Two-way SMS communication

Overview

Two-way SMS communication enables communication between mobile phones and IBM® MobileFirst® Server over an SMS channel. Mobile-originated SMS messages can be sent to MobileFirst Server via the SMS gateway and MobileFirst Server can send a response back.
Short codes or keywords are configured with a third-party SMS gateway. The gateway must be configured to forward the requests to MobileFirst Server or to a reverse proxy URL, depending on the topology of your environment.
The SMS messages that are sent from mobile devices are forwarded to an adapter procedure on the MobileFirst Server side, as configured by the developer.
The adapter procedure includes the logic to process the request, or the procedure can forward the request to a back end for processing. The response is sent back through the MobileFirst notification framework.

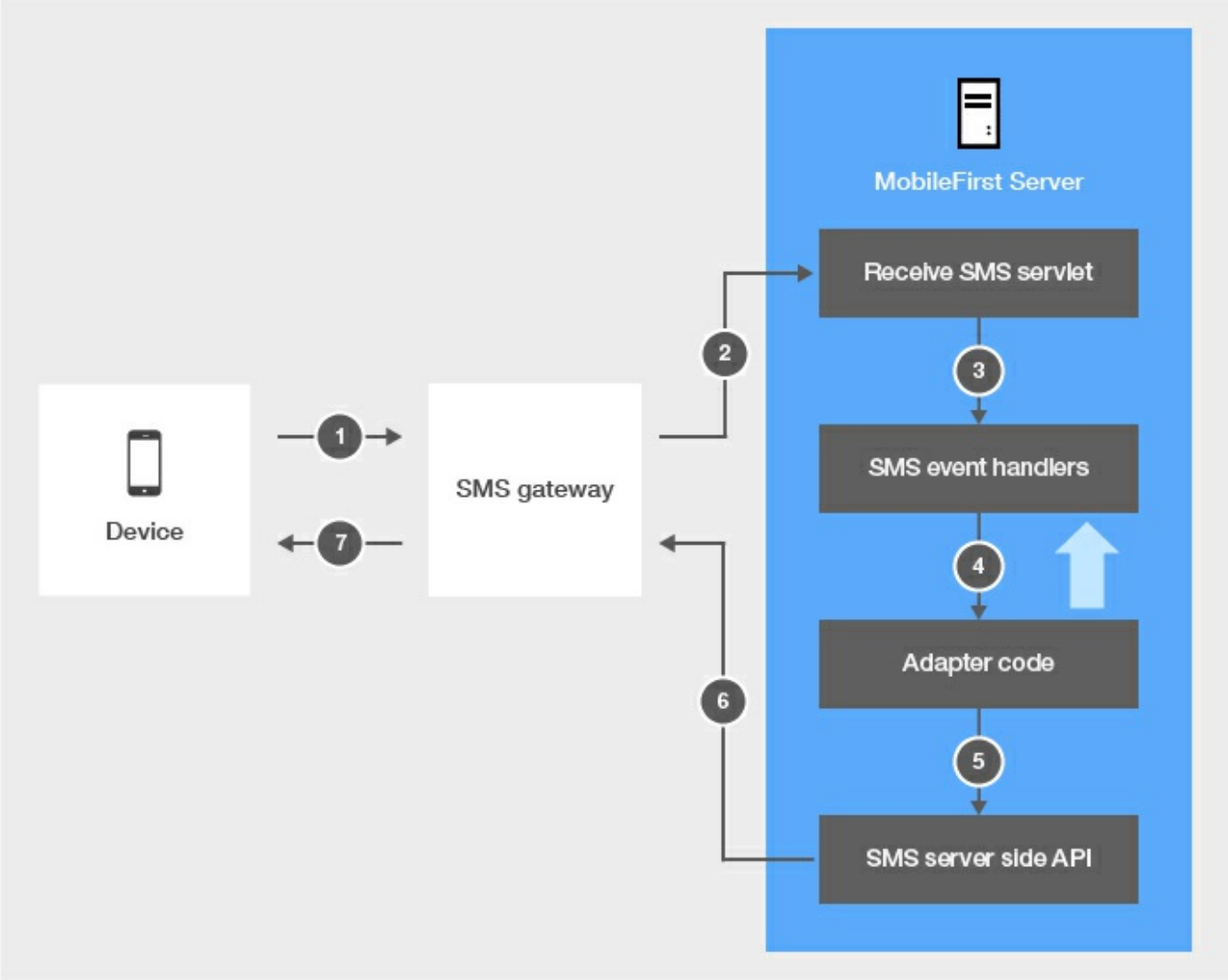
The following topics are covered:

- Device support
- Architecture
- API for two-way SMS
- Setup
- Sample application

Device support

IBM® MobileFirst Platform Foundation® supports two-way SMS communication on any mobile devices that support SMS functions.

Architecture



An adapter registers SMS event handlers on the MobileFirst Server. The event handler registration specifies what
adapter procedure callback is called.

