



XML Consortium

～ 第9回 XMLコンソーシアムWeek ～

Webサービス実証部会

気象庁防災情報XMLを使った実証実験

大規模災害発生時における 安否確認をどう行うか

2010年3月18日

アドソル日進株式会社

荒本道隆

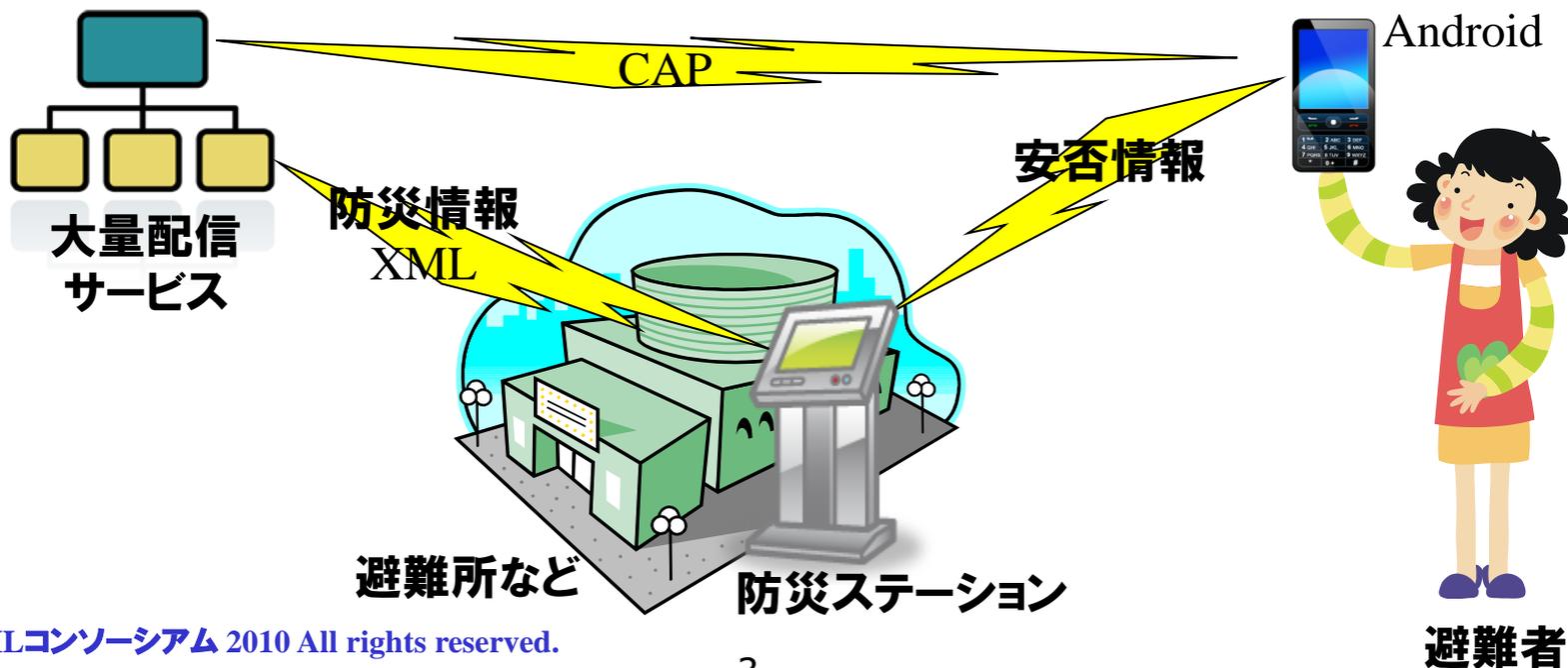
斉藤智嗣

大規模災害発生時に、ネットワークが断絶した場合においても、なんらかの手段を用いて必要としている情報の配信をおこないたい。

当初はAndroidを使ったP2Pネットワークの構築を考えていたが、課題が数多くあり、断念。

