
Unicredit - SMS Estero

Analisi Tecnica

Versione:	1.0
Data:	14/03/2012

CRONOLOGIA DELLE VERSIONI		
Numero versione	Data versione	Sintesi delle variazioni
1.0	14/03/2012	Prima stesura

Indice

1	Scopo del documento.....	4
2	La piattaforma ubimessaging.....	4
3	Ciclo di vita di un messaggio MT.....	6
4	Architettura	7
5	Flussi generali.....	8
5.1	Flusso messaggi push.....	8
5.2	Flusso messaggi mandatori.....	10
5.3	Flusso per il recupero degli status report.....	12
6	Templates di configurazione.....	12
6.1	Configurazione applicazione ubmPushUnicredit.....	13
	sms_billing_push_ev.xml.....	13
	template_sms_push_ev.xml.....	13
	scadenze_sms_push_ev.xml.....	13
6.2	Configurazione applicazione ubmMndUnicredit.....	14
	mapping_errori.xml.....	14
	mandatory_routers.xml.....	14
7	Tracciati per il colloquio con altre applicazioni.....	15
7.1	Tracciato ricevuto dalla banca su coda MQ.....	15
7.2	tracciato per la ricezione di uno status report da Hermess.....	18
7.3	tracciato per l'invio di un messaggio MT ad Hermess.....	18
7.4	Ricezione da Hermess di un messaggio mandatorio non spedito.....	19
7.5	invio ad Hermess di un messaggio da spedire sull'operatore di back up.....	20
7.6	risposta di Hermess nel caso di mancata spedizione del messaggio sul canale di back up.....	20

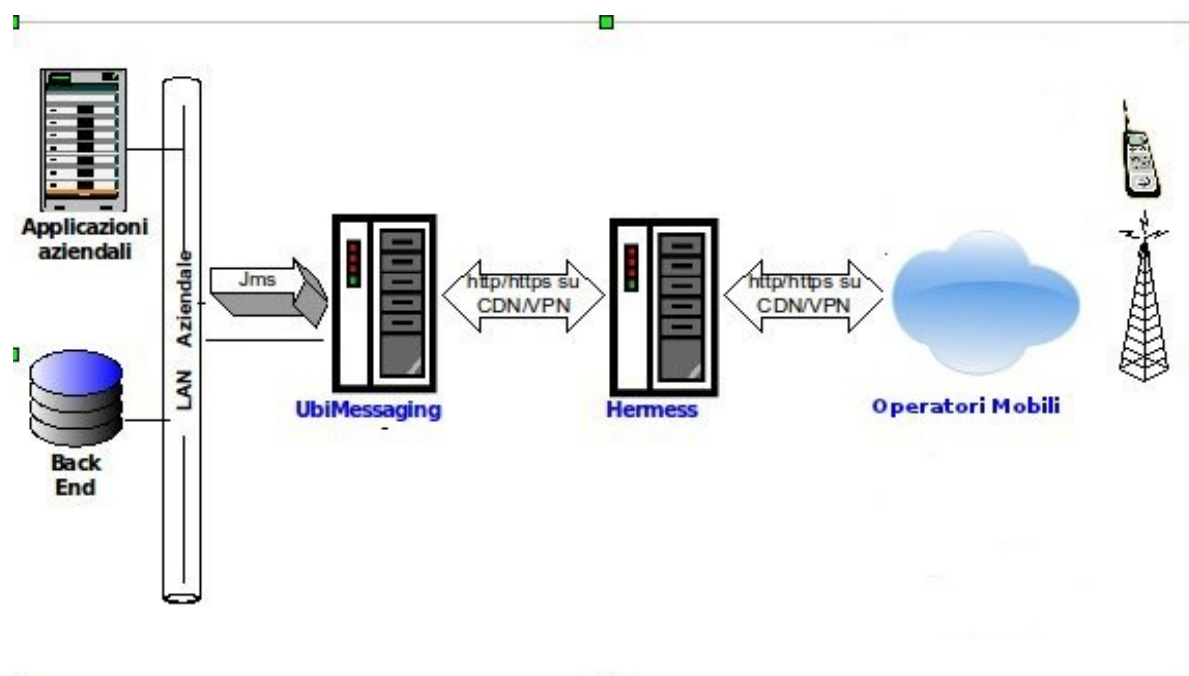
1 SCOPO DEL DOCUMENTO

Scopo del documento è descrivere la piattaforma Ubimessaging installata presso il data center di Unicredit Banca per l'invio di messaggi bulk relativi all'accesso in sicurezza ai sistemi della banca.

La piattaforma permette l'invio di messaggi bulk in Italia e all'estero. Per l'invio dei messaggi la piattaforma comunicherà con il centro stella Hermess.

