気象庁XML利活用セミナー

～気象庁XMLを入手しよう～

2013/07/11

気象庁気象衛星センター
浜田　浩

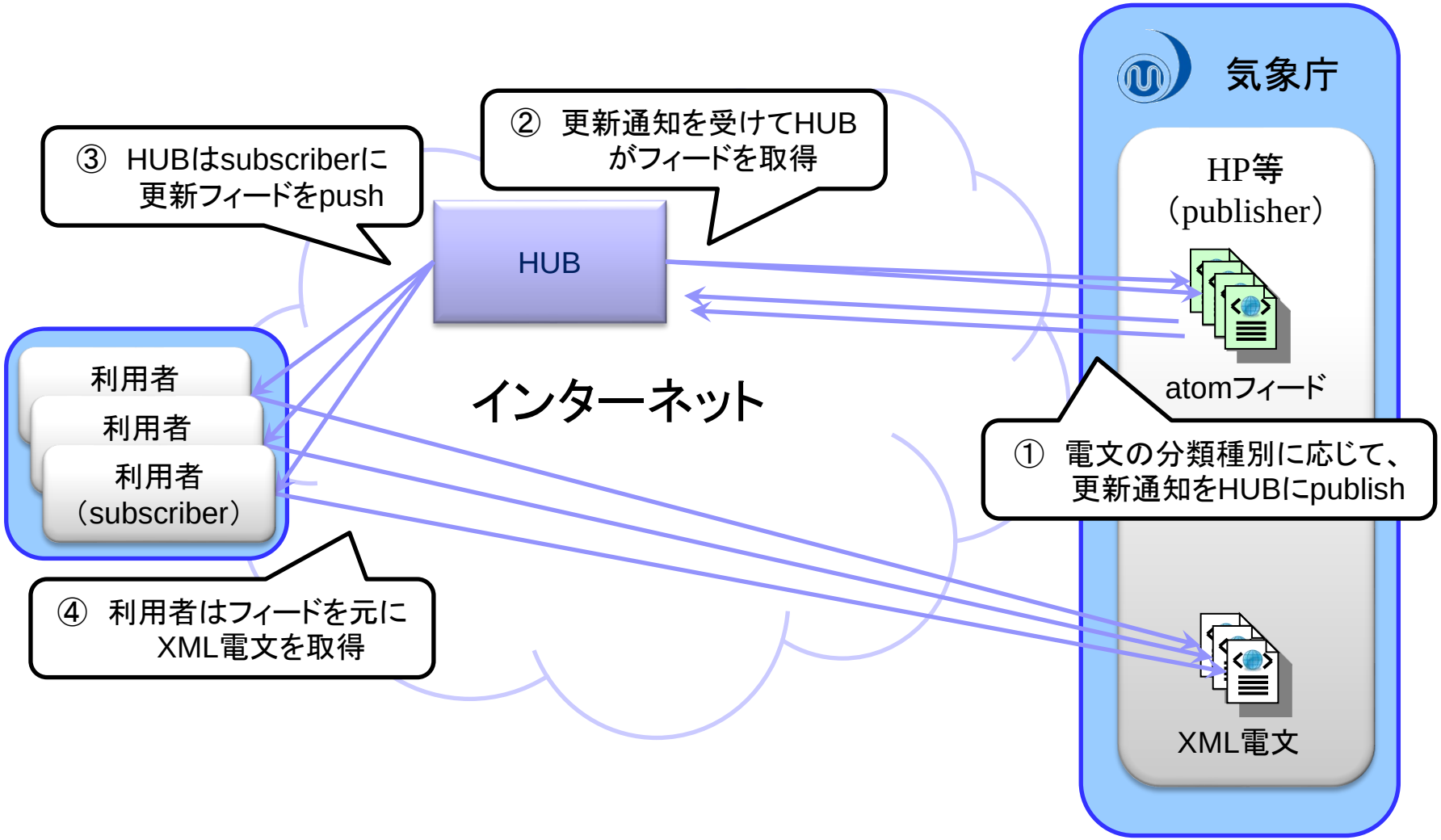
入手するには

1. フィード受信プログラムを書く
   （お手軽なPHPを例に）
2. プログラムをテストする
3. 気象庁に申請する
   - どんな情報が受け取れる？
     （どんなことができる？）

