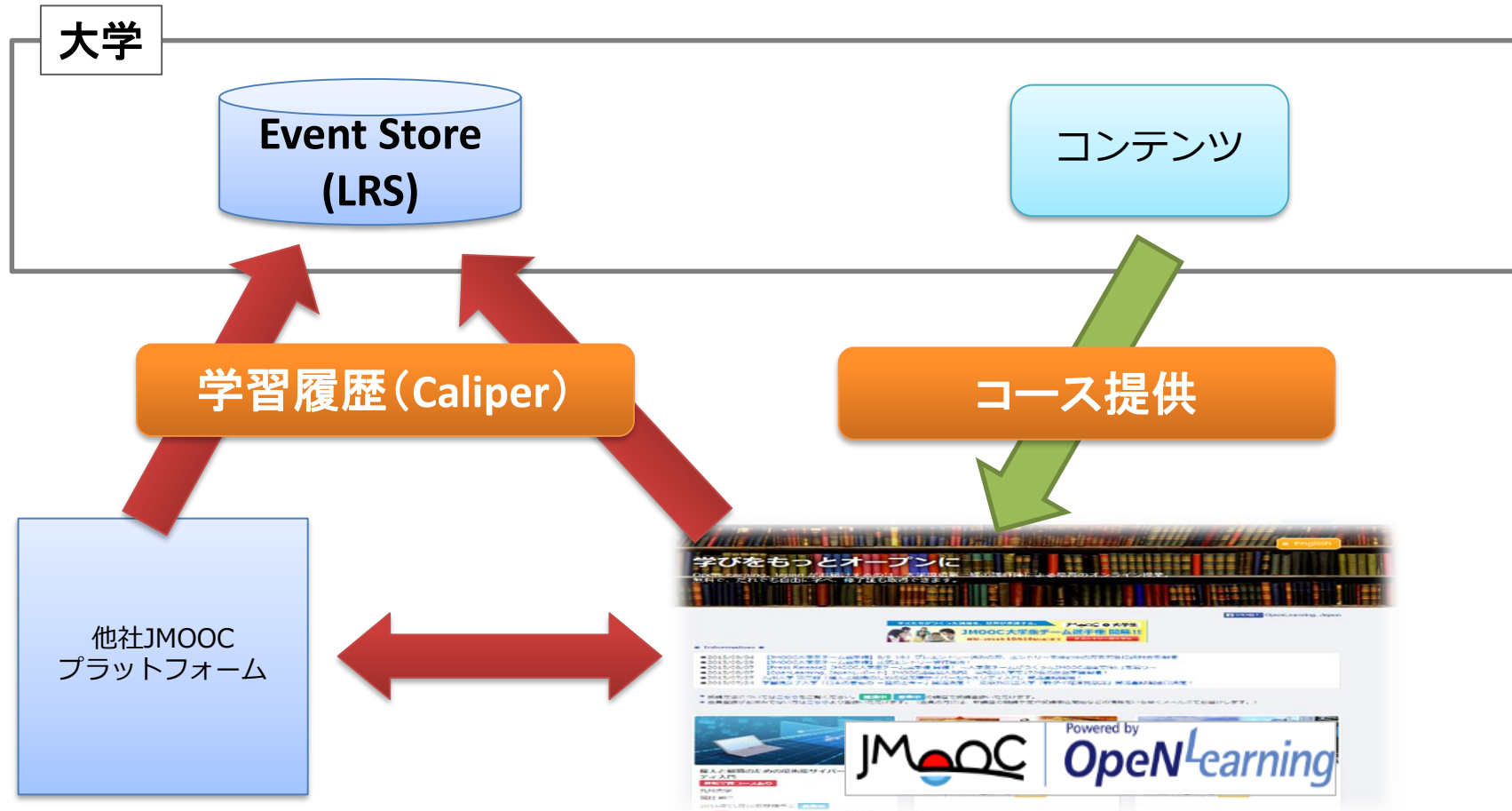
事例：ダッシュボード

- 学習状況をリアルタイムで表示

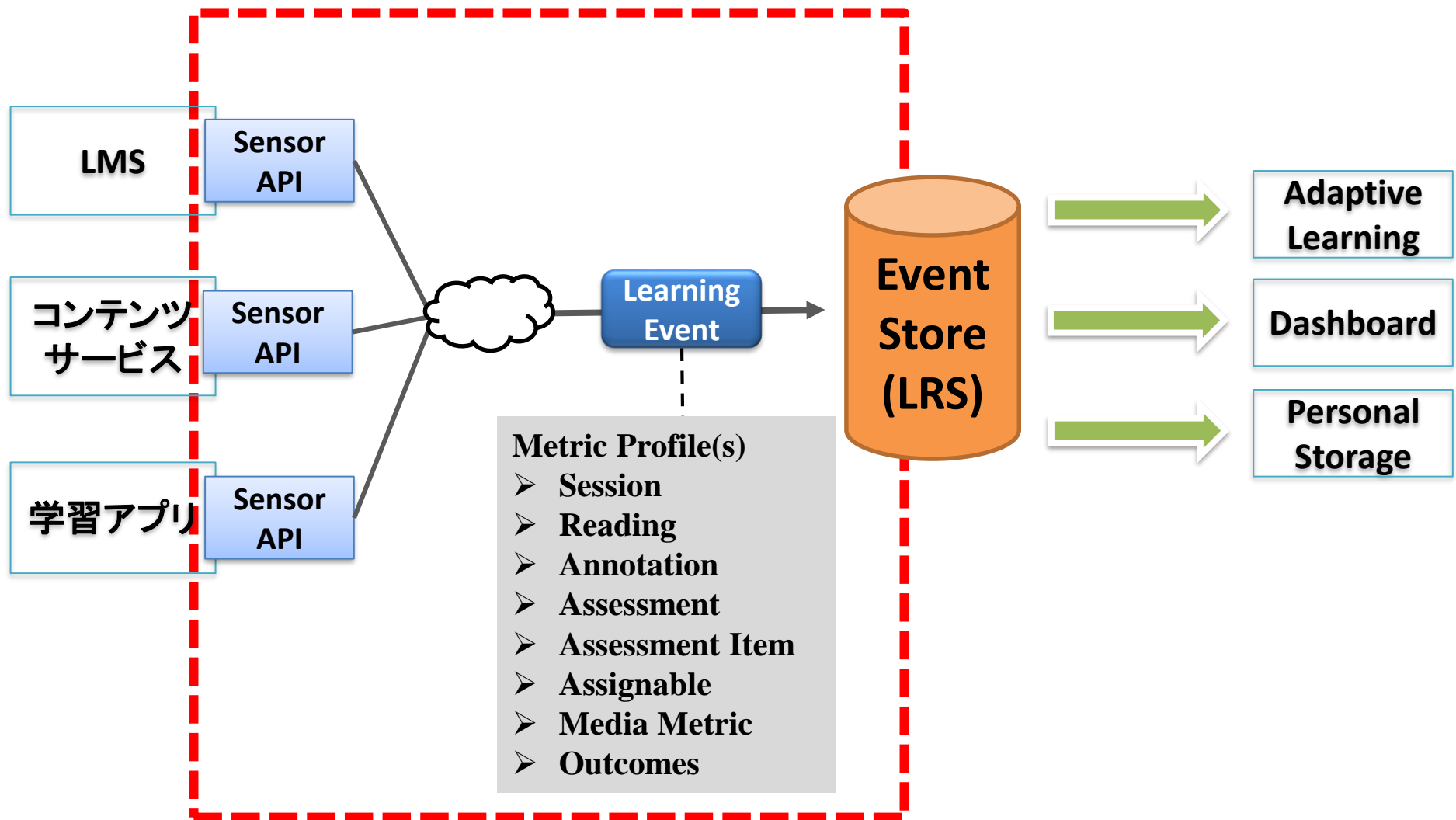


事例：JMOOCでの学習履歴連携

- コース提供元との学習履歴共有



Caliper Analytics 1.xの範囲



Caliper Analytics 1.xの構成要素

- Metric Profiles ... 学習履歴の標準化
- Sensor API ... 学習履歴の回収
- Event Store ... 学習履歴の蓄積

Caliper - Metric Profiles

学習履歴として何を記録するのかを定義

Reading

コンテンツ(EPUB3)の
閲覧、検索、遷移...

Annotation

ブックマーク、ハイライト、
タグ付け、シェア...

Assessment (Item)

クイズ、ホームワーク、
テスト...

Media

音声、動画コンテンツ...

Session

ログイン、ログアウト...