2 LA PIATTAFORMA UBIMESSAGING

Nel disegno viene rappresentato uno schema astratto dell'architettura necessaria per l'invio dei messaggi sms. Nel disegno sono quindi rappresentate le applicazioni della banca, la piattaforma Ubimessaging, il centro stella Hermess e gli operatori telefonici.



La piattaforma Ubimessaging installata in banca è costituita dai seguenti moduli ear:

- ubmPushUnicredit (per la presa in carico dei messaggi MT inviati dalla banca)

- ubmNotificheUnicredit (per la presa in carico e la scrittura su base dati locale degli status report inviati dagli operatori telefonici)
- ubmConsoleUnicredit (console web per la ricerca dei messaggi e la reportistica)
- ubmMandatoryUnicredit (per l'invio dei messaggi in fail over in caso di indisponibilità di un operatore telefonico)

Per l'installazione della piattaforma sono richiesti:

- la disponibilità di una base dati locale dove verranno create una/due tabelle ed alcune stored procedures (è previsto l'utilizzo di circa 250 Mb ogni 100.000 sms gestiti, per una stima più precisa si può valutare lo spazio occupato dalla base dati IBZ attualmente)
- la disponibilità di 3 code jms che vengono scritte e lette localmente (ubmMnd, ubmRetryPush, ubmRetryMnd)
- la disponibilità di una coda jms per la comunicazione con la banca
- l'eventuale generazione di un certificato client nel caso in cui la comunicazione con il centro stella Hermess debba avvenire con autenticazione del client
- due jvm. Si consiglia l'installazione della piattaforma su due distinte jvm in modo da separare gli ear che si occupano della spedizione degli sms da quelli che si occupano della gestione degli aggiornamenti di stato dei messaggi.
- una cartella per la scrittura dei log applicativi (**meglio 4**, una per ognuno degli ear installati)
- una cartella da cui leggere i templates di configurazione (**meglio 4**, una per ognuno degli ear installati). Nel caso di installazione in cluster questa cartella può essere su una Nas condivisa dalle macchine nel cluster
- una cartella da cui leggere files di configurazione con puntamenti ad applicazioni esterne (meglio 4, una per ognuno degli ear installati). Nel caso di

installazione in cluster si può valutare di mettere queste cartelle su una Nas condivisa dalle macchine nel cluster

- la disponibilità di risorse jndi di tipo Url, Jms, Jdbc, Timer su WebSphere (in fase di deploy saranno definite meglio)

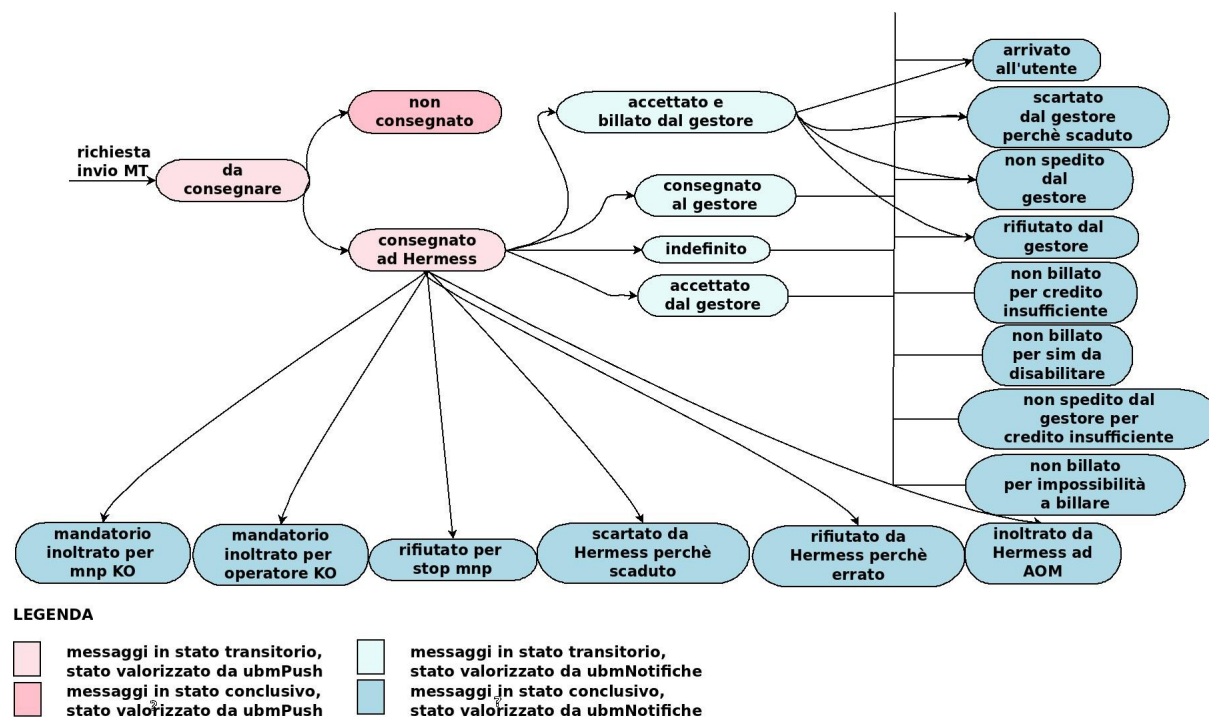
La piattaforma sarà installata su application server IBM Websphere 7.