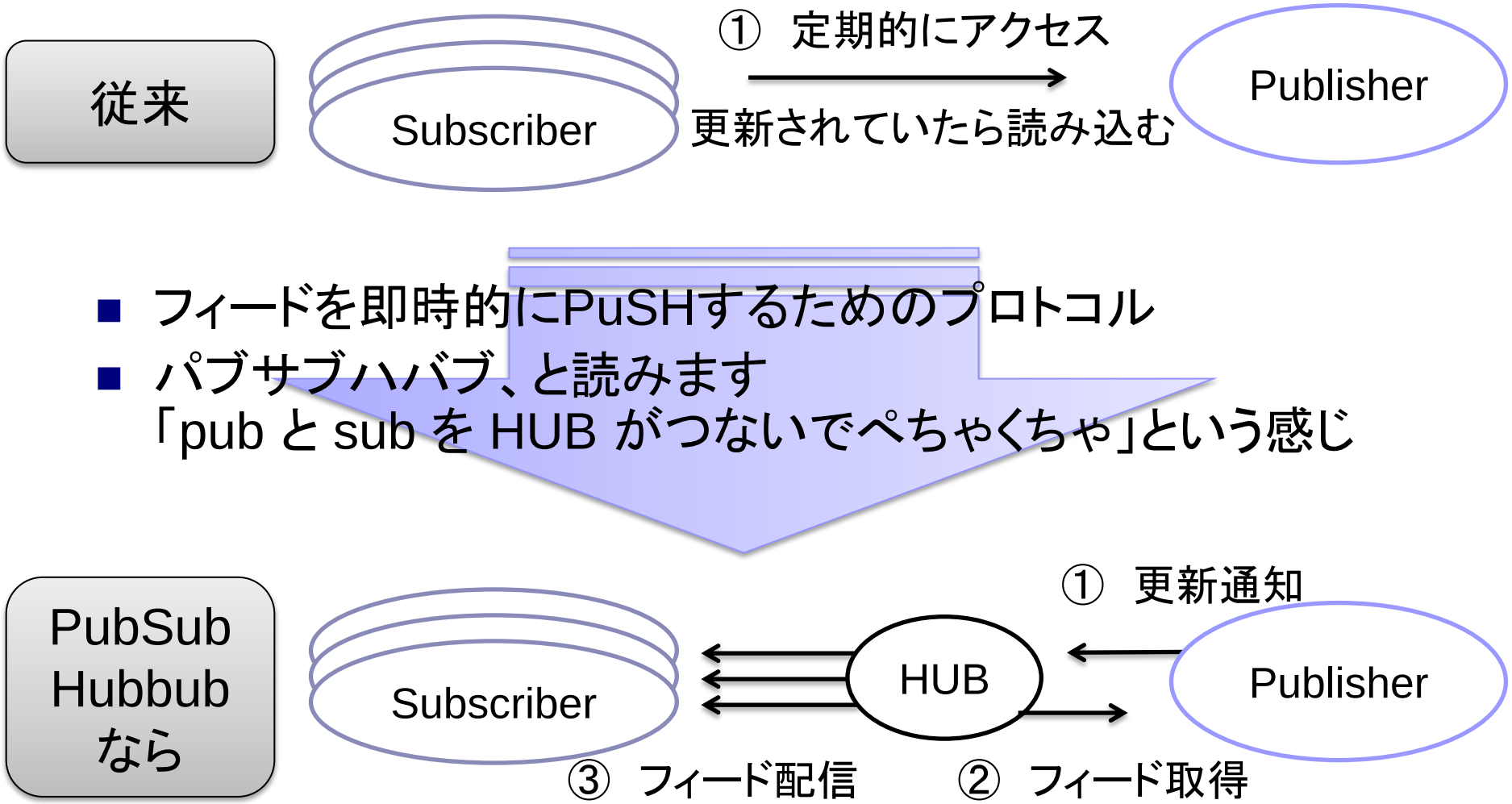
その前に・・・用語説明

- Publisher　フィード送信者
  - 気象庁XMLの更新情報を発行する側
- Subscriber　フィード受信者
  - 気象庁XMLの更新情報を購読する側
- HUB
  - publisherとsubscriberの仲をとりもつ
  - Googleなどに実装あり、利用できる

