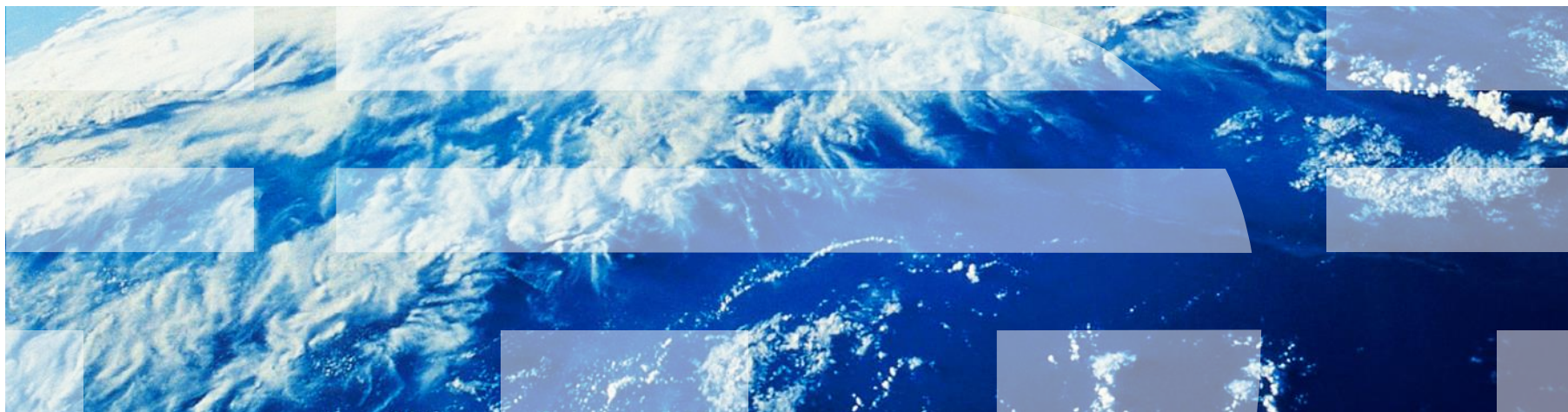


IBM Worklight Foundation V6.2.0 **入門**

SMS 通知



商標

- IBM、IBM ロゴ、ibm.com および Worklight は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。
- Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。
- Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- この資料は、事前に IBM の書面による許可を得ずにその一部または全部を複製することは禁じられています。

IBM® について

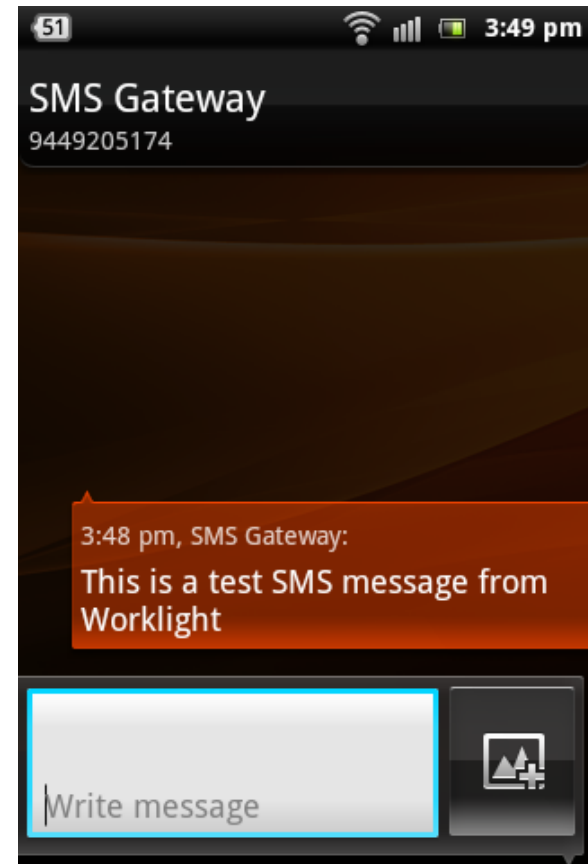
- <http://www.ibm.com/ibm/us/en/> を参照してください。

アジェンダ

- SMS 通知とは
- デバイス・サポート
- 通知アーキテクチャー
- サブスクリプション管理
- SMS 通知 API
- SMS サブスクライブ・サーブレット
- セットアップ

SMS 通知とは

- SMS 通知は、サーバーからプッシュされる SMS メッセージを受信する、モバイル・デバイスの機能です。
- 通知は、アプリケーションが実行中であるかどうかにかかわらず受信されます。



アジェンダ

- SMS 通知とは
- デバイス・サポート
- 通知アーキテクチャー
- サブスクリプション管理
- SMS 通知 API
- SMS サブスクライブ・サーブレット
- セットアップ

デバイス・サポート

- IBM® Worklight Foundation ® は、SMS 機能に対応する iOS、Android、Windows Phone、および BlackBerry の各デバイスで SMS 通知をサポートします。

アジェンダ

- SMS 通知とは
- デバイス・サポート
- 通知アーキテクチャー
- サブスクリプション管理
- SMS 通知 API
- SMS サブスクライブ・サーブレット
- セットアップ

通知アーキテクチャー: 用語

- イベント・ソース: モバイル・アプリケーションがサブスクライブできる通知チャンネル
 - イベント・ソースは Worklight アダプター内で定義されます。
- ユーザー ID: Worklight ユーザーの固有 ID
 - ユーザー ID は、認証または他の固有 ID (パーシスタント Cookie など) を通じて取得します。
- アプリケーション ID: Worklight アプリケーション ID
 - この ID は、モバイル・マーケットで特定の Worklight アプリケーションを識別します。

通知アーキテクチャー: サブスクリプション

- SMS 通知の受信を開始するには、アプリケーションがまず SMS 通知イベント・ソースにサブスクライブする必要があります。
- イベント・ソースは、SMS 通知サービス用のアプリケーションが使用する Worklight アダプターで宣言されます。
- SMS 通知にサブスクライブするために、ユーザーは携帯電話番号を提供し、通知サブスクリプションを承認します。
- サブスクリプション要求は、ユーザー承認の受信時に Worklight Server に送信されます。