3 CICLO DI VITA DI UN MESSAGGIO MT

Il ciclo di vita di un messaggio MT ha inizio quando arriva ad ubmPush la richiesta di spedizione del messaggio.

Il messaggio può trovarsi in uno stato transitorio (anche detto parziale) o conclusivo. L'aggiornamento di stato dei messaggi MT avviene attraverso operazioni sulla tabella SMS_OUT del database ubimessaging da parte delle applicazioni ubmPush e ubmNotifiche. La struttura del database è illustrata in seguito nel presente documento.

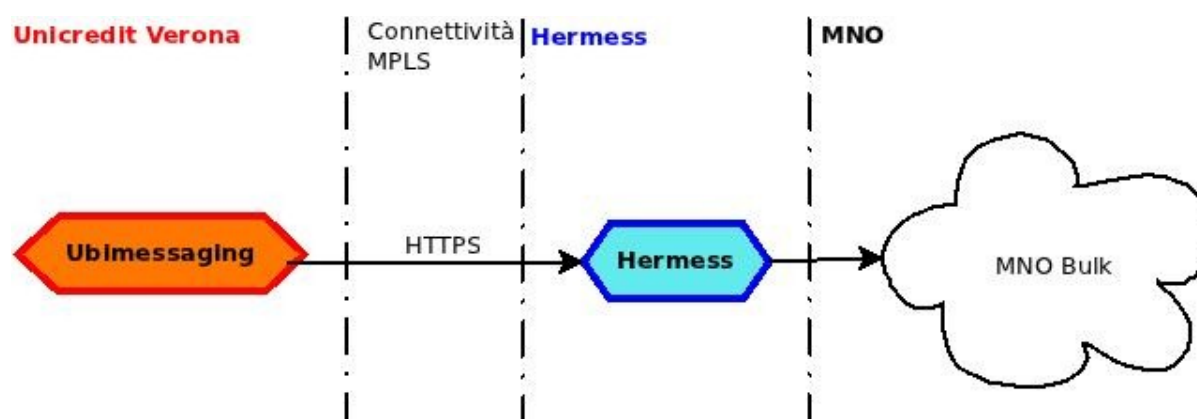
Nella figura seguente sono illustrati i passaggi di stato di un messaggio MT:



Nei paragrafi seguenti sono descritte le varie applicazioni di cui si compone la piattaforma. Per ogni applicazioni è specificato il tipo di interazione con il database.

4 ARCHITETTURA

Nell' architettura scelta è presente la gestione di un operatore di backup che viene utilizzato per la spedizione dei messaggi in caso di indisponibilità dell'operatore primario ma non sono gestiti canali di back up in caso di indisponibilità del centro stella Hermess. Nel disegno è rappresentata l'architettura:



5 FLUSSI GENERALI

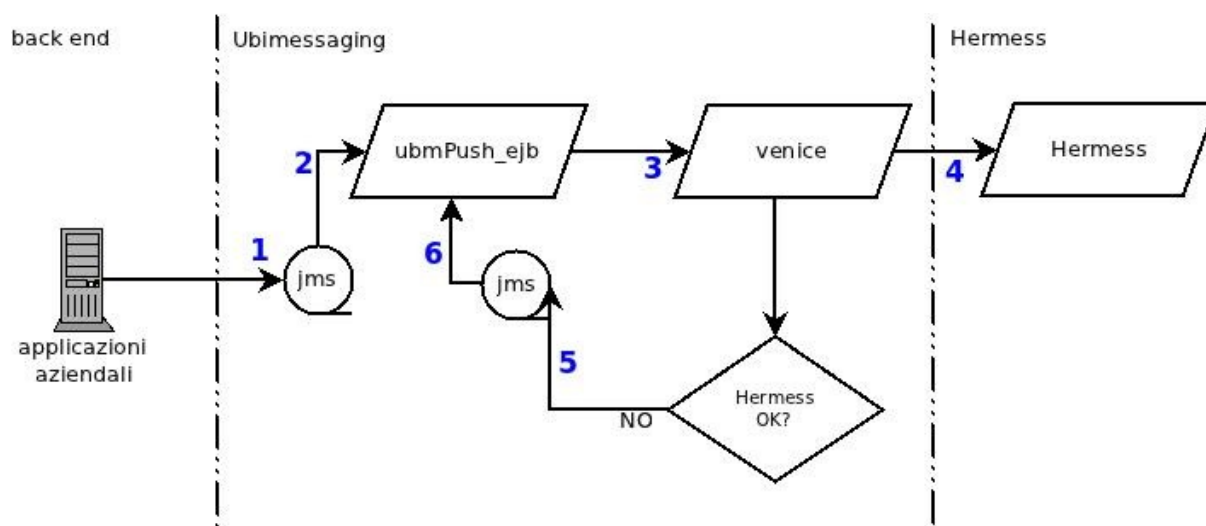
5.1 FLUSSO MESSAGGI PUSH

Nel seguente schema è rappresentato il flusso di un messaggio push. Il flusso è implementato dall'era ubmPushUnicredit

L'ear ubmPushUnicredit dell'applicazione è composto dai moduli:

- ubmPushUnicredit_ejb, che si occupa della lettura dalle code jms tramite message driven bean
- ubmPushUnicredit_war, che fornisce una interfaccia http per la comunicazione con la piattaforma

Nel disegno e nella descrizione del flusso si fa riferimento al componente venice che è una libreria di midlleware utilizzata dall'applicazione.



Punti del flusso:

1 Il messaggio viene inoltrato dalle applicazioni di back end su una coda jms

2 Il message driven bean implementato nella classe UbqSimpleJmsListenerBean scoda il messaggio. Per effettuare questa operazione una opportuna porta listener, in ascolto sulla coda, deve essere configurata su WebSphere

3 Il message driven bean implementato nella classe `UbqSimpleJmsListenerBean` chiama il servizio Venice interfaccia `PushEvento`.

Il nome del servizio da chiamare è specificato nel file `ejb-jar.xml` nella cartella `META-INF` del modulo `ejb` nella `env-entry`:

```
<env-entry>
<env-entry-name>venice/ApplicationService</env-entry-name>
<env-entry-type>java.lang.String</env-entry-type>
<env-entry-value>interfacciaPushEvento</env-entry-value>
</env-entry>
```

Il servizio Venice inserisce nella tabella `SMS_OUT` un record per il messaggio (stato: da consegnare).

4 Viene attribuito ai messaggi il billing (`idFP`, `idCat`, `mpn`, `canaleInvio`), il testo e la scadenza, utilizzando gli appositi template (descritti nel seguito). Viene effettuata la chiamata ad `Hermess`. Viene aggiornato nella tabella `SMS_OUT` lo stato del messaggio (stato: consegnato ad `Hermess`).

5 Se la chiamata ad `Hermess` va male viene gestito nel file `gestoreEsiti.xml` l'esito negativo chiamando il servizio `hiRetryHermesMR`. Questo servizio mette il messaggio nella coda `ubmRetryPush`.

6 Il message driven bean implementato nella classe `UbqSimpleJmsListenerBean` scoda il messaggio e chiama il servizio Venice corrispondente alla coda: `hiInviaRetryHermes` che effettua nuovamente l'invio ad `Hermess` (verificando prima che il messaggio non sia scaduto). Se il messaggio scade senza essere stato consegnato ad `Hermess` viene aggiornato lo stato sulla tabella `SMS_OUT` (stato: non consegnato ad `Hermess`)