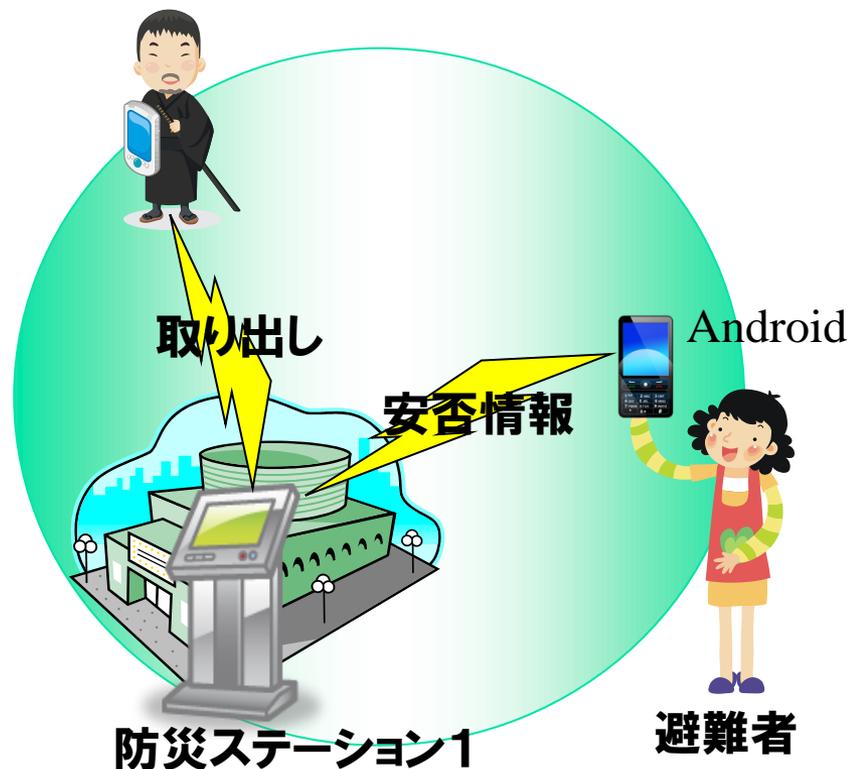
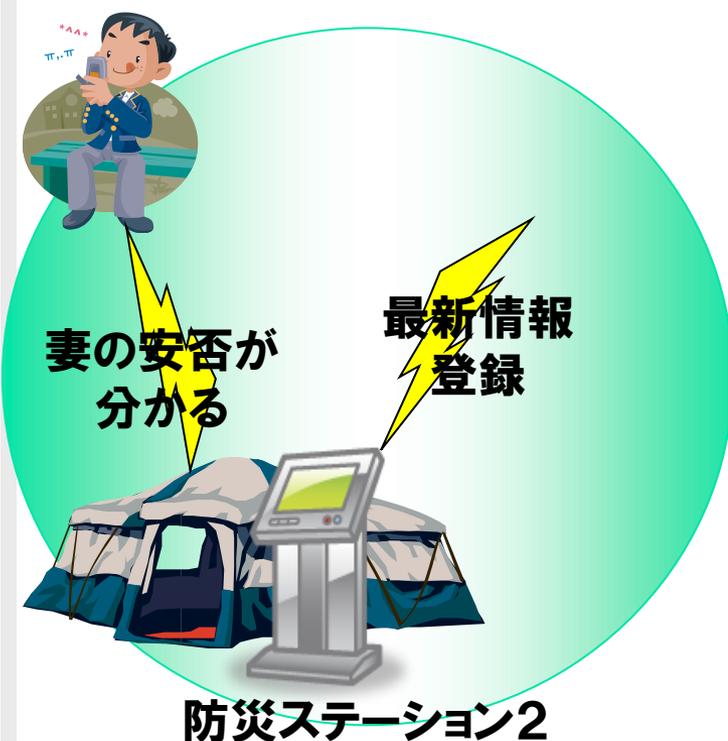
そこで、防災ステーション同士のP2Pを目指して実証実験を実施。Internetが使えなくなっても、Androidを使用してデータを持った人が移動する、という方式を検討した。

- **防災情報の登録と参照**
 - 大量配信サービスから配信される
 - 防災情報XML、もしくは、CAP(Common Alerting Protocol)
- **安否情報の登録と参照**
 - 各Android端末から、避難者によって登録されたもの
 - 電話番号・緯度経度・コメント