JMAXMLのpush概念図



PubSubHubbubの仕組み



- フィードを即時的にPuSHするためのプロトコル
- パブサブハバブ、と読みます
  「pub と sub を HUB がつないでぺちゃくちゃ」という感じ

購読環境の構築 （AWSを例に）

ここではアマゾンのEC2を利用する場合の具体例を挙げます。
（なおAWSのアカウント取得やEC2サービス利用の仕方については省略させていただきます・・・・）

EC2のインスタンス起動にあたって。
・AMIは低廉な「Amazon Linux AMI」（t1.micro）利用を想定
・Webサーバとして利用するので80（または443）ポートは開けておく

EC2を起動したあとに行っておくこと。
・httpdとphpをインストールする

```
$ sudo yum -y install httpd
$ sudo yum -y install mod_ssl
$ sudo yum -y install php
$ sudo vi /etc/php.ini
$ sudo service httpd start
```

← https を用いない場合は不要

← タイムゾーンを設定
date.timezone = "Asia/Tokyo"

テストに用いるHUB

気象庁の試行では Google.org による Alert HUB を利用
http://alert-hub.appspot.com/

（ここではテスト用として、GoogleがGAE上に立ち上げている
HUBを利用します）

http://pubsubhubbub.appspot.com/

（HUBのソースは公開されているので、その気になれば、自分でHUBを立ち上げる
ことも可能ですが、その場合は・・・・）

- Google App Engine SDK for Python を入手する
  https://developers.google.com/appengine/downloads?hl=ja
- HUBのリファレンス実装のソース（Python）を入手する
  http://code.google.com/p/pubsubhubbub/source/checkout
- Python は AMI にデフォルトで入っているものでOK・・・・という感じになります

PHPでSubscriber

気象庁 Japan Meteorological Agency

このようなフィードがHUBからpushされてきます

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xml:lang="ja">
<title>JMAXML publishing feed</title>
<subtitle>this feed is published by JMA</subtitle>
<updated>2013-02-18T18:49:01+09:00</updated>
<id>urn:uuid:f57b5866-0c8c-3c92-9aff-10a715cdf48b</id>
<link href="http://www.jma.go.jp/" rel="related"/>
<link href="http://xml.kishou.go.jp/feed/extra.xml" rel="self"/>
<rights>Published by Japan Meteorological Agency</rights>

<entry>
<title>気象警報・注意報</title>
<id>urn:uuid:98364413-e534-3566-ad57-210f1522cb7b</id>
<updated>2013-02-18T09:48:25Z</updated>
<author><name>気象庁予報部</name></author>
<link href="http://xml.kishou.go.jp/data/98364413-e534-3566-ad57-210f1522cb7b.xml" type="application/xml"/>
<content type="text">【東京都気象警報・注意報】伊豆諸島北部、伊豆諸島南部では、強風や高波に注意して下さい。
</content>
</entry>
<entry>
<title>気象警報・注意報</title>
<id>urn:uuid:026a5163-5b51-334a-b965-0688213b0eef</id>
<updated>2013-02-18T09:48:52Z</updated>
<author><name>熊本地方気象台</name></author>
<link href="http://xml.kishou.go.jp/data/026a5163-5b51-334a-b965-0688213b0eef.xml" type="application/xml"/>
<content type="text">【熊本県気象警報・注意報】熊本、天草・芦北地方では、強風に注意して下さい。</content>
</entry>
</feed>
```

PHPでSubscriber

気象庁 Japan Meteorological Agency

シンプルなSubscriberの例

