AITC
先端ＩＴ活用推進コンソーシアム

PubSubHubbub試用レポート

2012/08/30

クラウド・テクノロジー研究部会　メンバー
気象庁　　浜田　浩

試用の目的

AITC
先端ＩＴ活用推進コンソーシアム

- Web上の更新情報をタイムラグ無しに
  - たいていの場合、情報は不定期に発生する
  - 発生した情報を素早く
    - 周知したい（送り手）
    - キャッチしたい（受け手）
    - でも多数から頻繁なポーリングは困る（送り手）

- そこで・・・・PubSubHubbub

  Publish（発行）Subscribe（購読）Hub・・・bub

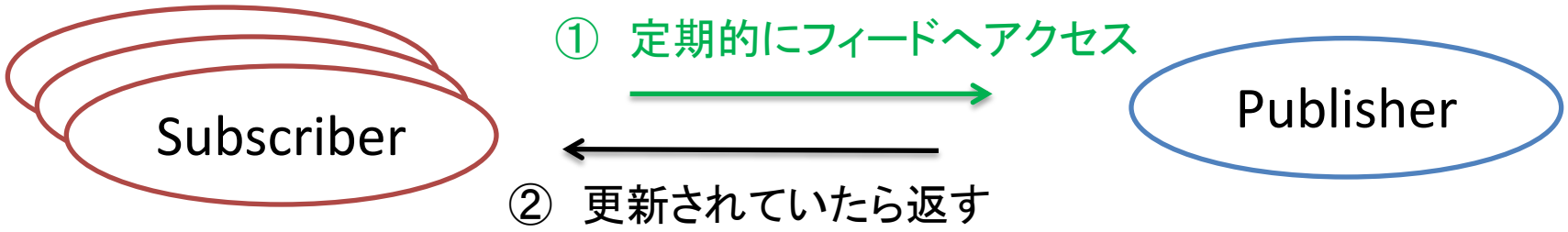
PubSubHubbub？

- パブサブハバブ、と読みます
- ４年ほど前、Googleの例の20%ルールから生まれたプロトコル
- AtomやRSSフィードなどの更新情報を即時的にPUSHできる仕組み
  - 一般的にはblogの更新通知に用いられている
- 「Project LA（Leads to Action）」のなかで使えそう？
  - あらたに発生したトピックを知らせる手段として、等・・・