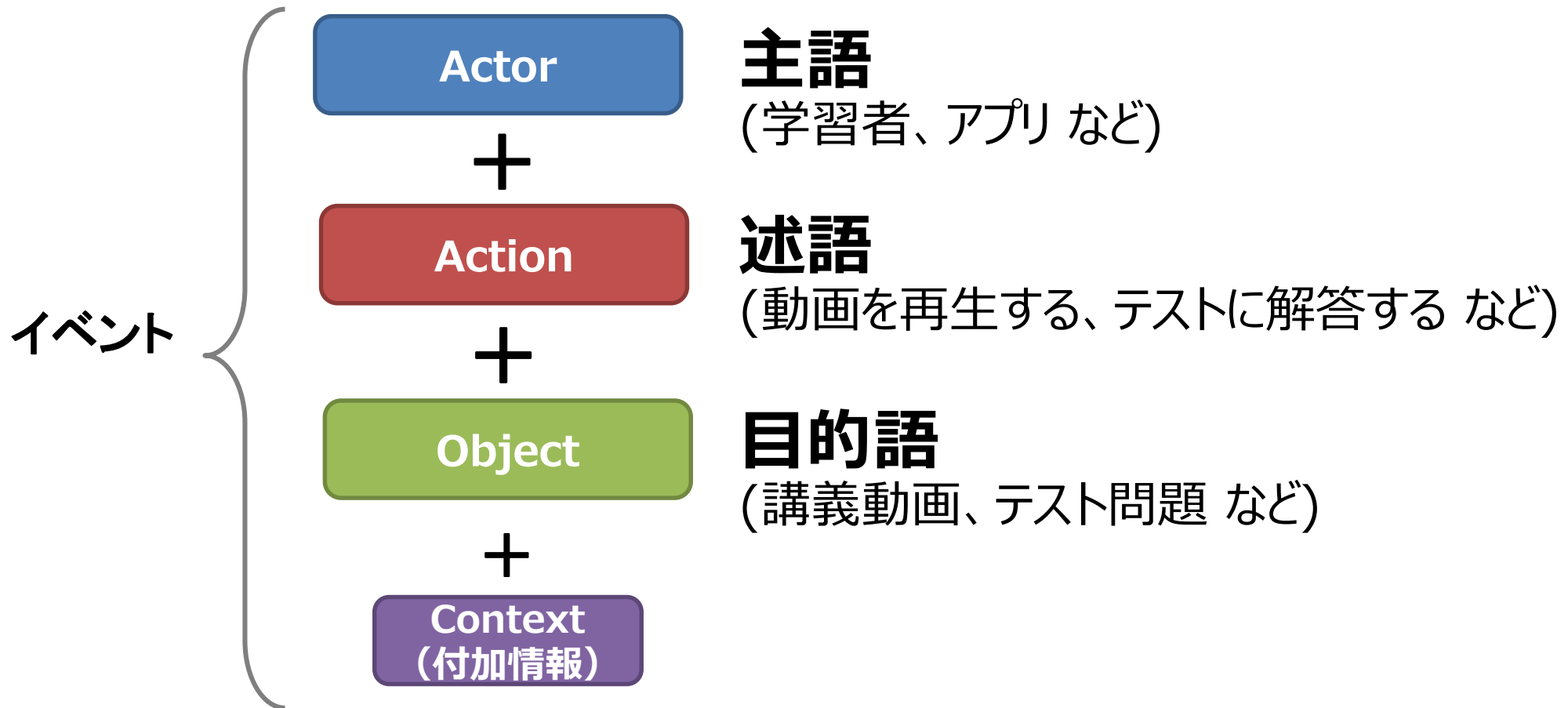
Assignable

アサイン、アクティベート...