通知アーキテクチャー: 通知の送信

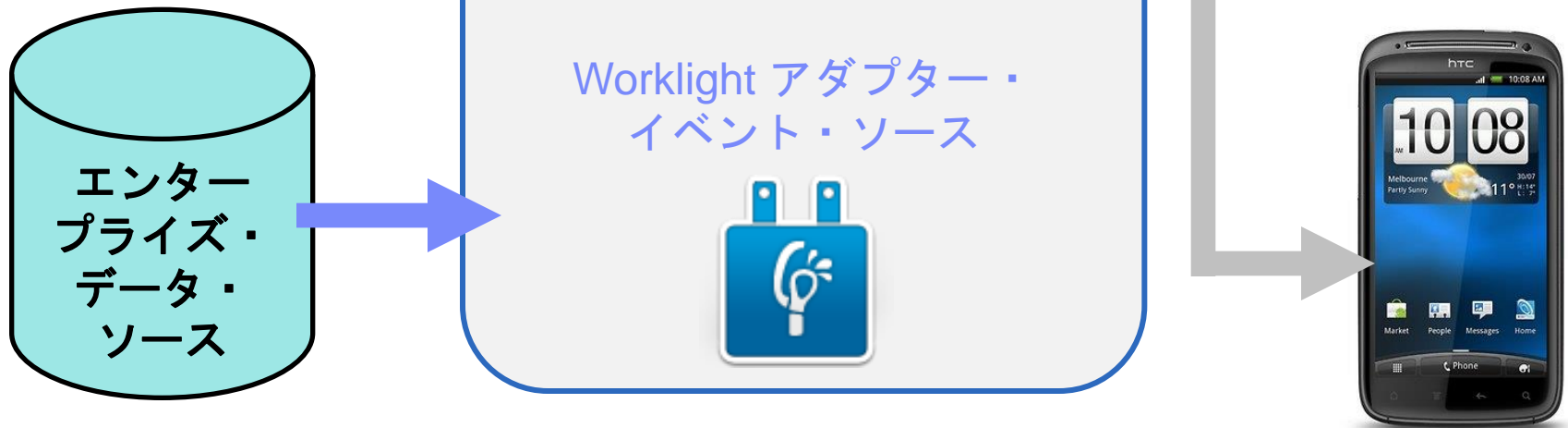
- IBM Worklight は統一されたプッシュ通知 API を提供します。
- アダプター API を使用して、以下を行うことができます。
 - サブスクリプションの管理
 - バックエンド・システムからの通知のプッシュおよびポーリング
 - プッシュ通知のデバイスへの送信
- アプリケーション API を使用すると、プッシュ通知イベント・ソースにサブスクライブすることができ、またこのイベント・ソースからアンサブスクライブすることができます。

通知アーキテクチャー: 通知の送信

- 通知を送信するには、まずバックエンド・システムから通知を取り出す必要があります。
- イベント・ソースは、バックエンド・システムから通知をポーリングでき、またバックエンド・システムが新しい通知を明示的にプッシュするのを待機することもできます。
- アダプターが通知を取得すると、通知は処理され、事前構成された SMS ゲートウェイを通じて送信されます。
- 通知を前処理するために、さらにカスタム・ロジックをアダプターに追加できます。
- SMS ゲートウェイが通知を受信し、それをデバイスに送信します。

