



Consiglio Nazionale delle Ricerche

I Controlli Periodici del Sistema Sincrono di Gestione dei Nomi a Dominio del Registro Italiano

M. Loffredo, M. Martinelli

IIT B4-05/2016

Nota Interna

Maggio 2016



Istituto di Informatica e Telematica

**I Controlli Periodici
del Sistema Sincrono di Gestione
dei Nomi a Dominio del Registro Italiano**

Loffredo M., Martinelli M.

Sommario

Abstract	1
1. Introduzione	1
2. Tecnologie di supporto	2
2.1 Message Driven Bean	2
2.2 Espressioni per la schedulazione dei job di Quartz	4
2.3 Java Message Service e Apache ActiveMQ.....	6
3. Coda di polling e notifiche via mail.....	6
4. I controlli periodici.....	7
4.1 Controllo a seguito della verifica della configurazione DNS.....	7
4.2 Controlli a seguito della scadenza di un periodo temporale.....	8
4.2.1 Controlli per le transizioni di stato	10
4.2.2 Controlli per il calcolo delle varie soglie di credito per i Registrar	15
4.2.3 Controlli per la cancellazione definitiva dei domini	16
4.2.4 Controlli per il re-invio delle richieste di verifica della configurazione DNS	16
4.2.5 Controlli per il recupero di domini nello stato inactive/notRenewed	17
4.2.6 Controlli per la verifica di consistenza su domini dopo operazioni del Registro.....	17
4.2.7 Controllo per la cancellazione dei messaggi obsoleti.....	17
4.2.8 Controllo per rimozione dei contatti non più utilizzati	18
4.2.9 Controllo per azzerare il limite dei comandi EPP	18
5. Conclusioni	18
Bibliografia.....	20

Abstract

Un sistema sincrono per la gestione dei nomi a dominio ha come sua caratteristica peculiare quella di mettere a disposizione degli operatori che si occupano della registrazione e manutenzione dei nomi a dominio, detti Registrar, per conto degli assegnatari, detti Registrant, un'applicazione in grado di elaborare in tempo reale le operazioni richieste riguardanti gli oggetti coinvolti (nomi a dominio, contatti, name server). Anche il Registro Italiano dei nomi a dominio nella sua transizione dalla gestione asincrona, fondata essenzialmente sulla elaborazione manuale di fax con fatturazione posticipata delle operazioni, alla gestione sincrona, caratterizzato dall'addebito anticipato, si è dotato di un server basato sul protocollo standard EPP⁽¹⁾. Tuttavia, accanto al server EPP, è stato necessario implementare una serie di procedure "asincrone" dedicate ad un insieme di aspetti coinvolti nella gestione sia dei nomi a dominio sia dei Registrar. Queste procedure erano già presenti nel sistema asincrono ma il passaggio al sistema sincrono con la conseguente introduzione del concetto di pre-pagato e la velocizzazione dei processi di registrazione e manutenzione ha comportato un generale ripensamento delle stesse e alla loro implementazione tramite nuove tecnologie. In questo report, sono illustrate una parte di tali procedure asincrone, la cui caratteristica comune è quella di attivarsi senza il ricorso all'interazione con un operatore. Sono illustrate anche le tecnologie di supporto all'implementazione delle suddette procedure in ambiente Java EE.

1. Introduzione

Un sistema sincrono di gestione dei nomi a dominio (di seguito semplicemente **sistema**) di un Top Level Domain deve prevedere, accanto ad un server che gestisca le richieste per eseguire operazioni in tempo reale sui vari oggetti coinvolti nella gestione di nomi a dominio, anche un serie di procedure rivolte all'esecuzione di un insieme di operazioni off-line.