Outcome

学習結果、成績

Metric Profileに従って記録される学習行動



Caliper – イベント例

学習者のイベントの例

Actor

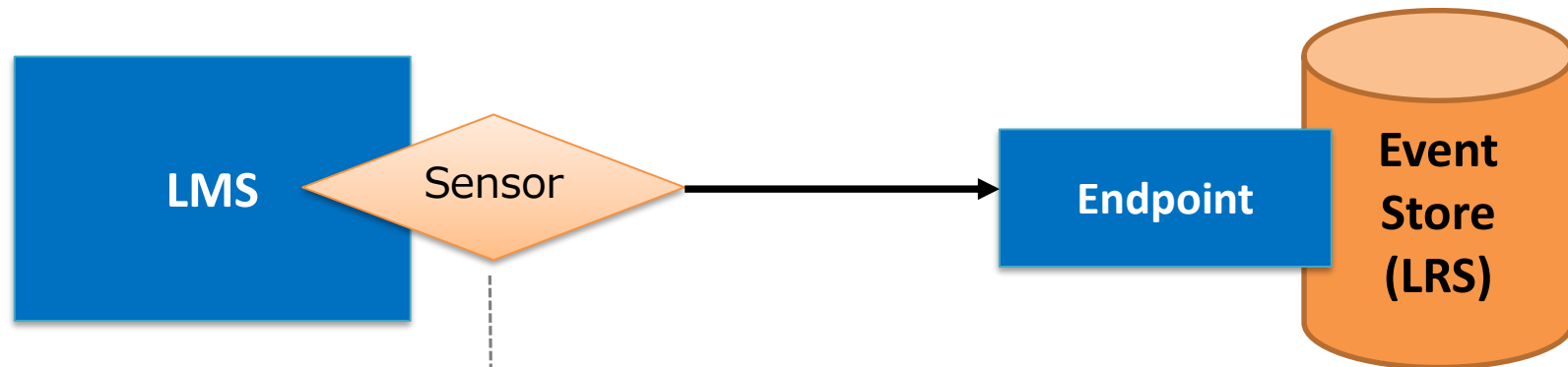
Action

Object

#	学習活動	Profile	Actor	Action	Object
1	アプリにログインした	Session	学習者	Logged In	学習App
2	説明ページに遷移した	Navigation	学習者	Navigated to	遷移先のページ
3	説明ページにタグ付けした	Annotation	学習者	Tagged	タグの内容
4	クイズページに遷移した	Assessment	学習者	Started	クイズページ
5	設問に遷移した	Assessment Item	学習者	Started	設問
6	設問に答えた	Assessment Item	学習者	Completed	設問
7	#5-6を繰り返す				
8	解答を送信した	Assessment	学習者	Submitted	クイズページ
9	スコアが表示された	Outcome	学習App	Graded	成績
10	アプリをログアウトした	Session	学習者	Logged Out	学習App

Sensor API

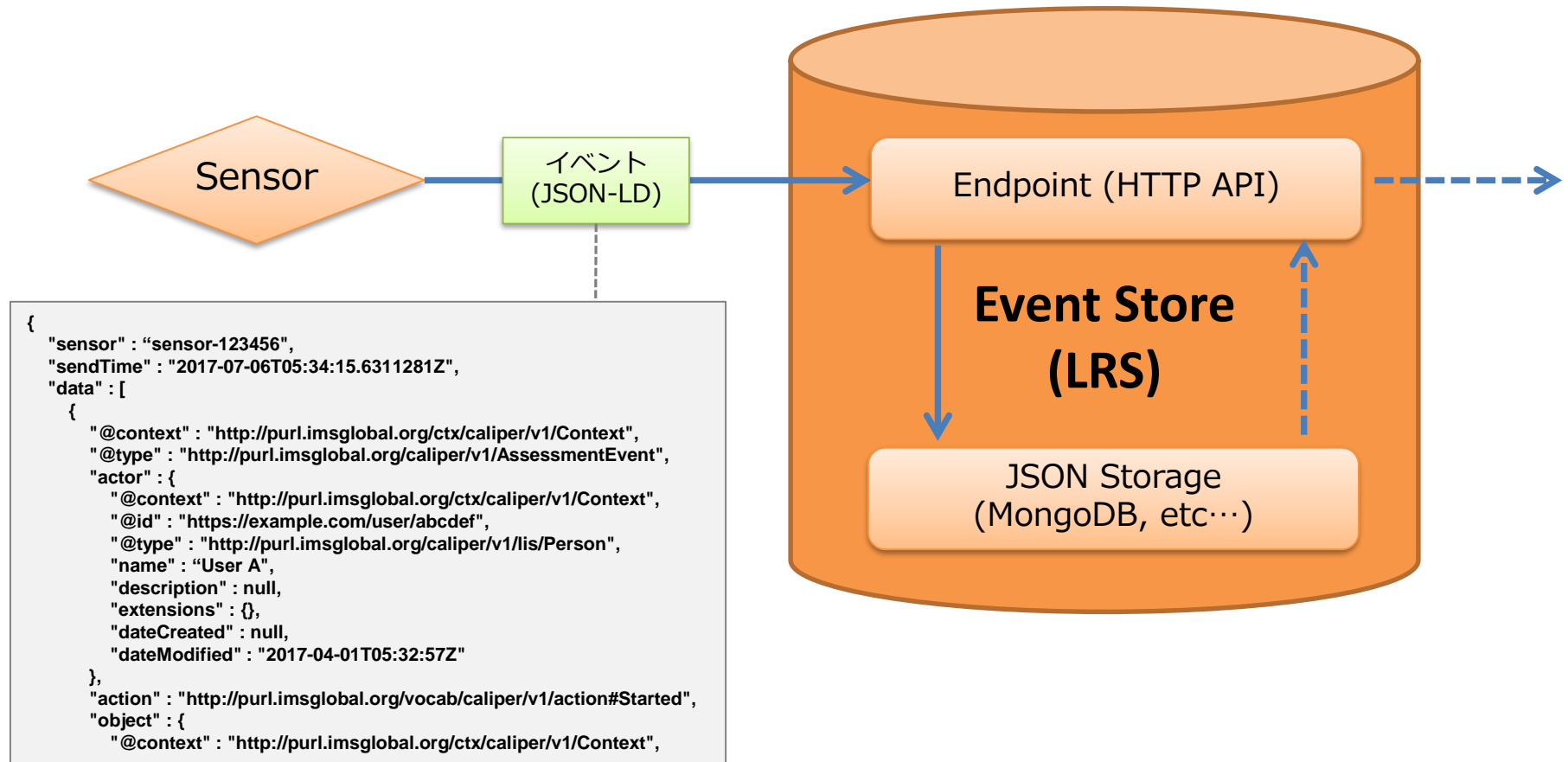
- 学習履歴が生成される場所にSensorを設置
 - SensorがイベントをEventStoreに送信する