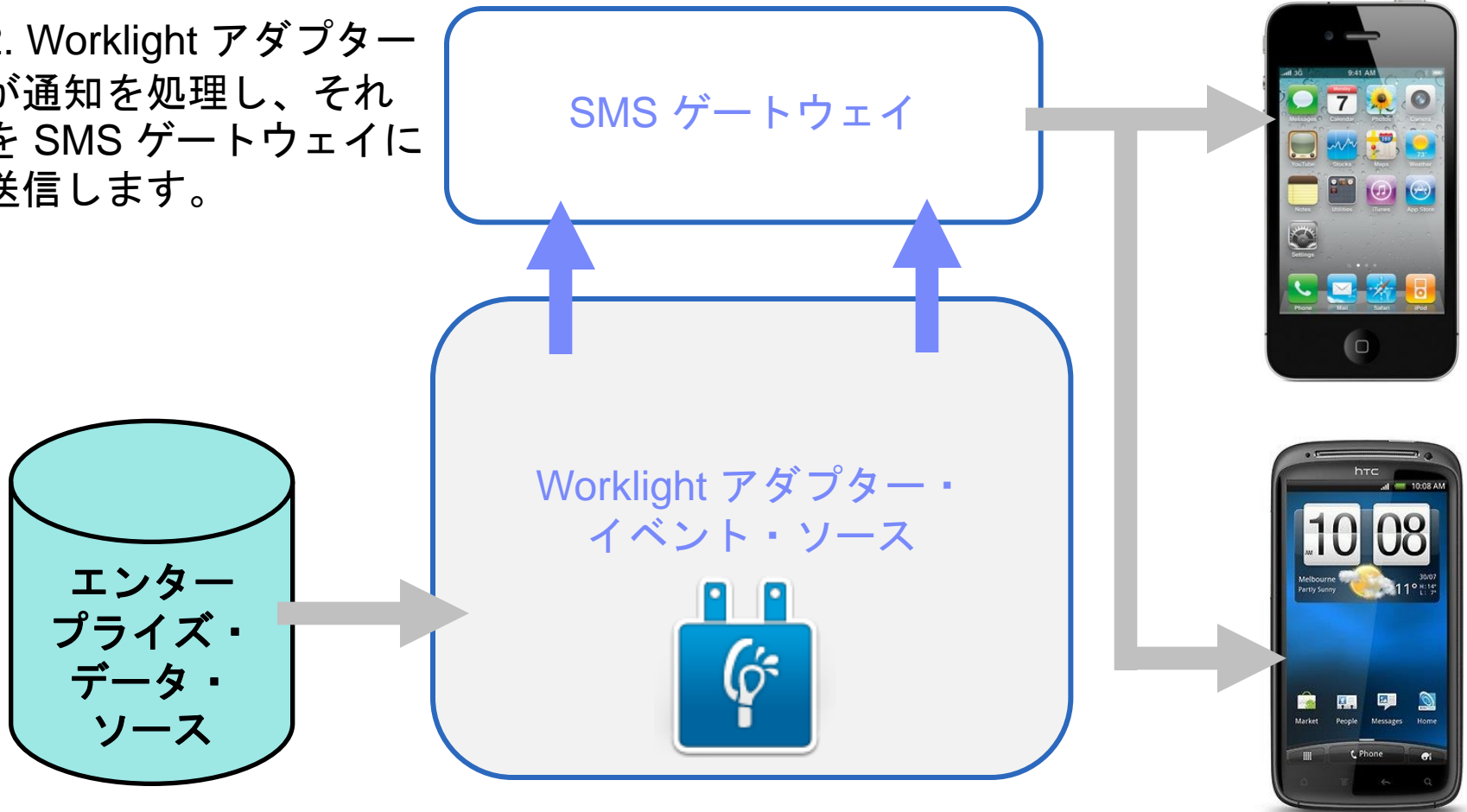
通知アーキテクチャー: 通知の送信

1. バックエンド・システムからのポーリングまたはプッシュのいずれかによって、Worklight アダプター・イベント・ソースが通知を取得します。



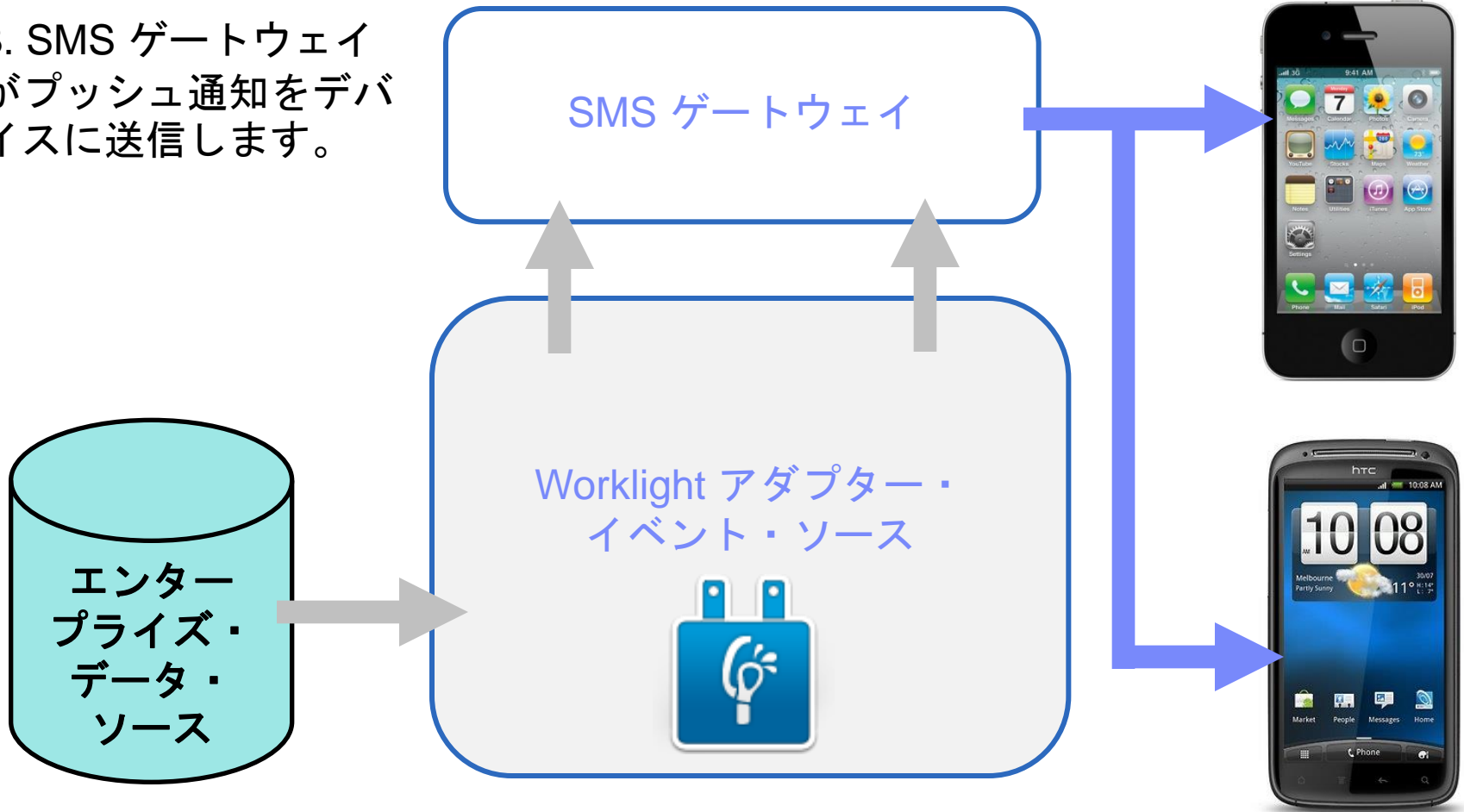
通知アーキテクチャー: 通知の送信

2. Worklight アダプターが通知を処理し、それを SMS ゲートウェイに送信します。



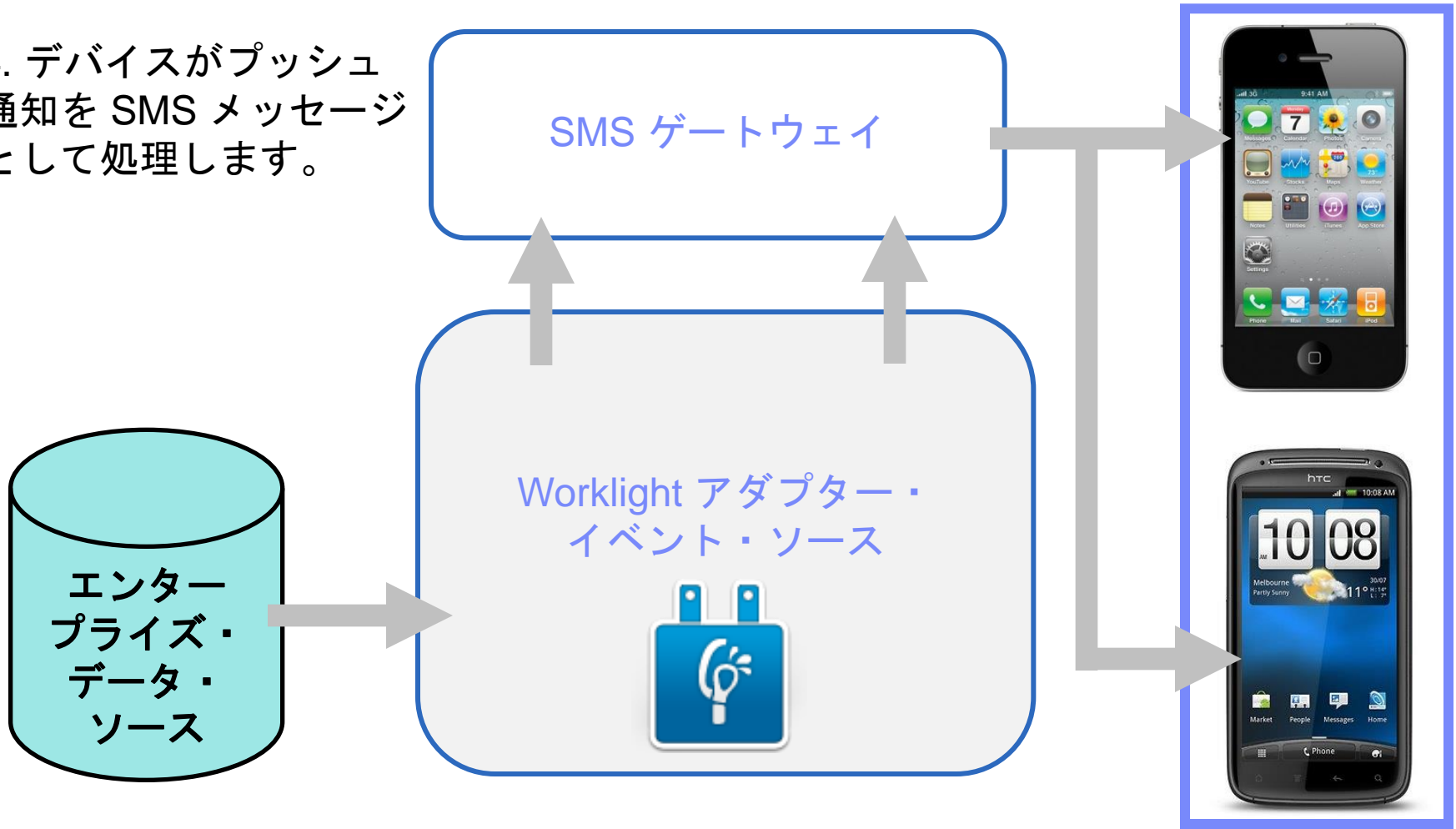
通知アーキテクチャー: 通知の送信

3. SMS ゲートウェイ
がプッシュ通知をデバ
イスに送信します。



通知アーキテクチャー: 通知の送信

4. デバイスがプッシュ通知を SMS メッセージとして処理します。



アジェンダ

- SMS 通知とは
- デバイス・サポート
- 通知アーキテクチャー
- サブスクリプション管理
- SMS 通知 API
- SMS サブスクライブ・サーブレット
- セットアップ

サブスクリプション管理: ユーザー・サブスクリプション

■ ユーザー・サブスクリプション

- ユーザー ID、デバイス ID、およびイベント・ソース ID を含むエンティティ。
- これは、特定のイベント・ソースから通知を受信するユーザーのインテントを表します。

■ 作成

- イベント・ソースのユーザー・サブスクリプションは、ユーザーが任意のデバイスから初めてそのイベント・ソースにサブスクライブしたときに作成されます。

■ 削除

- ユーザーがその全デバイスからのイベント・ソースからアンサブスクライブすると、ユーザー・サブスクリプションは削除されます。

■ 通知

- ユーザー・サブスクリプションが存在している間、Worklight Server は、サブスクライブ済みユーザーの通知を生成することができます。
- これらの通知は、アダプター・コードによって、サブスクライブされた一部またはすべてのデバイスに送信できます。

サブスクリプション管理: デバイス・サブスクリプション

- デバイス・サブスクリプションは、ユーザー・サブスクリプションに属し、特定のユーザーおよびイベント・ソースのスコープ内に存在します。1つのユーザー・サブスクリプションで複数のデバイス・サブスクリプションを持つことができます。
- デバイス上のアプリケーションが **WL.Client.Push.subscribeSMS** を呼び出すと、デバイス・サブスクリプションが作成されます。携帯電話番号が入力パラメーターです。