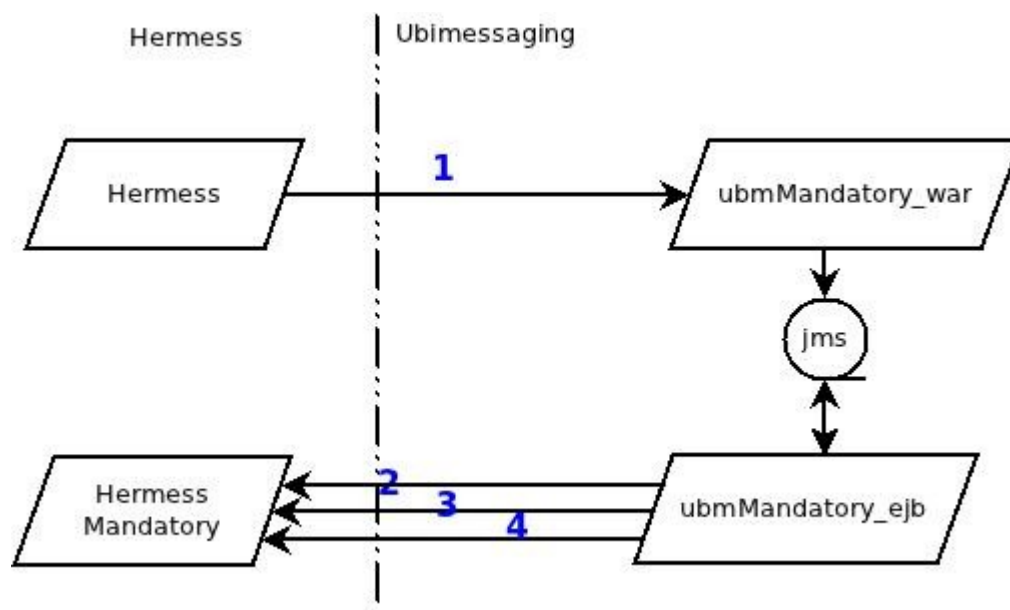
5.2 FLUSSO MESSAGGI MANDATORI

Nel seguente schema è rappresentato il flusso dei messaggi mandatori nel caso in cui il primo operatore utilizzato dal centro stella `Hermess` non sia disponibile.

Il flusso è implementato dall'ear `ubmMandatoryUnicredit`

L'ear `ubmMandatoryUnicredit` è composto dai moduli:

- `ubmMandatoryUnicredit_ejb`, che si occupa della lettura dalle code `jms` tramite message driven bean
- `ubmMandatoryUnicredit_war`, che fornisce una interfaccia `http` per la comunicazione con la piattaforma



1 Nel caso in cui Hermes non riesca a spedire all'operatore un messaggio con tag mandatory valorizzato a "true", viene chiamata la pagina `getSmsMandatory.jsp` del modulo `ubmMandatory_war` dell'applicazione `ubmMandatory`. Il messaggio è taggato con step 0 viene messo nella coda `ubmMnd` dell'applicazione `ubmMandatory`.

2 Il messaggio viene scodato dal modulo `ubmMandatory_ejb` dell'applicazione `ubmMandatory` e viene intrapreso il flusso del mandatory per inviare ad `HermessMandatory` un nuovo messaggio, con l'operatore di backup configurato nei files di configurazione del mandatory a questo step. Lo step viene incrementato di 1 rispetto a quello letto dalla coda.

Se `HermessMandatory` risponde con la presa in carico il flusso è terminato.

Se `HermessMandatory` risponde con `operatoreKO` si passa al punto 3

Se `HermessMandatory` non risponde il messaggio viene messo nella coda `ubmRetryMnd` e si passa al punto **4**

3 Il modulo `ubmMandatory_ejb` rimette il messaggio nella coda `ubmMnd` e si riprende il flusso dal punto 2.

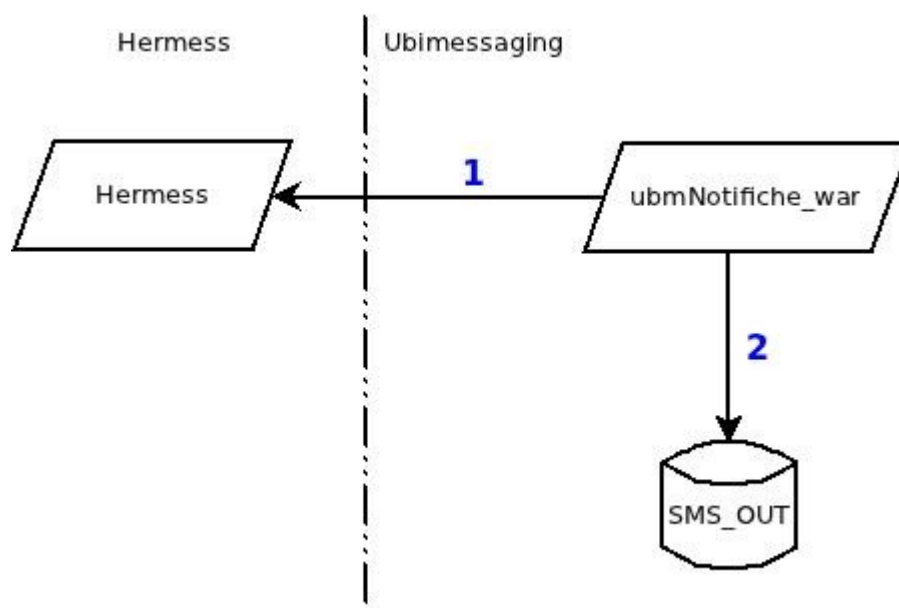
4 Il modulo `ubmMandatory_ejb` scoda il messaggio dalla coda `ubmRetryMnd` e cerca di inviarli ad `Hermess` fino alla scadenza.

5.3 FLUSSO PER IL RECUPERO DEGLI STATUS REPORT

Nel seguente schema è rappresentato il flusso per la presa in carico degli status report da Ubimessaging al centro stella Hermess.