人によるデータ配信

- 避難者(妻)が、安否情報を防災ステーション1に登録
- 別の人(夫)が、防災ステーション1からデータを持ち出す
- その人が移動して、防災ステーション2にデータを入れる
- 避難者(夫)が、妻の安否情報を確認



- **大量配信サービス⇒防災ステーション**
 - **防災情報XML登録**
- **大量配信サービス⇒Android**
 - **CAP(Common Alerting Protocol)登録**
- **防災ステーション⇔Android**
 - **安否情報登録**
 - **安否情報検索**
 - **興味のある人→電話帳に登録されている電話番号**
 - **エクスポート、インポート**
 - **データを持ち運ぶため**

- 登録: URLに対してXMLを投げる
- 検索: URLにアクセスするとXMLが返ってくる

```
POST /BousaiStation/anpi/new HTTP/1.1
User-Agent: Java/1.6.0_14
Host: localhost
Accept: text/html, image/gif, image/jpeg, *, q=.2, */*; q=.2
Connection: keep-alive
Content-type: application/x-www-form-urlencoded
Content-Length: 186
```

```
<Anpi>
  <Tel>0900000000</Tel>
  <Name>名前</Name>
  <Lat>35.65861</Lat>
  <Lng>139.745447</Lng>
  <Comment>池袋駅を出ました</Comment>
  <LastUpdate></LastUpdate>
  <APLocation></APLocation>
</Anpi>
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/plain
Content-Length: 2
Date: Sat, 13 Mar 2010 03:04:01 GMT
```

```
GET /BousaiStation/anpi/search/0900000000 HTTP/1.1
User-Agent: Java/1.6.0_14
Host: localhost
Accept: text/html, image/gif, image/jpeg, *, q=.2, */*; q=.2
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml;charset=UTF-8
Content-Length: 1312
Date: Sat, 13 Mar 2010 03:10:49 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Anpis>
  <Anpi Id="station1-1-9cdaaa76-6ca4-4b8a-a034-a3d310280a17">
    <Tel>0900000000</Tel>
    <Name>名前</Name>
    <Lat>35.65861</Lat>
    <Lng>139.745447</Lng>
    <Comment>池袋駅を出ました</Comment>
    <LastUpdate>2010-03-13T12:04:01+09:00</LastUpdate>
    <APLocation>第1防災ステーション</APLocation>
  </Anpi>
  <Anpi Id="station1-1-e10d0ad0-ce98-4297-b830-d7ec052460ef">
    <Tel>0900000000</Tel>
    :
```

■ エクスポート

- Android側は、Streamデータとして扱う
 - 1件ずつ個別に扱くと、性能が心配
 - 中身については関知しない
- データ形式を変えても、Android側は修正不要
 - 今回はデバッグし易いようにXMLのまま
 - 転送時間短縮→圧縮
 - セキュリティ→署名、暗号化

■ インポート

- 防災ステーションで、受け取ったデータをマージ
 - Idを参照してマージ

各機能の詳細

■ 登録

- 安否情報は、ユニークになるようにIdを付与
- 受け取ったXMLを1つの大きなXMLの一番上に足す
- スキーマを使用せず、バインドもしない
 - 未知の要素があっても、気にしない
 - 検索に使用する要素だけは必須
 - 異なるXMLへの対応も、興味のある要素を見るコードだけ