```
<?php
$method = $_SERVER['REQUEST_METHOD'];

// subscribe (or unsubscribe) a feed from the HUB
if ($method == 'GET') {
        $hubmode = $_REQUEST['hub_mode'];
        $hubchallenge = $_REQUEST['hub_challenge'];
        if ($hubmode == 'subscribe' || $hubmode == 'unsubscribe') {
                // response a challenge code to the HUB
                header('HTTP/1.1 200 "OK"', null, 200);
                header('Content-Type: text/plain');
                echo $hubchallenge;
        } else {
                header('HTTP/1.1 404 "Not Found"', null, 404);
        }
}

// receive a feed from the HUB
if ($method == 'POST') {
        // feed Receive
        $string = file_get_contents("php://input");
        // feed save
        $fp = fopen(date('YmdHis') . "_atom" . ".xml", "w");
        fwrite($fp, $string);
        fclose($fp);
}
?>
```

※GET時は登録確認

HUBから、登録確認のため初回および5日間隔で、チャレンジコード付きリクエストが飛んでくるので、そのまんまbodyに入れて返す。

※POST時はフィード配信

受信したフィードに対する処理を記述する。

（ここでは単にフィードをファイルに保存しています）

PHPでSubscriber

気象庁XMLを取得する例

```
// receive a feed from the HUB
if ($method == 'POST') {
        // feed Receive
        $string = file_get_contents("php://input");
        // feed Parse & XML GET
        if (FALSE === ($feed = simplexml_load_string($string))) {
                exit("feed Parse ERROR");
        }
        foreach ($feed->entry as $entry) {
                $url = $entry->link['href'];
                $ch = curl_init();
                curl_setopt($ch, CURLOPT_URL, $url);
                curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
                curl_setopt($ch, CURLOPT_TIMEOUT, 60);
                curl_setopt($ch, CURLOPT_HEADER, 0);
                $fp = fopen(basename($url), "w");
                curl_setopt($ch, CURLOPT_FILE, $fp);
                curl_exec($ch);
                curl_close($ch);
                fclose($fp);
        }
}
```

前頁から、POST処理の部分だけを、たとえば左記のように変更。

※ エラー処理やリトライ等は適宜記述を。

PHPでSubscriber

```
if ($hubmode == 'subscribe' || $hubmode == 'unsubscribe') {
        if ($_REQUEST['hub_verify_token'] != "somekey") {
                header('HTTP/1.1 404 "Unknown Request"', null, 404);
                exit("Unknown Request");
        }
        // response a challenge code to the HUB
        header('HTTP/1.1 200 "OK"', null, 200);
        header('Content-Type: text/plain');
        echo $hubchallenge;
} else {
        header('HTTP/1.1 404 "Not Found"', null, 404);
}
```

GETリクエスト受信時に hub.verify_token が申告したものと同一か確認

※ ここでは仮に verify_token を “somekey” とする

verify_token で 照合する場合の例