L'ear ubmNotificheUnicredit dell'applicazione è composto dai moduli:

- *ubmNotificheUnicredit_war*, che fornisce una interfaccia http per la comunicazione con la piattaforma



1 Il modulo ubmNotifiche_war fa polling, usando un timer WebSphere, sulla jsp leggiSR.jsp dell'applicazione HermessNotifiche per prelevare uno o più status report, se presenti. Il numero di status report che vengono presi in carico è determinato da una configurazione lato Hermess. **TODO spiegazione architettura Timer**

2 Per ogni status report prelevato da Hermess viene aggiornato il corrispondente record sulla tabella SMS_OUT.

6 TEMPLATES DI CONFIGURAZIONE

6.1 CONFIGURAZIONE APPLICAZIONE UBMPUSHUNICREDIT

I templates di configurazione delle applicazioni sono posizionati esternamente all'ear in una cartella di configurazione.

Sono i seguenti:

sms_billing_push_ev.xml

template_sms_push_ev.xml

scadenze_sms_push_ev.xml

SMS_BILLING_PUSH_EV.XML

E' il file per attribuire al messaggio, in base ai valori dei campi operatore e profiloServizio ricevuto dal backend, i corretti valori per i campi idFP, idCat, canaleInvio, mnp da passare ad Hermes.

TODO: mettere esempio di file

TEMPLATE_SMS_PUSH_EV.XML

E' il file che viene utilizzato per attribuire un testo ai messaggi che arrivino dal back end con il campo tipo valorizzato a "f".

Il testo è assegnato in base al campo tipologia presente nel tracciato ricevuto dal back end. E' possibile anche cambiare il formato dei campi rispetto a come sono stati ricevuti dal back end.

Nel file i token di tipo "testo" contengono le stringhe statiche da inserire nel messaggio, i token di tipo "campo" fanno riferimento ai parametri ricevuti dal back end. Nei token di tipo "campo" è possibile specificare un formato. L'implementazione dei formati è contenuta nel file xsl mdTransformOutput.xsl che va eventualmente modificato per l'aggiunta di nuovi formati.

In questo caso il file è vuoto perchè i messaggi arriveranno già formattati.

SCADENZE_SMS_PUSH_EV.XML

E' il file che viene utilizzato per attribuire una scadenza ai messaggi, nel caso in cui non venga valorizzata nel tracciato che arriva dal back end.

Per ogni tipologia è presente la corrispondente scadenza. E' possibile anche inserire una unica scadenza per tutte le tipologie, mettendo un servizio dal nome "TUTTE" e modificando opportunamente il file xsl loAssegnaDataRitardataInput.xsl

```
<servizi>
<servizio nome="INVIO_CREDENZIALI">
  <validita anno="" mese="" giorno="" ora="24" minuto="" secondo="" millisecondi=""/>
</servizio>
</servizi>
```

6.2 CONFIGURAZIONE APPLICAZIONE UBMMNDUNICREDIT

La configurazione dell'applicazione è affidata a due file che determinano il comportamento dei messaggi:

mapping_errori.xml
mandatory_routers.xml

MAPPING_ERRORI.XML

In questo file si scrive, per ogni operatore telefonico, quale è il codice di errore attribuito. Questo codice, scritto nel tag `mappingMandatorio`, indica un percorso specificato nel secondo file di configurazione.

In questo caso, avendo due operatori bulk a disposizione, si configura che in caso di chiamata da Hermess con errore operatore_KO e operatore 6 si considera l'errore 001 allo step 1 (errorNumber=1).

```
<mappingErrori>
  <errore>
    <errorNumber>1</errorNumber>
    <origine>Hermess</origine>
    <listaCategorie>
      <categoria>Hermess</categoria>
    </listaCategorie>
    <listaDescrizioneErrori>
      <descrizioneErrore>operatore_KO</descrizioneErrore>
    </listaDescrizioneErrori>
    <listaOperatori>
      <operatore>6</operatore>
    </listaOperatori>
    <mappingMandatorio>001</mappingMandatorio>
  </errore>
</mappingErrori>
```

MANDATORY_ROUTERS.XML

In questo file si configura, per ogni errore, quali sono i nuovi valore dei vari campi da utilizzare per la chiamata ad HermessMandatory.