■ 検索

- 検索条件もURLの一部として表現

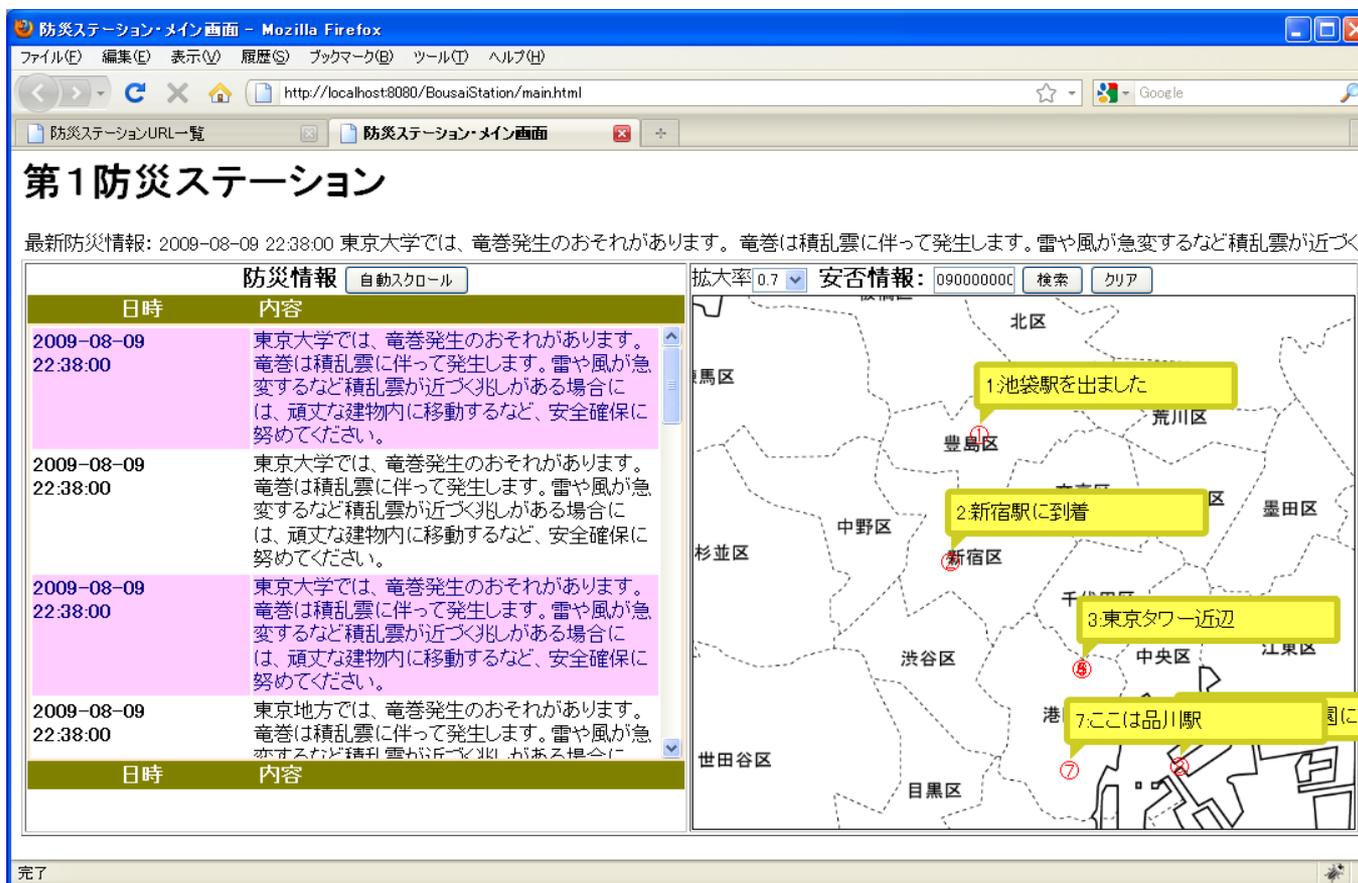
/api/search/0901234567

■ 情報を表示するためのWeb画面

- 防災ステーション自体を端末としても活用

防災ステーション画面

- 最新の防災情報XMLを表示
- 電話番号で安否情報を検索し、地図上に表示



第1 防災ステーション

最新防災情報: 2009-08-09 22:38:00 東京大学では、竜巻発生のおそれがあります。竜巻は積乱雲に伴って発生します。雷や風が急変するなど積乱雲が近づくと...

日時	内容
2009-08-09 22:38:00	東京大学では、竜巻発生のおそれがあります。竜巻は積乱雲に伴って発生します。雷や風が急変するなど積乱雲が近づくと...
2009-08-09 22:38:00	東京大学では、竜巻発生のおそれがあります。竜巻は積乱雲に伴って発生します。雷や風が急変するなど積乱雲が近づくと...
2009-08-09 22:38:00	東京大学では、竜巻発生のおそれがあります。竜巻は積乱雲に伴って発生します。雷や風が急変するなど積乱雲が近づくと...
2009-08-09 22:38:00	東京地方では、竜巻発生のおそれがあります。竜巻は積乱雲に伴って発生します。雷や風が急変するなど積乱雲が近づくと...