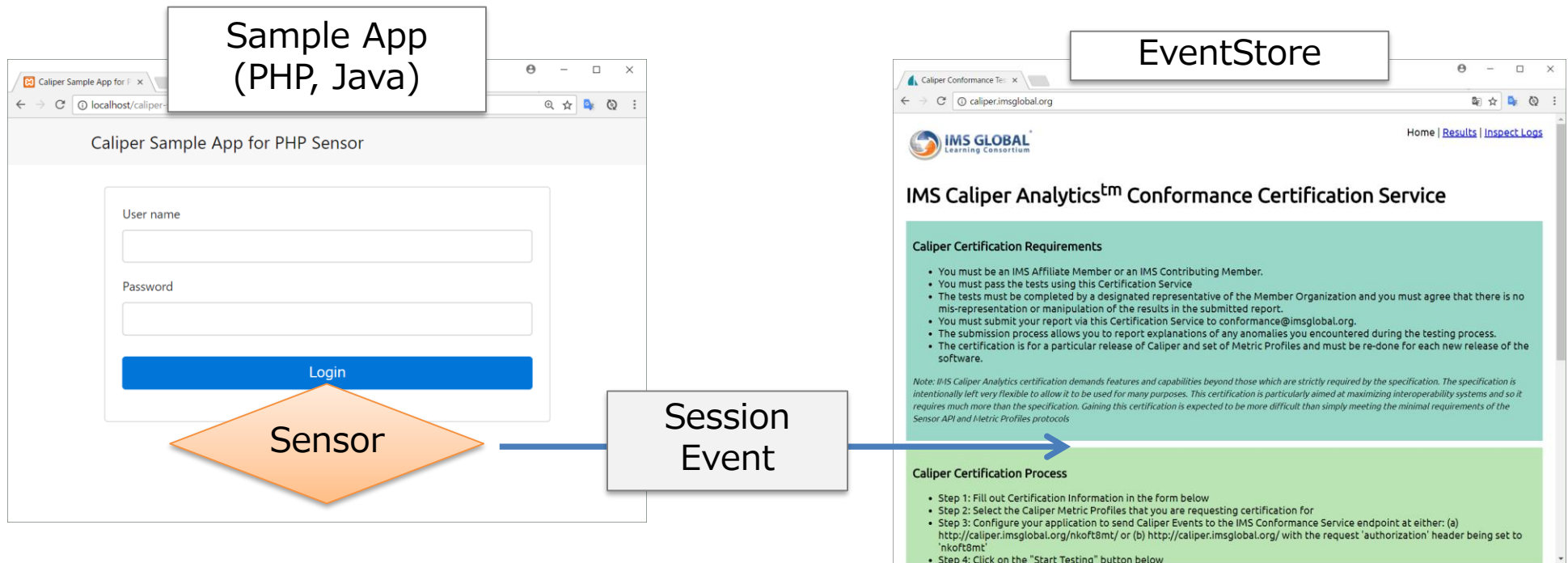
```
var sensor = new CaliperSensor("sensor-id");  
...  
sensor.RegisterEndpoint(  
    new CaliperEndpointOptions("https://caliper.example.com/events" ));  
...  
sensor.SendAsync( mediaEvent );
```