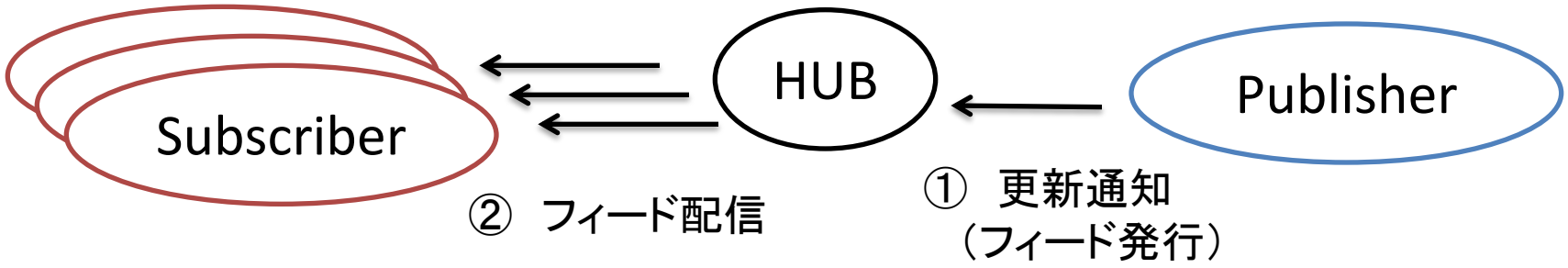
なにが違うの？

AITC
先端ＩＴ活用推進コンソーシアム

従来では

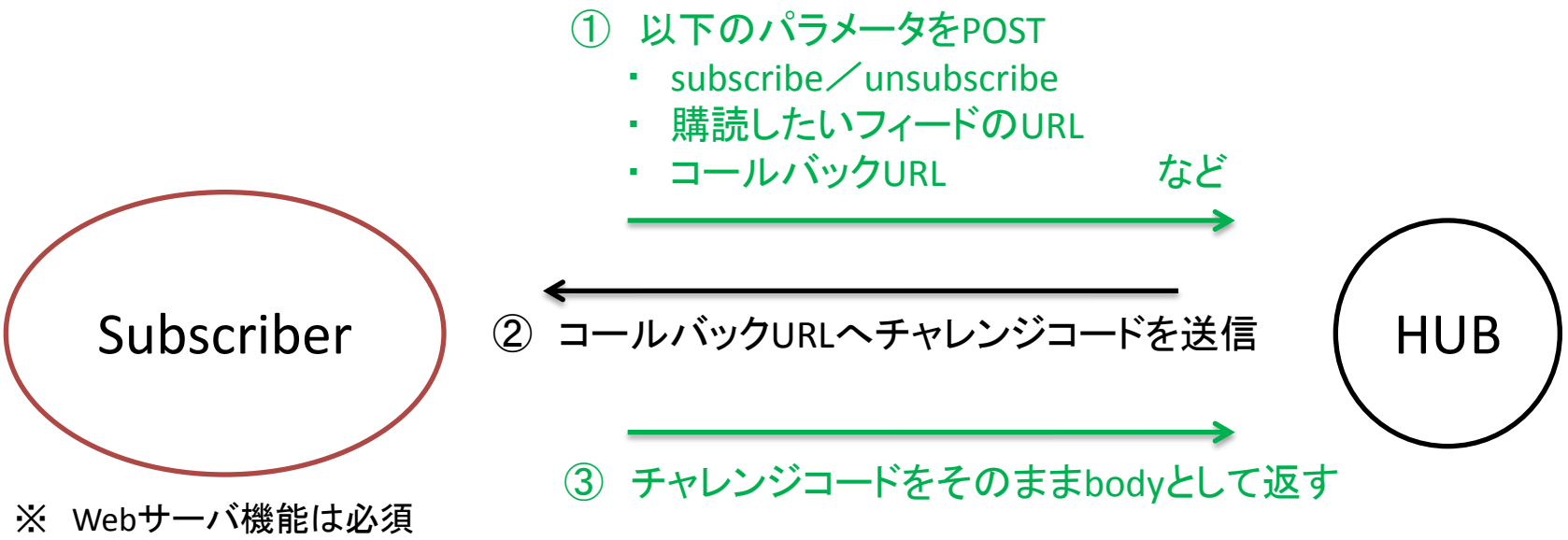


PubSubHubbubを使うと



購読者側の事前準備

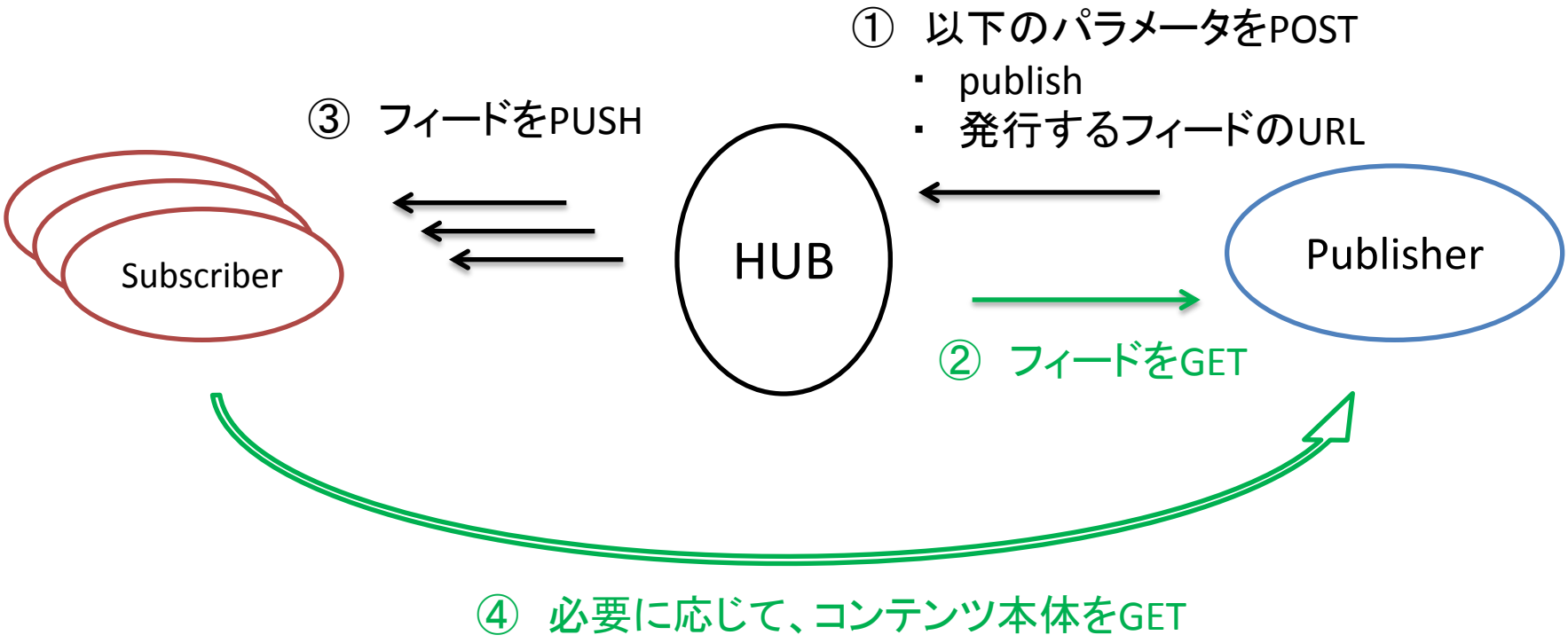
まずはHUBに購読の登録を



※ Webサーバ機能は必須

どんな動きをするの？

Publisherがフィードを発行すると



HUBの実装例など

- GAE上で動いているHUBのリファレンス実装

  http://pubsubhubbub.appspot.com/

- 実用例としては Google の Alert Hub など

  http://alert-hub.appspot.com/

  – Google.orgのプロジェクトのひとつ「Crisis Response」にて、緊急情報を全世界に配信するサービス基盤として利用

  http://www.google.org/intl/ja/crisisresponse/resources.html

- その他のHUB

  – Superfeedr ・・・ http://superfeedr.com/

    - アカウント登録が必要で、一部のサービス（機能）は有料
    - Publish時のBASIC認証や、Subscriber数の表示機能等あり

試用してみる①

（PublisherとSubscriberはAWSのEC2上に用意）

Subscriber登録

HUBに対し、以下のパラメータをPOST

hub.mode=subscribe

hub.topic=http://ec2-********.amazonaws.com/testatom.xml

hub.callback=http://ec2-********.amazonaws.com/subscribe.php

すると、subscribe.phpに対しHUBからチャレンジコードが送られてくるので、
そのままbodyとして返送すれば登録完了（以下のオレンジの部分）