SMS messages are sent from mobile devices to the SMS gateway, which is configured with a MobileFirst project. The gateway details are specified in the `SMSConfig.xml` file, in the MobileFirst project.

The SMS gateway forwards SMS messages to the configured MobileFirst URL. Configure the SMS gateway with the keyword and the MobileFirst Server URL to which it must forward the messages. For more information, refer to the specification of your gateway.

An SMS servlet on the MobileFirst Server matches the parameters with filters as defined in SMS event handlers, and calls the adapter procedure callback.

The adapter can process SMS messages and send them to a mobile device by using the SMS API. An adapter procedure can use the SMS server-side APIs for subscription and unsubscription. The procedure uses the SMS notification framework to send the response back to the mobile device.

API for two-way SMS

Registering SMS event handlers

```
WL.Server.setEventHandlers([
   WL.Server.createSMSEventHandler(<br />
      {keyword: WL.Server.configuration['sms.keyword.subscribe']},
      subscribeSMS,
      WL.Server.configuration['sms.gateway.id']
      ),
      WL.Server.createSMSEventHandler(
         {keyword: WL.Server.configuration['sms.keyword.unsubscribe']
},
      unsubscribeSMS, WL.Server.configuration['sms.gateway.id']
   )
]);
```

The name-value parameters sent from the SMS gateway that match the filters are passed to `handlerFunction`. For example, if the SMS gateway is sending the following parameters:

- `from`: The number that sends the SMS
- `text`: The SMS text
- `keyword`: A keyword configured in the SMS gateway

The `filter` parameter can be defined as `{keyword:"myKey"}`. In this case, when MobileFirst Server receives a request from the SMS gateway, it looks for a keyword with `myKey` as its value, and forwards the resulting request to `handlerFunction`.

`handlerFunction` receives an SMS event that has all parameters sent from the SMS gateway.

`gatewayID` corresponds to the SMS gateway that is configured in the `SMSConfig.xml` file. This gateway is used to receive SMS messages.

Server-side SMS API

MobileFirst Server uses SMS notification to send SMS messages. Before SMS messages can be sent, the server must subscribe to a phone number.

```
var subscription = WL.Server.subscribeSMS(eventSourceID, phoneNumber, gatewayID);
```

This method takes the following parameters:

- `eventSourceId`: The event source to which the phone number is subscribing.
- `phoneNumber`: The user's phone number
- `gatewayID`: The ID that identifies the gateway through which SMS messages are sent.

The method returns an SMS subscription object. Along with the `WL.Server.notifyDeviceSubscription` method, this object can be used to send an SMS message to the specified phone number.

If the phone number is already subscribed, get the subscription object and send an SMS message.

```
function sendSMS(phoneNumber, smsText) {
    var subscription = WL.Server.getSMSSubscription(eventSourceId, phoneNumber)
;
    if (subscription == null) {
        return { result: "No subscription found for user :: " + phoneNumber };
    }

    var notification = WL.Server.createDefaultNotification(smsText, 1, {});
    WL.Server.notifyDeviceSubscription(subscription, notification);
    return {
        result: "Notification sent to user :: " + phoneNumber
    };
}
```

A phone number can unsubscribe from the event source by a call to the following method:

```
WL.Server.unsubscribeSMS(eventSourceID, phoneNumber);
```

Setup

The `SMSConfig.xml` file specifies gateway details for both receiving and sending SMS messages.

```
<gateway id="myGateway" hostname="yourhostname.com" port="80" programName="backendProgra
m" toParamName="to" textParamName="text" fromParamName="from">
    <parameter encode="false" name="param1name" value="param1value"/>
    <parameter encode="false" name="param2name" value="param2value"/>
</gateway>
```

You can externalize the keywords, gateway identifier, and other parameters that the SMS event handler uses in the `worklight.properties` configuration file.

```
sms.keyword.subscribe=sub
sms.keyword.unsubscribe=unsub
sms.gateway.id=myGateway
```

By default, the `SubscribeServlet` object that receives the request from the gateway is protected by a security test. This security test takes a `RejectingLoginModule` parameter which by default rejects any request. To enforce the security mechanism that is supported by your gateway, comment out, or modify the default security test.

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

If you are using a reverse proxy in front of MobileFirst Server, you can specify a mapping to the reverse proxy to allow only requests from specific hosts or IP addresses (SMS gateway). If the application server that is running MobileFirst directly receives the request from a gateway, the application server needs to be configured to allow requests from the specific hosts or IP addresses.

Sample application

Click to download (https://github.com/MobileFirst-Platform-Developer-Center/SMS2WayProject/tree/release71) the MobileFirst project.

1. A user subscribes to SMS notification by sending the keyword `sub` from his mobile device to the predefined short code or phone number that is registered with a gateway from an independent software vendor.
2. The `BookStoreAdapter` adapter subscribes the phone number to receive notifications.
3. The `BookStoreAdapter` adapter sends a notification to the user when there is a discount offer.
4. The user can unsubscribe for SMS notification by sending the `unsub` keyword to the predefined short code or phone number.