- デバイス・サブスクリプションは、アプリケーションが **WL.Client.Push.unsubscribeSMS** を呼び出すか、管理者が Worklight Console を通じてユーザーをアンサブスクライブすると、削除されます。

アジェンダ

- SMS 通知とは
- デバイス・サポート
- 通知アーキテクチャー
- サブスクリプション管理
- SMS 通知 API
- SMS サブスクライブ・サーブレット
- セットアップ

通知 API: サーバー・サイド

- 最初にイベント・ソースを作成する
 - アダプター JavaScript コードで通知イベント・ソースをグローバル・レベル (JavaScript 関数外) で宣言します。

通知をバックエンド・システムによってプッシュする

```
WL.Server.createEventSource({  
  name: 'SMSEventSource',  
  onDeviceSubscribe: 'onDeviceSubscribeCallback',  
  onDeviceUnsubscribe: 'onDeviceUnsubscribeCallback',  
  securityTest: 'SMSRealm-mobile-securityTest'  
});
```

通知をバックエンド・システムからポーリングする

```
WL.Server.createEventSource({  
  name: 'SMSEventSource',  
  onDeviceSubscribe: 'onDeviceSubscribeCallback',  
  onDeviceUnsubscribe: 'onDeviceUnsubscribeCallback',  
  securityTest: 'SMSRealm-mobile-securityTest',  
  poll: {  
    interval: 3,  
    onPoll: getNotificationsFromBackend  
  }  
});
```

- **name** – イベント・ソースの参照に使用する名前
- **onDeviceSubscribe** – ユーザー・サブスクリプション要求の受信時に呼び出すアダプター関数
- **onDeviceUnsubscribe** – ユーザー・アンサブスクライブ要求の受信時に呼び出すアダプター関数
- **securityTest** – イベント・ソースの保護に使用する authenticationConfig.xml からのセキュリティー・テスト