In questo caso per l'errore 001 (route code 001) si utilizza l'operatore 8 e tutti gli altri valori specificati.

```
<mandatoryRoutes>
  <!-- al primo giro provo a mandare come bulk su Sybase -->
  <route code="001">
    <step>
      <stepNumber>1</stepNumber>
      <sms>
        <provider>Hermess</provider>
        <operatore>8</operatore>
        <canaleInvio>bulk</canaleInvio>
        <idFP>1</idFP>
        <idCat>9</idCat>
        <mntp>go</mntp>
        <canale>PUSH</canale>
        <cellulare/>
      </sms>
    </step>
  </route>
</mandatoryRoutes>
```

7 TRACCIATI PER IL COLLOQUIO CON ALTRE APPLICAZIONI

Nei seguenti paragrafi vengono descritti i tracciati di scambio tra le applicazioni della piattaforma Ubimessaging ed il mondo esterno

7.1 TRACCIATO RICEVUTO DALLA BANCA SU CODA MQ

La banca posiziona nella coda MQ di scambio dati un tracciato XML analogo al seguente:

```
<body>
<header>
  <recapiti>
    <recapito>
      <tipoRecapito>SMS</tipoRecapito>
      <valoreRecapito>00393331234567</valoreRecapito>
      <operatore></operatore>
      <profiloServizio>bulk</profiloServizio>
    </recapito>
  </recapiti>
  <tipo>f</tipo>
  <codiceTipologia>tipologia</codiceTipologia>
  <dataScadenza>aaaammgghhmmss</dataScadenza>
```

```
<idMessaggio>id</idMessaggio>
<mandatory>true</mandatory>
<fatturazione>Banca1</fatturazione>
</header>
<footer>
  <messaggio>testo del messaggio</messaggio>
</footer>
</body>
```

I tag dell'header che hanno un valore sono riempiti seguendo la tabella:

Tag XML	Descrizione	Richiesto	Formato	Note
<tipoRecapito>	Tipo di recapito	SI	Costante, assume il valore: SMS	
<valoreRecapito>	Numero telefonico	SI	Numerico, max 25. Eventuale prefisso internazionale nella forma 00xx o 00xxx	
<operatore>	L'operatore telefonico da utilizzare	SI per premium NO per bulk	Se non è vuoto assume i valori: 1=TIM, 2=VODAFONE, 3=WIND, 4=H3G, 7=POSTEMOBILE 8=COOPVOCE	Per i messaggi bulk, gratuiti per l'utente finale, il tag può essere lasciato vuoto.
<profiloServizio>	Permette di attribuire agli utenti differenti classi di billing	SI	Assume il valore bulk	E' previsto solo l'invio di messaggi bulk.
<tipo>	Stabilisce se la formattazione è a carico di	SI	Assume i valori: uf = unformatted, f = formatted	Se vale uf sono presenti i campi con cui comporre il testo del messaggio,

	ubimessaging			altrimenti è presente il testo già composto
<codiceTipologia>	Identificativo della tipologia	SI	Stringa, max 25	Permette di estrarre report suddivisi per tipologia
<dataScadenza>	Data scadenza del messaggio	NO	Stringa, Formato: aaaammgghhm mss	Se non valorizzato può essere attribuito da ubimessaging secondo un default
<idMessaggio>	Identificativo univoco del messaggio	NO	Stringa, max 100	Se non valorizzato viene attribuito da UbiMessagging
<mandatory>	Specifica se il messaggio è mandatorio (in caso di disservizio di un operatore viene mandato attraverso altro canale)	NO	Boolean, assume i valori true e false	Se non valorizzato è false
<fatturazione>	Identificativo del soggetto cui fatturare il messaggio	SI	Stringa, max 25	

I tag del footer sono valorizzati a seconda del valore del tag tipo dell'header.
Se il messaggio arriva già formattato (tipo=f) il footer contiene il testo del messaggio già composto:

Tag XML	Descrizione	Richiesto	Formato	Note
<testo>	Il testo del messaggio	SI	Stringa, max 160 caratteri. Charset ISO 8859-1	

Se il messaggio arriva non formattato (tipo=uf) il footer contiene i parametri necessari alla composizione del messaggio della tipologia indicata.

```
<parametri>
<parametro>
<nome>
</nome>
<valore>
</valore>
</parametro>
...
</parametri>
```

7.2 TRACCIATO PER LA RICEZIONE DI UNO STATUS REPORT DA HERMESS

La seguente stringa XML viene prelevata su Hermes dall'applicazione ubmNotificheUnicredit che fa polling su un url.

```
<outputIn>
<id_msg>
<codice></codice>
<data></data>
<stato></stato>
<!-- è uno degli stati azzurri nel grafico in Illustrazione 1-->
<codOperatore></codOperatore>
<descrErrore></descrErrore>
<tipoNotifica></tipoNotifica><!--vale "P" per notifiche transitorie o perziali e "C" per
notifiche conclusive
<cliente></cliente>
<operatore>1</operatore>
<!-- è l'operatore con cui è estato effettivamente inviato il messaggio -->
<operatoreMNP>2</operatoreMNP>
<!-- è l'operatore indicato dal db di MNP -->
<applicazione>UbiMessaging</applicazione>
<chiave></chiave>
</id_msg>
</outputIn>
```

7.3 TRACCIATO PER L'INVIO DI UN MESSAGGIO MT AD HERMESS

Il tracciato xml inviato ad Hermess per la spedizione del messaggio da parte dell'applicazione ubmPushUnicredit è il seguente.