Event Store

- Sensorから送られたイベント(JSON-LD形式)を蓄積



サンプルプログラムの構成



- 今回はEventStoreとしてIMSのCaliper Conformance Testing (認証取得用試験ページ)を利用する。
 - <http://caliper.imsglobal.org/>

(PHP)ソースコード解説 1

ログイン画面での処理

```
<?php

// セッション開始
session_start();

error_reporting(E_ALL & ~E_NOTICE);
ini_set("display_errors", 1);

// SessionEvent送信関数の読み込み
require_once(' ../lib/caliper-session.php');

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    $user_name = $_POST['user_name'];
    $login_time = new DateTime();

    if(sendSessionLoggedIn($user_name, $login_time)) {
        // イベント送信成功時に表示するメッセージ
        $_SESSION['message'] = 'SessionEvent(Logged In) was sent successfully.';
    } else {
        // イベント送信失敗時に表示するメッセージ
        $_SESSION['message'] = 'Failed to send SessionEvent(Logged In).';
    }

    $_SESSION['user_name'] = $user_name;
    $_SESSION['login_time'] = $login_time;

    header('Location: ../logout.php');
    exit;
}
```

イベント送信処理を呼び出す。
成功／失敗に応じて異なるメッセージを表示する。

ログアウト時に利用するため、
セッションにユーザ名とログイン
に日時を保存する。

login.php

(PHP)ソースコード解説 2

Sensorの準備

/* 設定項目。任意の情報に書き換えること */

```
$endpoint = 'http://caliper.imsglobal.org/'; // 送信先URL  
$apikey   = 'apikey'; // APIキー  
$sensorId = 'sensorId';  
$clientId = 'clientId';  
$appId    = 'https://example.com/app/123456789';  
$appName  = 'IMS-JS Caliper';
```

送信先Event StoreのURL,
apikeyなどを設定

```
// Sensorのインスタンス作成  
$sensor = new Sensor($sensorId);
```

Sensorのインスタンスを作成

```
$options = (new Options())  
->setApiKey($apikey)  
->setDebug(true)  
->setHost($endpoint);
```

Sensorに送信先を登録

```
// Sensorインスタンスの設定  
$sensor->registerClient('http', new Client($clientId, $options));
```

```
// このアプリの情報の設定  
$app = (new entities\agent\SoftwareApplication($appId))  
->setName($appName);
```

caliper-session.php

(PHP)ソースコード解説 3

```
function sendSessionLoggedIn($user_name, $login_time)
{
    global $sensor, $app;

    $actor = (new entities\agent\Person('https://example.com/users/' . $user_name))
        ->setName($user_name);

    $session = (new entities\session\Session('https://example.com/sessions/12345'))
        ->setActor($actor)
        ->setDateCreated($login_time)
        ->setDateModified($login_time)
        ->setStartedAtTime($login_time);

    $event = (new events\SessionEvent())
        ->setActor($actor)
        ->setAction(new actions\Action(actions\Action::LOGGED_IN))
        ->setObject($app)
        ->setGenerated($session)
        ->setEventTime($login_time);

    // イベント送信
    try {
        $sensor->send($sensor, $event);
        return true;
    } catch (\RuntimeException $exception) {

        echo 'Error sending event: ' . $exception->getMessage() . PHP_EOL;
        return false;
    }
}
```

caliper-session.php

LoggedInイベントの送信

送信する情報を表す
Entity(Person, Sessionなど)
を構築する。

Sessionイベントを作成する。
イベントのActor, Objectとし
て、直前に構築したEntityを
設定する。

イベント送信の実施

(PHP)サンプルプログラムの動かし方

1. caliper-session.phpのAPIキーを修正する。
2. publicフォルダを外部からアクセス可能とする
(サンプルプログラムをhtdocs以下に設置など)
3. public/login.phpをWebブラウザで開く