通知 API: サーバー・サイド

- 最初にイベント・ソースを作成する

- アダプター JavaScript コードで通知イベント・ソースをグローバル・レベル (JavaScript 関数外) で宣言します。

通知をバックエンド・システムによってプッシュする

```
WL.Server.createEventSource({
  name: 'SMSEventSource',
  onDeviceSubscribe: 'onDeviceSubscribeCallback',
  onDeviceUnsubscribe: 'onDeviceUnsubscribeCallback',
  securityTest: 'SMSRealm-mobile-securityTest'
});
```

通知をバックエンド・システムからポーリングする

```
WL.Server.createEventSource({
  name: 'SMSEventSource',
  onDeviceSubscribe: 'onDeviceSubscribeCallback',
  onDeviceUnsubscribe: 'onDeviceUnsubscribeCallback',
  securityTest: 'SMSRealm-mobile-securityTest',
  poll: {
    interval: 3,
    onPoll: getNotificationsFromBackend
  }
});
```

- **poll** – 通知の取り出しに使用するメソッド。以下のパラメーターが必要です。
 - **Interval**: ポーリング間隔 (秒)
 - **onPoll**: ポーリング実装: 指定された間隔で呼び出されるアダプター関数

通知 API: サーバー・サイド

- 通知を送信する

```
function sendSMS(userId, smsText){
    var userSubscription =
        WL.Server.getUserNotificationSubscription('SMSAdapter.SMSEventSource', userId);

    if (userSubscription==null){
        return { result: "No subscription found for user :: " + userId };
    }

    var badgeDigit = 1;

    var notification = WL.Server.createDefaultNot

    WL.Logger.debug("sendSMS >> userId :: " + use

    WL.Server.notifyAllDevices(userSubscription,

    return {
        result: "Notification sent to user :: " +
    };
}
```

通知は、バックエンド・システムからポーリングするか、バックエンド・システムによってプッシュすることができます。

この例では、**sendSMS()** アダプター関数が、通知を送信する外部 API としてバックエンド・システムによって呼び出されます。

通知 API: サーバー・サイド

- 通知を送信する
 - 通知データを取得します。

```
function sendSMS(userId, smsText){  
    var userSubscription =  
        WL.Server.getNotificationSubscription('SMSAdapter.SMSEventSource', userId);  
  
    if (userSubscription == null){  
        return { result: "No subscription found for user :: " + userId };  
    }  
  
    var badgeDigit = 1;  
  
    var notification = WL.Server.createDefaultNotification(userId, smsText, badgeDigit);  
  
    WL.Logger.debug("sendSMS >> userId :: " + userId);  
  
    WL.Server.notifyAllDevices(userSubscription, notification);  
  
    return {  
        result: "Notification sent to user :: " + userId;  
    };  
}
```

sendSMS 関数が、通知の送信先ユーザー ID と、SMS テキストを引数として受け取ります。

これらの引数は、この関数を呼び出すバックエンド・システムによって提供されます。

通知 API: サーバー・サイド

- 通知を送信する
 - アクティブ・ユーザーを検索し、それを使用してユーザー・サブスクリプション・データを取得します。

```
function sendSMS(userId, smsText){  
    var userSubscription =  
        WL.Server.getUserNotificationSubscription('SMSAdapter.SMSEventSource', userId);  
  
    if (userSubscription==null){  
        return { result: "No subscription found for user :: " + userId };  
    }  
}
```

```
var badgeDigit = 1;  
  
var notification = WL.Server.  
WL.Logger.debug("sendSMS : ");  
WL.Server.notifyAllDevices(  
  
return {  
    result: "Notification"  
};  
}
```

ユーザー・サブスクリプション・オブジェクトにはすべてのユーザー・サブスクリプションに関する情報が含まれています。各ユーザー・サブスクリプションは複数のデバイス・サブスクリプションを持つことができます。

指定されたイベント・ソースに対するサブスクリプションをユーザーが持っていない場合、ヌル・オブジェクトが返されます。

通知 API: サーバー・サイド

- 通知を送信する
 - ユーザー・サブスクリプション・データを取得します。

```
[  
  {  
    "alias": "myPushSMS",  
    "applicationId": "SMSPushApp-windowsphone-1.0",  
    "device": "4811C50B6D106DCA354D52BBFFFFFFFF",  
    "eventSourceId": "SMSPushAdapter.SMSPushEventSource",  
    "options": {  
    },  
    "platform": "SMS",  
    "token": "3656886544",  
    "userAgent": "SMS"  
  }  
]
```

getDeviceSubscriptions API を使用して、各ユーザー・デバイスのサブスクリプション・データを個別に取得することができます。

結果は、ここに示す構造を持つオブジェクトの配列です。

通知 API: サーバー・サイド

- 通知を送信する
 - 通知をユーザー・デバイスに送信します。

```
function sendSMS(userId, smsText){
    var userSubscription =
        WL.Server.getUserNotificationSubscription('SMSAdapter.SMSEventSource', userId);

    if (userSubscription==null){
        return { result: "No subscription found for user :: " + userId };
    }

    var badgeDigit = 1;

    var notification = WL.Server.createDefaultNotification(smsText, badgeDigit, {});

    WL.Logger.debug("sendSMS >> userId :: " + userId + ", text :: " + smsText);

    WL.Server.notifyAllDevices(userSubscription, notification);

    return {
        result: "Notification sent to user :: " +
    };
}
```

WL.Server.notifyAllDevices
API は通知を、サブスクライブ
されたすべてのユーザー・
デバイスに送信します。

通知 API: サーバー・サイド

- 通知を送信するには、以下の API を使用します。
 - **WL.Server.notifyAllDevices(userSubscription, options):** 通知をすべてのユーザー・デバイスに送信します。(前のスライドを参照)
 - **WL.Server.notifyDevice(userSubscription, device, options):** 通知を特定のユーザー・サブスクリプションの特定のデバイスに送信します
 - **WL.Server.notifyDeviceSubscription(deviceSubscription, options):** 通知を特定のデバイスに送信します

通知 API: クライアント・サイド

- イベント・ソース – サブスクライブとアンサブスクライブ
 - サブスクライブするために、ユーザーはまず認証されていることが必要です。
 - 以下の API を使用してイベント・ソースにサブスクライブします。

```
function subscribeSMSButtonClicked() {
    WL.Client.Push.subscribeSMS("myPushSMS", "SMSPushAdapter",
        "SMSPushEventSource", "919844634XXX", {
        onSuccess: subscribeSMSOnSuccess,
        onFailure: subscribeSMSOnFailure
    });
}
function subscribeSMSOnFailure(response) {
    alert("subscribeSMS OnFailure");
}
function unsubscribeSMSOnSuccess(response) {
    alert("unsubscribeSMS OnSuccess");
}
```