Il Registro Italiano dei nomi a dominio (di seguito **Registro**), ovviamente, non fa eccezione a questo schema per cui, accanto al server basato sul protocollo EPP⁽²⁾, dispone anche di una serie di procedure asincrone che hanno la finalità di implementare:

1. Operazioni che non si esauriscono con l'invio di una richiesta al server EPP il cui completamento con successo o fallimento avviene dopo un certo lasso temporale (ad esempio il completamento di un'operazione di trasferimento o di un'operazione di modifica della configurazione DNS)
2. Operazioni conseguenti all'accadimento di precisi eventi temporali (ad esempio la scadenza di un nome a dominio o la scadenza del periodo massimo in cui un nome a dominio può restare in un determinato stato)
3. Operazioni la cui finalità è principalmente la notifica a Registrant e Registrar di alcuni fatti di primaria importanza per la gestione dei nomi a dominio (ad esempio la notifica ai Registrar dell'ammontare del proprio credito in relazione ad un certa soglia minima)
4. Operazioni necessarie a mantenere l'integrità e l'efficienza della base di dati di supporto al sistema (ad esempio la rimozione di contatti non utilizzati da molto tempo)
5. Operazioni che riguardano la gestione di un nome a dominio da un punto di vista amministrativo/legale e che sono necessariamente dipendenti dal regolamento del Registro (ad esempio la contestazione di un nome a dominio o la sua revoca)
6. Operazioni messe a disposizione dei Registrar le quali non possono essere implementate nel protocollo EPP perché necessitano di una verifica umana (ad esempio il trasferimento di un numero

massiccio di domini anche detto bulk transfer o la cessione del contratto Registrar da una società ad un'altra con conseguente migrazione dei domini e dei contatti)

7. Operazioni attualmente non implementate dal protocollo EPP perché considerate troppo onerose dal punto di vista delle risorse necessarie al loro espletamento in tempo reale (ad esempio le ricerche che restituiscono un set consistente di dati)
8. Operazioni dipendenti dall'interazione con altri sistemi esterni al Registro (come ad esempio il versamento di un certo ammontare sul credito di un Registrar attraverso bonifico o carta di credito oppure la fatturazione delle operazioni dipendente dall'interazione con il Sistema Informativo per la Gestione delle Linee di Attività dell'Amministrazione Centrale del C.N.R. anche detto SIGLA)

In questo documento vengono descritte una parte di queste procedure, appartenenti principalmente alle prime quattro categorie, note all'interno del sistema come controlli periodici (di seguito **controlli**). Le operazioni riguardanti le restanti categorie sono implementate tramite altri strumenti messi a disposizione del Registrar (portale RAIN e nella nuova versione RAIN-NG) oppure strumenti messi a disposizione degli operatori del Registro (portali MIDA e jBilling).

2. Tecnologie di supporto

2.1 Message Driven Bean

L'ambiente di riferimento per lo sviluppo di tutte le applicazioni del Registro è costituito dal linguaggio Java e da tutte le tecnologie proprie di questo linguaggio dedicate allo sviluppo di applicazioni distribuite che compongono la piattaforma **Java 2 Enterprise Edition (J2EE)**.

I modelli di applicazioni distribuite più utilizzati e supportati dalla piattaforma J2EE sono:

- Applicazioni browser-based, in cui il lato server è implementato con Java Server Pages (JSP) e/o Servlet (come il server EPP)
- Applicazioni basate sull'invocazione delle funzionalità messe a disposizione da un componente server da parte di un componente client (Java Remote Method Invocation o RMI)
- Applicazioni basate sulla gestione di eventi
- Applicazioni service-based che offrono servizi tramite i protocolli SOAP/HTTP (Web Services)

Gli Enterprise Java Beans (EJB)⁽³⁾ sono le componenti più importanti della piattaforma J2EE; rappresentano una vera e propria tecnologia per l'implementazione della logica di business di un sistema distribuito.

Il secondo modello di applicazione distribuita è implementato tramite un EJB chiamato Session Bean il quale rappresenta logicamente una sessione di operazioni di cui un'applicazione client o un altro bean può aver bisogno; praticamente è una classe che implementa una logica di business in un'applicazione server-side e i cui metodi possono essere richiamati in locale o da remoto.