(Java)ソースコード解説 1

```
package imsjava.caliper;

import java.io.IOException;

@WebServlet(
    name = "LoginServlet",
    urlPatterns = {"/login"}
)
public class LoginServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String userName = request.getParameter("user_name");
        DateTime loginTime = DateTime.now();

        // ログインイベントの送信
        if (CaliperSession.sendSessionLoggedIn(userName, loginTime)) {
            // イベント送信成功
            request.getSession().setAttribute("message", "SessionEvent(Logged In) was sent successfully.");
        } else {
            // イベント送信失敗
            request.getSession().setAttribute("message", "Failed to send SessionEvent(Logged In).");
        }

        request.getSession().setAttribute("user_name", request.getParameter("user_name"));
        request.getSession().setAttribute("login_time", loginTime);

        response.sendRedirect("/logout.jsp");
    }
}
```

LoginServlet.java

ログイン画面での処理

イベント送信処理を呼び出す。
成功／失敗に応じて異なるメッセージを表示する。

ログアウト時に利用するため、
セッションにユーザ名とログインに日時を保存する。

(Java)ソースコード解説 2

```
public class CaliperSession {  
  
    private static String ENDPOINT = "http://caliper.imsglobal.org/"; // 送信先URL  
  
    private static String API_KEY = "apikey"; // APIキー  
  
    private static String SENSOR_ID = "sensorId";  
  
    private static String CLIENT_ID = "clientId";  
  
    private static String APP_ID = "https://example.com/app/123456789";  
  
    private static String APP_NAME = "IMS-JS Caliper";  
  
    private static Sensor<String> sensor;  
  
    static {  
        sensor = new Sensor<String>(SENSOR_ID);  
        Options options = new Options();  
        options.setHost(ENDPOINT);  
        options.setApiKey(API_KEY);  
        sensor.registerClient(CLIENT_ID, new Client(CLIENT_ID, options));  
    };  
  
    private static SoftwareApplication app;  
  
    static {  
        app = SoftwareApplication.builder()  
            .id(APP_ID)  
            .name(APP_NAME)  
            .build();  
    };  
}
```

CaliperSession.java

Sensorの準備

送信先Event StoreのURL,
apikeyなどを設定

Sensorのインスタンスを作成

Sensorに送信先を登録

(Java)ソースコード解説 3

```
public static boolean sendSessionLoggedIn(String userName, DateTime loginTime) {
    DateTime now = DateTime.now();

    Person actor = Person.builder()
        .id("https://example.com/persons/" + userName)
        .name(userName)
        .build();

    Session session = Session.builder()
        .id("https://example.com/sessions/12345")
        .actor(actor)
        .dateCreated(loginTime)
        .dateModified(loginTime)
        .startedAtTime(loginTime)
        .build();

    // Java版のAPIでは指定が必須
    MediaLocation location = MediaLocation.builder()
        .id("https://example.com/page/default")
        .name("Default Page")
        .build();

    SessionEvent sessionEvent = SessionEvent.builder()
        .actor(actor)
        .action(Action.LOGGED_IN.getValue())
        .object(app)
        .generated(session)
        .target(location)
        .eventTime(now)
        .build();

    try {
        sensor.send(sensor, sessionEvent);
        return true;
    } catch (Error e) {
        e.printStackTrace();
        return false;
    }
}
```