```
// feed Receive
$string = file_get_contents("php://input");
if (isset($_SERVER['HTTP_X_HUB_SIGNATURE'])) {
        $sign = explode('=', $_SERVER['HTTP_X_HUB_SIGNATURE']);
        $sha1 = hash_hmac("sha1", $string, "somekey");
        if ($sign[1] != $sha1) {
                header('HTTP/1.1 404 "Invalid X-Hub-Signature"', null, 404);
                exit("Invalid X-Hub-Signature");
        }
}
// feed Parse & XML GET
if (FALSE === ($feed = simplexml_load_string($string))) {
        exit("feed Parse ERROR");
}
```

POSTリクエスト受信時に ‘HTTP_X_HUB_SIGNATURE’ と verify_token をキーにして求めたハッシュ値が同一か確認

※ GAEの実装上ではsecretキーの代用として、一応、機能します

Subscriberをテストする

【前準備】
・ブログなど、任意のタイミングで容易にフィード更新できるものを用意しておく

【購読】
・HUBのsubscriber登録/解除用URLにブラウザでアクセスする
　http://pubsubhubbub.appspot.com/subscribe
・「Callback」欄に、subscriberのURLを入力する
　（前頁のサンプルを元に作成したsubscriberのURLを指定します）
・「Topic」欄に、テスト用 atom feed のURL（前準備で用意したもの）を入力する
・「Mode」欄で、”Subscribe”を選択し、[Do it]ボタンで、登録完了

【発行】・・・・PubSubHubbubに対応したブログ等の場合、ここのステップは不要
・HUBのpublish用URLにブラウザでアクセスする
　http://pubsubhubbub.appspot.com/publish
・「Topic」欄に、テスト用 atom feed のURL（前準備で用意したもの）を入力する
・[Publish]ボタンを押すと、HUBへ更新通知される

【テスト】
・フィードを更新し、上記の【発行】を行う、これを繰り返し、subscriberをテストする

完成したら登録申請を

- 動作に問題が無ければ、気象庁宛に登録申請を願います
  - 本試行においては HUB へのユーザ登録は気象庁側で行います

- 登録に必要な情報としては、
  - 氏名、連絡先、subscriberのURLなど
  - verify_tokenの登録を希望する場合はtokenを記載
  - 登録を希望するatomフィードの分類種別
    （試行を開始した時点では以下の4種）

    定時：気象に関する情報のうち、天気概況など定時発表されるもの

    随時：気象に関する情報のうち、警報・注意報など随時発表されるもの

    地震火山：地震、火山に関する情報

    その他：上記3種類のいずれにも属さないもの

なりすましを回避したい場合

（詳しくは、こちらを参照ください）

http://xml.kishou.go.jp/open_trial/index.html

試行公開されている情報の一覧

【定時】
天気概況
府県天気予報／地域時系列予報
週間天気予報
全般季節予報
地方季節予報
など・・・・

いろいろ有ります

【地震火山】
震度速報
震源・震度に関する情報
津波警報・注意報・予報
津波情報
東海地震予知情報
東海地震注意情報
噴火警報・予報
噴火に関する火山観測報
など・・・・

【その他】
特殊気象報
生物季節観測報告気象報
紫外線観測データ
府県海氷予報
など・・・・

【随時】
全般台風情報
台風解析・予報情報
気象警報・注意報
指定河川洪水予報
土砂災害警戒情報
記録的短時間大雨情報
竜巻注意情報
全般気象情報
地方気象情報
府県気象情報
スモッグ気象情報
異常天候早期警戒情報
地方高温注意情報
府県高温注意情報
など・・・・

気をつけておきたいポイント

- 登録確認は5日間隔で飛んできます（21時または20時）
  - チャレンジコードを返しそびれると登録が解除されますので、ご注意
- 必要な公開XML電文は24時間以内に取得してください
  - サーバ側の保存期間は最短24時間です
- 情報が真正であることを確認する場合は・・・
  - フィードや公開XML電文のドメインは「http://xml.kishou.go.jp/」です
    （ただし、公開XML電文のドメインについては、今後、変更となる可能性あり）
  - 他には、verify_tokenを利用する、Callback URLを他人に知られないようにする、等
- 先月（2月）の実績では、10186件のフィード、22274件のエントリを発行
  - ただし、同じフィードが重複して push されることがあります
- 迅速・確実な電文配信を保証するものではありません（試行）
  - 多少のタイムラグがあります

購読によって、出来ること、いろいろ

- 直接データ利用、例としては・・・
  - 天気予報や気象警報・注意報を利用
  - 生物季節観測や紫外線観測データを利用
- 二次利用、例としては・・・
  - 情報を蓄積、検索できるサイトを立ち上げる
  - websocket等を使って、クライアントへ二次配信
- 第Ⅱ部の「Project　LA（Leads to Action）」でもデータ利用される予定です
  - ・・・・詳しくは、このあとの各種プレゼンをご覧ください

ありがとうございました