Il terzo modello è implementato tramite un EJB chiamato Message Driver Bean (MDB) che è molto simile al Session Bean con la differenza che non ha metodi che vengono invocati da un'applicazione client, ma vengono attivati in corrispondenza del verificarsi di eventi.

Per l'implementazione dei controlli asincroni del Registro sono stati presi in considerazione due tipologie di attivazione:

1. In conseguenza della ricezione di un messaggio in una coda gestita tramite un middleware che implementa la specifica Java Message Service (JMS)⁽⁴⁾ come ad esempio ActiveMQ⁽⁵⁾
2. In conseguenza della gestione di un evento temporale innescato da uno schedatore di Job come ad esempio Quartz⁽⁶⁾

Nel primo caso, la classe che contiene il codice da eseguire al momento della ricezione del messaggio deve essere opportunamente annotata e deve implementare l'interfaccia `javax.jms.MessageListener` la quale richiede che sia implementato il metodo `onMessage` (Fig.1)

```
@MessageDriven(activationConfig =
{
    @ActivationConfigProperty(propertyName="destinationType", propertyValue="javax.jms.Queue"),
    @ActivationConfigProperty(propertyName="destination", propertyValue="myqueue")
})
@TransactionManagement(TransactionManagementType.BEAN)
@ResourceAdapter(value="activemq-ra.rar")
public class MyQueueConsumer implements MessageListener {

    @Override
    public void onMessage(Message message) {

    }
}
```

Fig.1 – Attivazione di un MDB tramite ricezione di un messaggio su una coda

Nel secondo caso, invece la classe che contiene il codice, anch'essa opportunamente annotata, deve implementare l'interfaccia `org.quartz.Job` la quale richiede che sia implementato il metodo `execute` (Fig.2)

```
@MessageDriven(activationConfig =
{
    @ActivationConfigProperty(propertyName = "cronTrigger", propertyValue = "0/2 * * * * ?")
})
@ResourceAdapter("quartz-ra.rar")
public class AnnotatedQuartzMDBBean implements Job
{
    public void execute(JobExecutionContext jobExecutionContext) throws JobExecutionException {
    }
}
```

Fig.2 – Attivazione di un MDB tramite un job di Quartz con uso di annotazioni

Invece di utilizzare l'annotazione, è possibile configurare l'MDB per l'integrazione con Quartz attraverso un descrittore specificato nel file `META-INF/ejb-jar.xml` (Fig.3) e attraverso un resource adapter specificato nel file `META-INF/jboss.xml` (Fig.4)

```

<message-driven>
  <ejb-name>ExampleMDB</ejb-name>
  <ejb-class>org.jboss.tutorial.quartz.bean.QuartzMDBBean</ejb-class>
  <transaction-type>Container</transaction-type>
  <activation-config>
    <activation-config-property>
      <activation-config-property-name>cronTrigger</activation-config-property-name>
      <activation-config-property-value>0 15 7 * * ?</activation-config-property-value>
    </activation-config-property>
  </activation-config>
</message-driven>

```

Fig.3 – Attivazione di un MDB tramite job di Quartz con uso di descrittore nel file ejb-jar.xml

```

<message-driven>
  <ejb-name>ExampleMDB</ejb-name>
  <resource-adapter-name>quartz-ra.rar</resource-adapter-name>
</message-driven>

```

Fig.4 – Specifica del resource adapter per Quartz nel file jboss.xml

Infine, un'altra possibilità è utilizzare, al posto di Quartz, l'EJB Timer Service. In questo caso è sufficiente utilizzare una semplice annotazione (Fig.5)

```

@Stateless
public class MyBean {

    @Schedule(second="0", minute="15", hour = "7", persistent=false, info="description")
    private void mymethod(Timer timer) throws Exception {

    }

}

```

Fig.5 – Attivazione di un MDB tramite EJB Timer Service

2.2 Espressioni per la schedulazione dei job di Quartz

Le espressioni per la schedulazione dei job di Quartz o di EJB Timer Service, anche dette “cron expressions”, sono utilizzate per configurare le istanze di “cron trigger” che hanno il compito di innescare i vari job.