74.125.184.41 - - [14/Aug/2012:06:35:44 +0000] "GET /subscribe.php?hub.challenge=jRzl1SmLc3ohTKtGnShHT5hBwr--_gvneWZIFRgWO-Y22o2OtcEve3btm0R7I6dTBOwHW99dDL2lP546Wq_xaM_NYREhUOp-y8k5r4yn5wD9jUQSBR_M4Mk5WQQPD0P4&hub.topic=http%3A%2F%2Fec2-********.amazonaws.com%2Ftestatom.xml&hub.mode=subscribe&hub.lease_seconds=432000 HTTP/1.1" 200 128 "-" "AppEngine-Google; (+http://code.google.com/appengine; appid: pubsubhubbub)"

試用してみる②

（HUBはGAE上の http://pubsubhubbub.appspot.com/ を利用）

Publisher -> HUB

HUBに対し、以下のパラメータをPOST

hub.mode=publish

hub.url=http://ec2-********.amazonaws.com/testatom.xml

すると HUB が hub.url にフィードを取得しに来る

74.125.184.31 - - [15/Aug/2012:05:32:59 +0000] “GET /testatom.xml HTTP/1.1” 200 5940 “-” “AppEngine-Google; (+http://code.google.com/appengine; appid: pubsubhubbub)“

試用してみる③

HUB は登録されている Subscriber たちへフィードを送信

HUB -> Subscriber

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xml:lang="ja">
（中略）
<id>urn:uuid:ff363839-e032-351c-8049-a7f7909986f9</id>
<link href=“http://ec2-********.amazonaws.com/testatom.xml" rel="self"/>