Il camp subCli viene valorizzato in base al valore del campo fatturazione ricevuto dal back end.

```
<messaggio>
  <operatoreBE></operatoreBE><!-- operatore ricevuto dal back end -->
  <numtel></numtel>
  <idFP></idFP><!-- per la fascia di prezzo -->
  <idCat></idCat><!-- per la categoria del messaggio (caring, attivazione...) -->
  <mnp></mnp><!-- go/stop/forward per l'uso della MNP -->
  <servizio></servizio>
  <canale></canale><!--push/pull-->
  <testo></testo>
  <dataScadenza></dataScadenza><!-- formato aaaammgghhmmss -->
  <codice></codice><!-- è il campo CODICE_HERMESS sulla sms_out, non
corrisponde al codice eventualmente ricevuto dal back end.--->
  <applicazione></applicazione>
  <canaleInvio></canaleInvio><!--bulk/premium-->
  <mandatory></mandatory><!-- facoltativo -->
  <command></command><!-- facoltativo -->
  <alias></alias><!-- facoltativo -->
  <subCli></subCli><!-- facoltativo -->
</messaggio>
```

7.4 RICEZIONE DA HERMESS DI UN MESSAGGIO MANDATORIO NON SPEDITO

Questo è un esempio del tracciato ricevuto da Hermess sulla jsp getSmsMandatory.jsp dell'applicazione ubmMandatoryUnicredit. Inizialmente l'attributo stepN ha valore 0. Ai passaggi successivi il messaggio viene messo da ubmMandatory con il valore dell'attributo incrementato di uno.

```
<messaggio applicazione="UbiMessaging" stepN="0" msgEsito="operatore_KO"
codEsito="Hermess" esito="0">
  <operatore>3</operatore>
  <numtel>3937199828</numtel>
  <idFP>2</idFP>
  <idCat>5</idCat>
  <servizio>TEST</servizio>
  <canale>PUSH</canale>
```

```
<testo>prova1</testo>
<dataScadenza>"+data+"</dataScadenza>
<codice>"+codice+"</codice>
<applicazione>Ubimessaging</applicazione>
<canaleInvio>bulk</canaleInvio>
<alias/>
<command>0</command>
<mandatory>true</mandatory>
<cliente>Fineco</cliente>
</messaggio>
```

7.5 INVIO AD HERMESS DI UN MESSAGGIO DA SPEDIRE SULL'OPERATORE DI BACK UP

Il tracciato xml inviato ad Hermess per la spedizione del messaggio da parte dell'applicazione ubmMndUnicredit sul canale di back up:

```
<messaggio>
  <operatore>6</operatore>
  <numtel>3331234567</numtel>
  <idFP>1</idFP>
  <idCat>9</idCat>
  <mnt>go</mnt>
  <servizio>TEST</servizio>
  <canale>PUSH</canale>
  <testo>prova1</testo>
  <dataScadenza>20111219145514967</dataScadenza>
  <codice>20110617173401111_10397_M1</codice>
  <applicazione>Ubimessaging</applicazione>
  <canaleInvio>bulk</canaleInvio>
  <alias/>
  <mandatory>true</mandatory>
</messaggio>
```

7.6 RISPOSTA DI HERMESS NEL CASO DI MANCATA SPEDIZIONE DEL MESSAGGIO SUL CANALE DI BACK UP

Sotto la risposta sincrona ricevuta da ubmMndUnicredit nella chiamata ad Hermess nel caso in cui l'operatore verso cui si invia il messaggio manadatorio non è disponibile (l'esempio si riferisce al primo step del flusso):

```
<bearer-result esito="0" codEsito="0" msgEsito="">
```

```
<messaggio esito="0" codEsito="Hermess" msgEsito="operatore_KO" stepN="1"
applicazione="UbiMessaging">
<operatore>1</operatore>
<numtel>3402517812</numtel>
<idFP>1</idFP>
<idCat>9</idCat>
<servizio>disposizioni_on_line_free</servizio>
<canale>PUSH</canale>
<testo>testo del messaggio</testo>
<dataScadenza>20111219145514967</dataScadenza>
<codice>1286268715848178000_M1</codice>
<applicazione>UbiMessaging</applicazione>
<cliente>BPM</cliente>
<esitoUb/>
<codEsitoUb/>
<sender>40717</sender>
<esito>valido</esito>
<alias/>
<mandatory>true</mandatory>
</messaggio>
</bearer-result>
```