安否情報: 0900000000 検索 クリア

1. 池袋駅を出ました

2. 新宿駅に到着

3. 東京タワー近辺

7. ここは品川駅

■ 実行環境

- Windows+Java1.6+Tomcat5.5
- 地図画像
 - <http://www.freemap.jp/>
- JavaScriptライブラリ
 - jquery-1.3.2
- 吹き出し描画のためのJavaScriptライブラリ
 - jquery-qtip-1.0.0-beta3

■ **安否確認**

- 安否情報のインポート(防災ステーションを最新に)
- 安否情報のエクスポート(Android内を最新に)
- 安否情報の検索(表示用データを取得)

■ **安否情報登録**

- 自分の安否情報を登録
 - 自動:電話番号、緯度経度
 - 入力:表示名、コメント

■ **防災情報の送受信**

- 防災情報のインポート(防災ステーションを最新に)
- 防災情報のエクスポート(Android内を最新に)

■ CAP情報の閲覧

- 大量配信サービスから受信したCAP情報を表示
- AndroidでAPサーバ(i-jetty)を起動し、そこで受信
 - **真のPush配信**
- 配信してもらうために大量配信サービスに登録
 - 端末ID
 - 緯度・経度
 - URL

- **安否確認(アプリ起動時)**
 - 電話帳に登録してある人の安否情報を自動取得して表示
- **安否情報登録**
 - 自分の安否情報を登録
- **防災情報の送受信**
 - 防災ステーションの情報を更新
 - 最新情報を再取得
- **CAP情報の閲覧**
 - 配信されたCAP情報を表示