LoggedInイベントの送信

送信する情報を表すEntity(Person, Sessionなど)を構築する。

Sessionイベントを作成する。
イベントのActor, Objectとして、
直前に構築したEntityを設定する。

イベント送信の実施

CaliperSession.java

(Java)サンプルプログラムの動かし方（事前準備）

1. GitHubからSensor API(Java)のソースを入手する

<https://github.com/IMSGlobal/caliper-java-public>

2. Sensor APIをMavenでローカルインストールする

```
$ mvn clean install
```

(Java)サンプルプログラムの動かし方

1. CaliperSession.javaのAPIキーを修正する
2. サンプルプログラムをMavenでビルドする

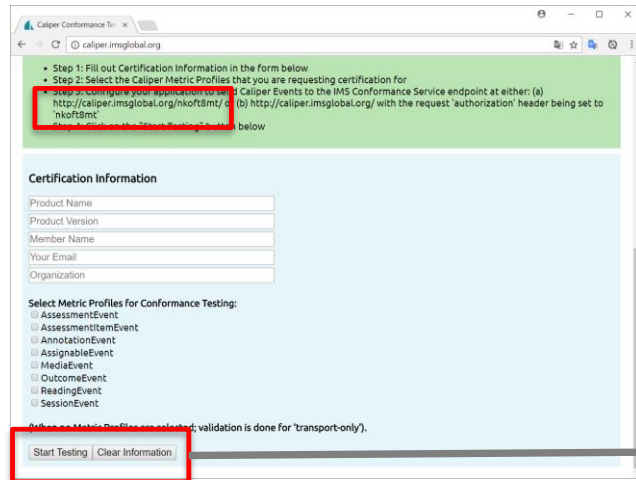
```
$ mvn package
```

3. tomcat-embedを実行する

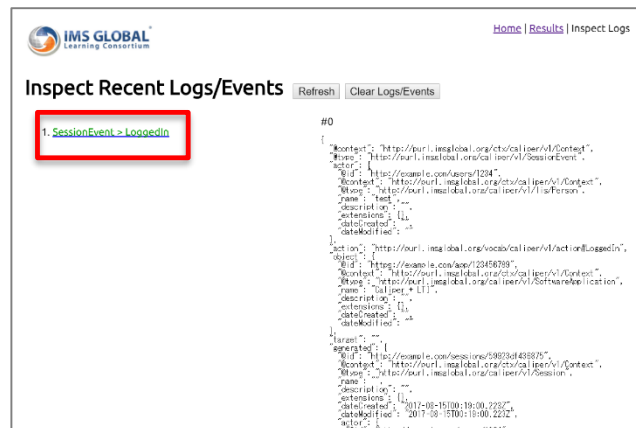
```
$ target¥bin¥tomcat.bat
```

4. Webブラウザで <http://localhost:8888/login.jsp> を開く

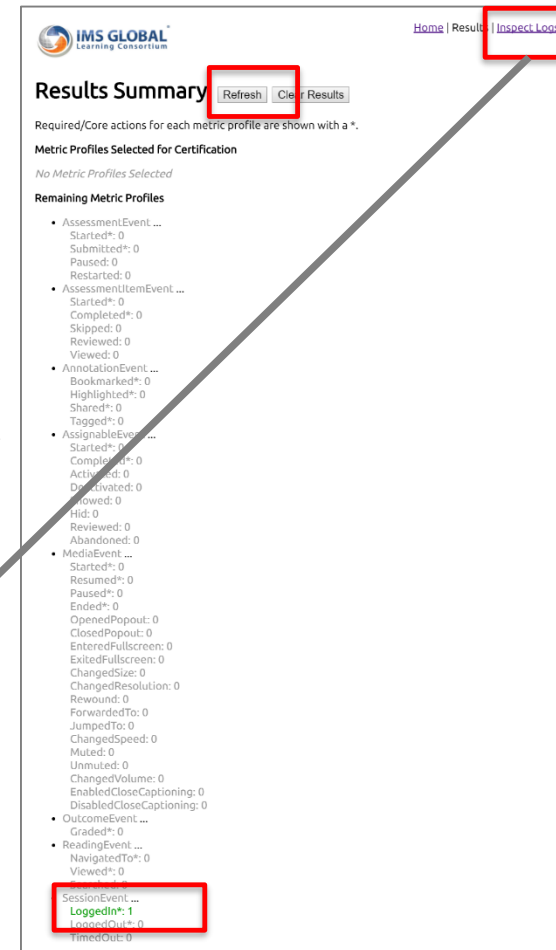
Caliper Conformance Testingの利用方法



1. <http://caliper.imsglobal.org/> にアクセスする。
2. APIキーをサンプルプログラムに設定する。
3. Start Testingを選択するとResultに遷移する。



1. 画面左側に送信されたイベントリストが表示される。
2. イベントを選択すると画面右側にJSONが表示される。



1. サンプルプログラムからのイベント送信を実施する。
2. Refreshを選択して表示内容を更新すると、送信されたEventに対応するActionのカウントが増加する。
3. 画面上部のInspect Logsを選択するとJSON表示に遷移する。