- **WL.Client.Push.subscribeSMS()** には以下のパラメーターがあります。
 - サブスクリプションを識別するための別名
 - イベント・ソースが定義されているアダプター名
 - クライアントがサブスクライブしているイベント・ソース名
 - 通知が送信されるユーザー携帯電話番号
 - 携帯電話番号には、数字 (0 から 9)、正符号 (+)、負符号 (-)、およびスペース文字 () のみを使用できます。
 - オプションの **onSuccess** コールバック
 - オプションの **onFailure** コールバック
- コールバックは、必要な場合は応答オブジェクトを受け取ります。

通知 API: クライアント・サイド

- イベント・ソース – サブスクライブとアンサブスクライブ
 - 以下の API を使用してイベント・ソースからアンサブスクライブします。

```
function unsubscribeSMSButtonClicked() {
    WL.Client.Push.unsubscribeSMS("myPushSMS", {
        onSuccess: unsubscribeSMSOnSuccess,
        onFailure: unsubscribeSMSOnFailure
    });
}

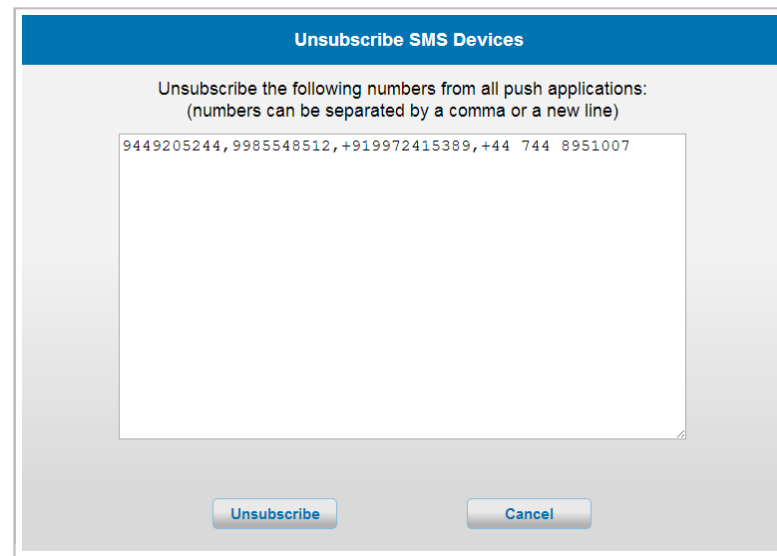
function unsubscribeSMSOnSuccess(response) {
    alert("unsubscribeSMS OnSuccess");
}

function unsubscribeSMSOnFailure(response) {
    alert("unsubscribeSMS OnFailure");
}
```

- **WL.Client.Push.unsubscribeSMS** には以下のパラメーターがあります。
 - 別名 - WL.Client.Push.subscribeSMS で定義されているものと同じ
 - オプションの **onSuccess** コールバック
 - オプションの **onFailure** コールバック
- コールバックは、必要な場合は応答オブジェクトを受け取ります。

通知 API: クライアント・サイド

- アンサブスクライブは、IBM Worklight Console を使用して行うこともできます。アンサブスクライブする携帯電話番号を指定します (電話番号が複数の場合はコンマで区切ります)。



Unsubscribe SMS Devices

Unsubscribe the following numbers from all push applications:
(numbers can be separated by a comma or a new line)

9449205244,9985548512,+919972415389,+44 744 8951007

Unsubscribe Cancel

- 追加クライアント・サイド API:
 - **WL.Client.Push.isPushSMSSupported()** – SMS 通知がプラットフォームでサポートされている場合は **true** を、サポートされていない場合は **false** を返します。
 - **WL.Client.Push.isSMSSubscribed(alias)** – 現在ログインしているユーザーが指定のイベント・ソース別名にサブスクライブしている場合は **true** を、サブスクライブしていない場合は **false** を返します。

アジェンダ

- SMS 通知とは
- デバイス・サポート
- 通知アーキテクチャー
- サブスクリプション管理
- SMS 通知 API
- SMS サブスクライブ・サブレット
- セットアップ

SMS サブスクライブ・サーブレット

- SMS サブスクライブ・サーブレットを使用して SMS 通知にサブスクライブします。
- ユーザーは、サーブレットに対して HTTP GET 呼び出しを行うことで、デバイス上のアプリケーションなしで SMS 通知にサブスクライブすることができます。
- SMS サブスクライブ・サーブレットでは、サブスクライブとアンサブスクライブの両方のオプションがサポートされています。
- 次の URL を使用して SMS サブスクライブ・サーブレットにアクセスしてください。
 - `http://<hostname>:<port>/<context>/subscribeSMS?<parameters>`
- サーブレット呼び出しを行う前に、アダプター内のアプリケーションとイベント・ソースが Worklight Server にデプロイされている必要があります。

SMS サブスクライブ・サーブレット: URL パラメーター

- 以下のパラメーターを SMS サブスクライブ・サーブレットに渡すことができます。

URL パラメーター	使用法
option	オプション。デフォルト値は <code>subscribe</code> です。 <code>subscribe</code> 以外の非空白・ストリングを指定した場合、アンサブスクライブ・アクションが実行されます。
eventSource	必須。 <code>AdapterName.EventSourceName</code> 。
alias	オプション。イベント・ソース名を定義する短縮 ID。
phoneNumber	必須。SMS 通知の送信先の番号。
username	オプション。値を指定しない場合は、電話番号がユーザー名として使用されます。
appId	必須。SMS ゲートウェイ定義を含むアプリケーションの ID。 例: <code>SMSPushApp-android-1.0</code>

SMS サブスクライブ・サブレット: サブスクライブ URL と応答

- サブスクライブ URL:

`http://<host>:<port>/<context>/subscribeSMS?alias=<aliasName>&appld=<appld>&eventSource=<evenSourceName>&username=<username>&phoneNumber=<number>`

- Worklight Server からの応答例:

```
·{"result":"FWLSE00146I: User 'vk' with phone number '0987654321' successfully subscribed to SMS notifications","isSuccessful":"true"}
```