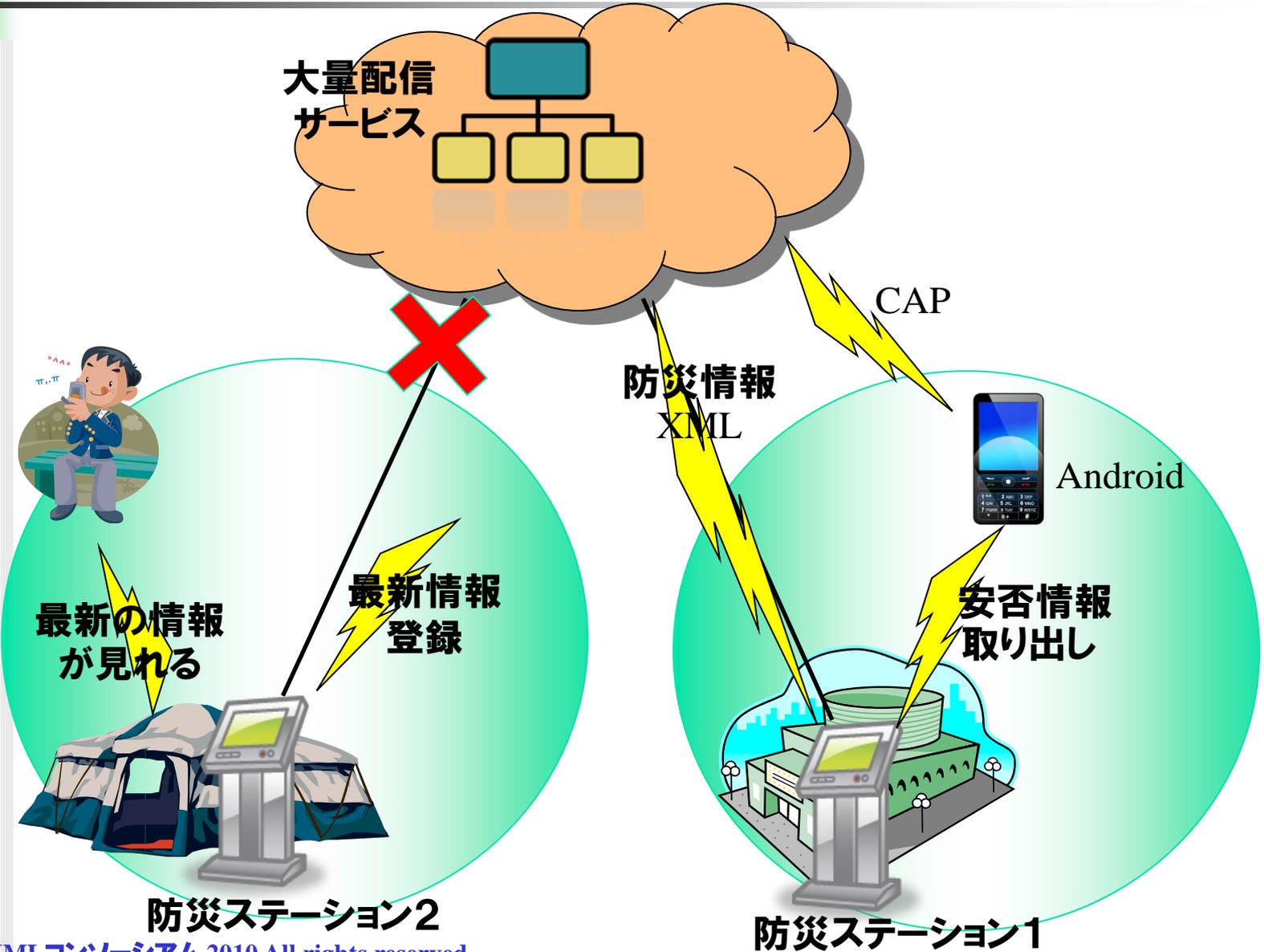


■ 実行環境

- Android 1.5
- CAP受信のためのAPサーバ
 - i-jetty 2.1

デモ

- CAPの配信
 - 大量配信サービスから届いたCAPを表示
- 安否情報の表示
 - 電話番号に登録されている人の安否情報を確認
- 安否情報の登録
 - Androidから安否情報を登録
 - 防災ステーションで確認
- データの持ち運び
 - 防災ステーション1→2へデータをコピーする
 - 防災ステーション2で、防災情報、安否情報の確認



- 防災ステーションは、同じIPアドレスを使用
 - 無線LANごとに同じローカルセグメントを使用
 - 移動時にAndroidアプリの設定を変更しない
 - DNSを工夫すれば、違うIPでも対応可能

- **今回はXMLのままで持ち運んだ**
- **セキュリティ対策を行うとすると**
 - **防災情報XML, CAP**
 - Androidで直接見れたら便利
 - でも改ざんされては困る
 - →電子署名による改ざん防止
 - **安否情報**
 - 個人情報なので、見られたら困る
 - →暗号化による機密性の確保

- Androidではできるだけ重い処理をしない
 - 1つの防災情報XMLが1データだと
 - 件数分だけSQLiteでのINSERTが必要となり、遅そう
 - 全体を1つのStreamデータとして扱う
 - ストレージにベタ書きするだけ
 - 無線LANやストレージへの書き込み性能がそのまま出るはず
 - 複雑な処理は、防災ステーション側でおこなう

- 参考値
 - 防災情報XML100件(約1MByte)の送信と受信
 - 無線LANを使った場合、送受信合わせて約10秒

- **安否情報を運んでもらう動機付け**
 - 「自分の安否情報を登録」「知人の安否が知りたい」ための行動で、他の人の安否情報が配信される

- **実際の災害時には、大量のデータが発生**
 - 効率的な配信方式を検討する必要がある
 - 最新のを優先、分割して一部分だけを持ち運んでもらう
 - セキュリティや性能との兼ね合い

- **Id管理が大切**
 - 確実に重複しないIdを付ける存在が必要
 - マージ処理の時、Idの位置や考え方が違くと面倒

- ⊕ 実証実験システム全体像のご紹介(松山)
- ⊕ 気象庁防災情報XML配信サーバをクラウドを使って実装する(上村)
- ⊕ 気象庁防災情報XMLをGAEで大量配信する一手法(高橋)
- ⊕ 写真登録・管理サービス(荒本)
- ⊕ 大規模災害発生時における安否確認をどう行うか(荒本/斉藤)
- ⊕ **気象庁防災情報XMLを使った実証実験の総括(松山)**