Tali espressioni sono stringhe composte di 7 campi che descrivono i dettagli della schedulazione:

1. Secondi
2. Minuti
3. Ore
4. Giorno-del-mese
5. Mese
6. Giorno-della-settimana
7. Anno (opzionale)

Un esempio di cron expression è la stringa “0 0 12 ? * WED” che significa “ogni Mercoledì alle 12:00:00”

Ogni campo può contenere intervalli e/o liste. Ad esempio, il campo “Giorno-della-settimana” potrebbe avere i seguenti valori:

1. "MON-FRI" che significa "da Lunedì a Venerdì"
2. "MON,WED,FRI" che significa "il Lunedì, il Mercoledì, il Venerdì"
3. "MON-WED,SAT" che significa "da Lunedì a Mercoledì ed il Sabato"

Il carattere "*" può essere utilizzato per indicare tutti i possibili valori di un campo. Quindi, ad esempio il carattere "*" nella cron expression precedente significa "ogni mese".

Tutti i campi hanno un insieme di valori ammissibili:

1. 0-59 per i campi Secondi e Minuti
2. 0-23 per il campo Ore
3. 1-31 per il campo Giorno-del-mese
4. 0-11 oppure JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV e DEC per il campo Mese
5. 1-7 (dove 1 = Sunday) oppure SUN, MON, TUE, WED, THU, FRI e SAT per il campo Giorno-della-settimana

Il carattere "/" può essere utilizzato per specificare incrementi nei valori che un campo può assumere. Ad esempio, se specifichiamo "0/15" nel campo Minuti significa "ogni 15 minuti dell'ora a partire dal minuto 0" il quale ha lo stesso significato di "0,15,30,45". Da notare che "0/35" non significa ogni 35 minuti ma "ogni 35 minuti dell'ora a partire dal minuto 0" cioè ai minuti 0 e 35.

Il carattere "?" è consentito per i campi Giorno-del-mese e Giorno-della-settimana ed ha il significato di "nessun valore specifico". Ciò è utile quando dobbiamo specificare uno dei due campi ma non l'altro.

Anche il carattere "L" è consentito per i campi Giorno-del-mese e Giorno-della-settimana ed ha il significato di "last" (ultimo) ma ha un significato differente per ciascuno dei due campi:

- Nel campo Giorno-del-mese significa "l'ultimo Giorno-del-mese" (il giorno 31 per Gennaio, il giorno 28 per Febbraio se l'anno non è bisestile, etc.). E' possibile anche specificare un offset dall'ultimo Giorno-del-mese. Ad esempio, "L-3" significa "il terzo giorno precedente l'ultimo Giorno-del-mese".
- Nel campo Giorno-della-settimana significa semplicemente "Sabato" (che ha lo stesso significato di "7" o "SAT"). Ma se viene utilizzato in combinazione con un altro valore significa "l'ultimo xxx del mese". Ad esempio, "6L" o "FRIL" significa "l'ultimo Venerdì del mese".

Quando si utilizza l'opzione "L", è necessario non specificare liste e o intervalli di valori perché il risultato potrebbe essere inaspettato.

Il carattere "W" è utilizzato per specificare "il Giorno-della-settimana feriale più vicino ad un dato giorno". Ad esempio, se scriviamo "15W" significa "il giorno feriale più vicino al 15 del mese".

Il carattere "#" è utilizzato per specificare "l'n-esimo xxx giorno feriale del mese". Ad esempio, il valore "6#3" o "FRI#3" nel campo Giorno-della-settimana significa "il terzo Venerdì del mese".

Nella tabella di seguito sono riportate alcuni esempi di cron expressions.