SMS サブスクライブ・サブレット: アンサブスクライブ URL と応答

- アンサブスクライブ URL:

`http://<host>:<port>/<context>/subscribeSMS?option=unsubscribe&eventSource=SMSAdapter.SMSEventSource&userName=<username>&phoneNumber=<phonenumber>`

- Worklight Server からの応答例:

```
{"result":"FWLSE00147I: Successfully unsubscribed from SMS notifications for user 'vk',"isSuccessful":"true"}
```

SMS サブスクライブ・サーブレット: 特長

- SMS サブスクライブ・サーブレットを使用して作成されたサブスクリプションは、デバイスによって作成されたサブスクリプションから独立しています。
- SMS サブスクライブ・サーブレットを使用してアンサブスクライブを行っても、デバイスから作成されたサブスクリプションは削除されません。
- セキュリティー
 - デフォルトでは、SMS サブスクライブ・サーブレットは静的リソースとして保護されます。
 - デフォルトでは、authenticationConfig.xml ファイルは、SMS サブスクライブ・サーブレットに対するすべての要求を拒否ログイン・モジュールを使用して拒否するように構成されています。
 - SMS サブスクライブ・サーブレットへのアクセスを許可するには、管理者が適切なオーセンティケータとログイン・モジュールを使用して authenticationConfig.xml ファイルを変更する必要があります。

SMS サブスクライブ・サーブレット: セキュリティー

- デフォルトでは、authenticationConfig.xml は、SMS サブスクライブ・サーブレットへのすべての要求を拒否するように構成されています。

```
<resource id="subscribeServlet" securityTest="SubscribeServlet">
  <urlPatterns>/subscribeSMS*</urlPatterns>
</resource>

<customSecurityTest name="SubscribeServlet">
  <test realm="SubscribeServlet" isInternalUserID="true" />
</customSecurityTest>

<realm name="SubscribeServlet" loginModule="rejectAll">
  <className>com.worklight.core.auth.ext.HeaderAuthenticator</className>
</realm>

<loginModule name="rejectAll">
  <className>com.worklight.core.auth.ext.RejectingLoginModule</className>
</loginModule>
```

- サーブレットがデフォルト認証構成を使用して呼び出された場合。

```
/*-secure-
{"WL-Authentication-Failure":{"SubscribeServlet":{"reason":"illegal state"}}}*/
```

SMS サブスクライブ・サーブレット: セキュリティー

- 管理者は、SMS サブスクライブ・サーブレットにアクセスできるように authenticationConfig.xml を変更することができます。

```
<resource id="subscribeServlet" securityTest="SubscribeServlet">
  <urlPatterns>/subscribeSMS*</urlPatterns>gular Snip
</resource>

<customSecurityTest name="SubscribeServlet">
  <test realm="SMSRealm" isInternalUserID="true" />
</customSecurityTest>

<realm name="SMSRealm" loginModule="smsHeader">
  <className>com.worklight.core.auth.ext.HeaderAuthenticator</className>

<loginModule name="smsHeader">
  <className>com.worklight.core.auth.ext.HeaderLoginModule</className>
  <parameter name="user-name-header" value="vk"></parameter>
</loginModule>
```

- サーブレットがカスタム認証構成を使用して呼び出された場合。

```
{"result":"FWLSE00146I: User 'vk' with phone number '0987654321' successfully subscribed to SMS notifications","isSuccessful":"true"}
```

アジェンダ

- SMS 通知とは
- デバイス・サポート
- 通知アーキテクチャー
- サブスクリプション管理
- SMS 通知 API
- SMS サブスクライブ・サーブレット
- セットアップ

セットアップ

■ SMSConfig.xml

- SMS 通知を送信するには、Worklight Server でサポートされるサード・パーティー・ゲートウェイ・サービスを SMSConfig.xml ファイルで定義します。

```
<?xml version="1.0" encoding="UTF-8"?>
<sms:config xmlns:sms="http://www.worklight.com/sms/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <gateway id="myGateway" hostname="yourhostname.com" port="80" programName="backendProgram"
    toParamName="to" textParamName="text">
    <!-- <parameter name="" value=""/> -->
  </gateway>
</sms:config>
```

■ application-descriptor.xml

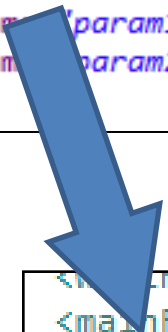
- アプリケーションで SMS 通知を使用するには、<smsGateway id="..."/> エレメントを application-descriptor.xml ファイルに追加することによって、使用する SMS ゲートウェイを指定します。

```
<smsGateway id="myGateway"/>
```


セットアップ

- SMSConfig.xml ファイル内のゲートウェイ ID は、application-descriptor.xml ファイル内の SMS ゲートウェイ ID に対応する必要があります。

```
<gateway id="myGateway" hostname="yourhostname.com" port="80" programName="backendProgram"  
  toParamName="to" textParamName="text">  
  <parameter name="param1name" value="param1value"/>  
  <parameter name="param2name" value="param2value"/>  
</gateway>
```

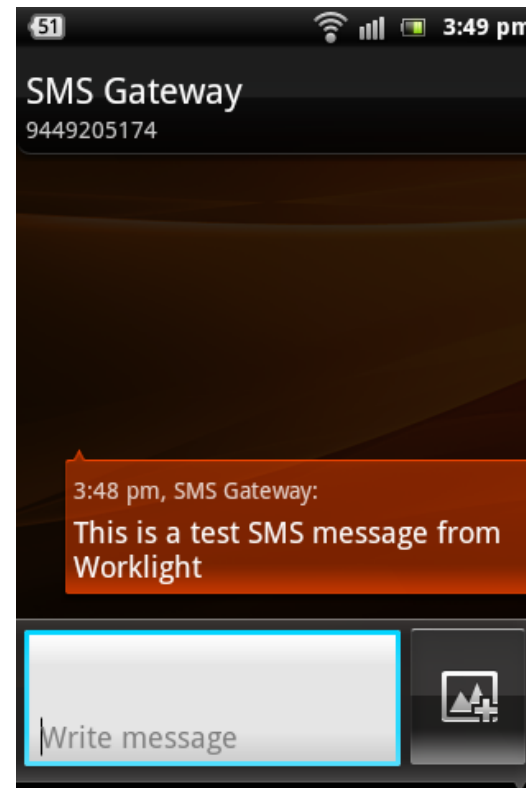


```
<mainFile>SMSReceiverApp.html</mainFile>  
<thumbnailImage>common/images/thumbnail.  
<smsGateway id="myGateway"/>  
<android version="1.0" securityTest="SMS"  
  <worklightSettings include="true"/>  
<security>  
  <encryptWebResources enabled="fa  
  <testWebResourcesChecksum enable
```