<entry>
<title>テスト : 00000032</title>
<link href="http://ec2-********.azonaws.com/pub/ef63947f-f670-3ad0-a837-
    002d1b403c2c.xml" type="text/xml"/>
<id>urn:uuid:ef63947f-f670-3ad0-a83
（中略）
<content type=“text”>これはテストです</content>
</entry>
</feed>
```

HUB から Subscribe された Feed の例
（HUB が過去に処理した entry は含まれない）

HUBを立ち上げてみる①

（ここでは例として、AWSのEC2上で動かしてみる）

前準備として　Google App Engine SDK for Python　を入手し展開しておく
https://developers.google.com/appengine/downloads?hl=ja

HUBのリファレンス実装のソース（Python）を svn でチェックアウトする
http://code.google.com/p/pubsubhubbub/source/checkout

チェックアウトしたソースのディレクトリに移動する

```
$  cd  pubsubhubbub-read-only/
```

HUBを立ち上げる（-aオプションで自ホスト名を指定）

```
$  ../google_appengine/dev_appserver.py  -a  ec2-********aws.com  hub
```

Publisherを立ち上げる（-pオプションでポートを8081に指定）

```
$  ../google_appengine/dev_appserver.py  -a  ec2-********aws.com  -p  8081  publisher
```

Subscriberを立ち上げる（-pオプションでポートを8082に指定）

```
$  ../google_appengine/dev_appserver.py  -a  ec2-********aws.com  -p  8082  subscriber
```

HUBを立ち上げてみる②

ローカルPCのブラウザにて HUB, Publisher, Subscriber を表示

http://ec2-********aws.com:8080　（HUB）
http://ec2-********aws.com:8081　（Publisher）
http://ec2-********aws.com:8082　（Subscriber）

HUB(8080)のSubscribeページでテスト用のフィードを「購読」するよう登録を行う
http://ec2-********aws.com:8080/subscribe
Callback: http://ec2-********aws.com:8082/subscriber
Topic: http://ec2-********aws.com:8081/feed

Publisher(8081)のページでテスト用のフィードを拵える
http://ec2-********aws.com:8081

HUB(8080)のPublishページでテスト用のフィードを「発行」
http://ec2-********aws.com:8080/publish
Topic: http://ec2-********aws.com:8081/feed

すると・・・・

流れのみ端的に書くと・・・・

HUBを立ち上げてみる③

Subscriber(8082)のページで、更新通知がPUSHされていることを確認できます
http://ec2-********aws.com:8082

試用して気になった点

- 仕様の最新版は？
  - Googleのサイト上に仕様のドラフト版あり

http://pubsubhubbub.googlecode.com/svn/trunk/pubsubhubbub-core-0.3.html

    - 2009.7.8（ver0.1），2009.9.1（ver0.2），2010.2.8（ver0.3）（最新?）
  - 別のサイト上にver0.4（2012.2.5）あり（Superfeedr用?）

http://superfeedr-misc.s3.amazonaws.com/pubsubhubbub-core-0.4.html

- Publisher側でSubscriberを把握できない
  - 購読者がどのくらい存在するか知りたいときは？
  - ドラフト版 0.3 までは User-Agent に購読者数の情報が付加されるような記述あるも、ドラフト版 0.4 では無くなっている
  - Superfeedr は、このあたり付加価値としての機能を持つ
    - http://superfeedr.com/

最後に、感想など

AITC
先端ＩＴ活用推進コンソーシアム

- Publish時以外にもHUBは15分毎にポーリングに来る
  - Publishしなくても、最大15分のタイムラグでフィードを拾ってくれそう、だが、HUBの挙動に依存するのも宜しくない？
- あまり利用されている気配が無い？
  - Googleの各種サービスを中心に、裏方で利用
  - Google Buzz, FeedBurner, Blogger, Google Alerts, ・・・
  - 一般的な需要としては、Twitterで事足りてしまった？
  - 地味な存在だけど、今後さらに広まってゆくか・・・
- 「Project LA（Leads to Action）」では？
  - なんらかのかたちで使えそうである （今後の検討）