結果

- このトレーニング・モジュールのサンプルは、IBM Worklight Foundation 文書 (<http://www.ibm.com/mobile-docs>) の「入門」ページにあります。
- プッシュ通知をアクティブにするには、SMS ゲートウェイ・サブスクリプションが必要であることに注意してください。

The screenshot shows a web interface titled "SMS Notifications". It contains several interactive elements: a button labeled "is SMS Supported?", a button labeled "is SMS Subscribed?", a text input field for "Phone Number" with the value "1234567890", a button labeled "Subscribe SMS", and a button labeled "Unsubscribe SMS".



バックエンド・エミュレーター

- このトレーニング・モジュールのサンプルには、バックエンド・システム・エミュレーターが付属しています。このエミュレーターを使用して、バックエンド・システムによる SMS 通知送信をシミュレートすることができます。
- バックエンド・システム・エミュレーターを実行するには、以下のようになります。
 1. コマンド・プロンプトを開きます。
 2. サンプル・プロジェクトの `SMSBackendEmulator` フォルダーに移動します。
 3. 次のコマンドを入力して、提供されている `SMSBackendEmulator.jar` ファイルを実行します。

```
java -jar SMSBackendEmulator.jar <userId> <notificationText> <serverPort>
```

- `<userId>` は、サンプル・アプリケーションへのログインに使用したユーザー名です。

例:

```
java -jar SMSBackendEmulator.jar user1 "hello from sms" 8080
```

バックエンド・エミュレーター

- バックエンド・システム・エミュレーターは、Worklight Server への接続と、**sendSMS()** アダプター・プロシーチャーの呼び出しを試行します。この後、コマンド・プロンプト・コンソールに呼び出し結果が出力されます。

- 成功

```
C:\worklight\workspace\test\module_42_0_SMSNotifications\SMSBackendEmulator>java
-jar SMSBackendEmulator.jar user1 "hello from sms" 8080
SMSBackendEmulator
User Id: user1
Notification text: hello from sms
Server URL: http://localhost:8080
sending notification
Server response :: {  "isSuccessful": true,  "result": "Notification sent to u
ser :: user1"}
```

- 失敗

```
C:\worklight\workspace\test\module_42_0_SMSNotifications\SMSBackendEmulator>java
-jar SMSBackendEmulator.jar user1 "hello from sms" 8080
SMSBackendEmulator
User Id: user1
Notification text: hello from sms
Server URL: http://localhost:8080
sending notification
Server response :: {  "isSuccessful": true,  "result": "No subscription found
for user :: user1"}
```

特記事項

- これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。
- 本書は米国 IBM が提供する製品およびサービスについて作成したものです。
- 本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、または サービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。
- IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。
 - 〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

- この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。
- 本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。
- IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。
- 本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム（本プログラムを含む）との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

- IBM Corporation
Dept F6, Bldg 1
294 Route 100
Somers NY 10589-3216
USA

- 本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。
- 本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。
- IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

著作権使用許諾:

- 本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめめしたり、保証することはできません。
- それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。
 - © (お客様の会社名) (西暦年) このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. 年を入れる。 All rights reserved.

プライバシー・ポリシーの考慮事項

- サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品（「ソフトウェア・オファリング」）では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的な事項を確認ください。
- このソフトウェア・オファリングは、展開される構成に応じて、(アプリケーション・サーバーが生成する) セッション情報を収集するセッションごとの Cookie を使用場合があります。これらの Cookie は個人情報を含まず、セッション管理のために要求されるものです。加えて、匿名ユーザーの認識および管理のために持続的な Cookie が無作為に生成される場合があります。これらの Cookie も個人情報を含まず、要求されるものです。
- この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、IBM の『IBM オンラインでのプライバシー・ステートメント』(<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』(<http://www.ibm.com/software/info/product-privacy>) を参照してください。

サポートおよびコメント

- IBM Worklight の一連の文書、トレーニング資料、および質問をポストできるオンライン・フォーラムはすべて、次の IBM Web サイトからご覧になれます。
 - <http://www.ibm.com/mobile-docs>
- サポート
 - ソフトウェア・サブスクリプション & サポート (ソフトウェア・メンテナンスと呼ばれる場合もあります) は、パスポート・アドバンテージおよびパスポート・アドバンテージ・エクスプレスから購入されたライセンスに含まれています。International Passport Advantage Agreement および IBM International Passport Advantage Express Agreement の追加情報については、次のパスポート・アドバンテージ Web サイトを参照してください。
 - <http://www.ibm.com/software/passportadvantage>
 - ソフトウェア・サブスクリプション & サポートが有効になっている場合、IBM は、インストールおよび使用法 (ハウツー) に関する短期間の FAQ に対するサポートや、コード関連の質問に対するサポートを提供します。詳しくは、次の IBM ソフトウェア・サポート・ハンドブックを参照してください。
 - <http://www.ibm.com/support/handbook>
- ご意見
 - 本資料に関するご意見をお寄せください。本資料の具体的な誤りや欠落、正確性、編成、題材、または完成度に関するご意見をお寄せください。お寄せいただくご意見は、本マニュアルまたは製品の情報、およびその情報の提示方法に関するもののみとさせていただきます。
 - 製品の技術的な質問および情報、および価格については、担当の IBM 営業所、IBM ビジネス・パートナー、または認定リマーカーターにお問い合わせください。
 - IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。IBM またはいかなる組織も、お客様から提示された問題についてご連絡を差し上げる場合にのみ、お客様が提供する個人情報を使用するものとします。
 - どうぞよろしくお願いいたします。
 - 次の IBM Worklight Developer Edition サポート・コミュニティにご意見をお寄せください。
 - <https://www.ibm.com/developerworks/mobile/worklight/connect.html>
 - IBM からの回答を希望される場合は、以下の情報をご連絡ください。
 - 氏名
 - 住所
 - 企業または組織
 - 電話番号
 - E メール・アドレス

ありがとうございました