Espressione	Significato
0 0/5 * * * ?	Ogni 5 minuti
10 0/5 * * * ?	Ogni 5 minuti al secondo 10 cioè alla 10:00:10, 10:05:10, etc.
0 30 10-13 ? * WED,FRI	Alle 10:30, 11:30, 12:30 e 13:30 di ogni Mercoledì e Venerdì
0 0/30 8-9 5,20 * ?	Ogni mezz'ora tra le ore 8 (compreso) e le 10 (escluso) nel 5° e 20° Giorno-del-mese

2.3 Java Message Service e Apache ActiveMQ

Un sistema di messaggistica è fondamentale per far colloquiare in modo asincrono i diversi agenti software di un'architettura complessa come quella di un sistema di gestione dei nomi a dominio.

La specifica J2EE che definisce un insieme di API le quali consentono ad applicazioni Java di scambiarsi messaggi fra di loro è denominata **Java Message Service (JMS)**. Tale specifica prevede i seguenti componenti:

1. **JMS provider o broker:** l'implementazione della specifica JMS per un particolare Message Oriented Middleware (MOM)
2. **JMS client:** un'applicazione o un processo che invia e/o riceve messaggi
3. **JMS producer:** un JMS client che crea ed invia messaggi
4. **JMS consumer:** un JMS client che riceve i messaggi
5. **JMS queue:** una coda che raccoglie i messaggi inviati da un mittente che sono in attesa di essere letti da un destinatario. I messaggi vengono consegnati nell'ordine in cui vengono inviati ed una volta letti vengono rimossi dalla coda. E' possibile inoltre impostare un tempo limite entro il quale i messaggi presenti in una coda devono essere letti prima di essere rimossi (time-to-leave). Questo modello di gestione dei messaggi è chiamato **Point-To-Point**.
6. **JMS topic:** una sorta di lista di distribuzione di messaggi inviati da un mittente (publisher) a più destinatari (subscribers) ognuno dei quali si è iscritto alla lista di distribuzione. Un subscriber che si è iscritto ad un topic potrà consumare solamente messaggi pubblicati dopo la sua iscrizione. Questo modello di gestione dei messaggi è detto **Publish/Subscribe**.

Apache ActiveMQ è il più popolare MOM open source che supporta la specifica JMS. La caratteristica più rilevante di un MOM consiste dunque nella possibilità di instaurare una comunicazione asincrona fra mittente e ricevente i quali, per comunicare fra loro, non devono necessariamente essere attivi nello stesso istante.

Fra le caratteristiche salienti di ActiveMQ si possono annoverare:

1. La possibilità di utilizzare vari protocolli di trasporto: VM, TCP, UDP, SSL, AMQP, MQTT, HTTP/HTTPS, etc. Nell'implementazione del Registro è stato utilizzato il protocollo TCP. Non c'è necessità di utilizzare una comunicazione sicura dal momento che non esistono comunicazioni che avvengono al di fuori della sottorete del Registro.
2. La possibilità di instaurare un livello di trasporto cosiddetto **failover** al di sopra del livello fisico di trasporto la cui finalità può essere:
 - a. Assicurare un broker di backup secondario qualora i client fossero impossibilitati a connettersi al broker primario
 - b. Bilanciare il carico di interazioni con i client fra una serie di brokerNell'implementazione del Registro il livello failover è stato utilizzato con la prima finalità.
3. La possibilità di supportare vari protocolli per la specifica dei messaggi scambiati (OpenWire, Stomp, AMQP, MQTT, XMPP, etc.). Nell'implementazione del Registro è stato utilizzato il protocollo OpenWire che è anche quello di default utilizzato dai JMS client in linguaggio Java.

3. Coda di polling e notifiche via mail

La coda di polling è il meccanismo del protocollo EPP per fare in modo che un Registro possa informare i suoi Registrar riguardo:

- Il completamento con successo o fallimento di ogni azione pendente innescata da una richiesta pervenuta da un Registrar tramite un client EPP
- Il completamento di ogni altra azione compiuta dal Registro che non è in risposta ad una richiesta pervenuta da un Registrar tramite un client EPP

La maggior parte dei controlli hanno come effetto secondario la generazione di un messaggio nella coda di polling di un Registrar.

La generazione del messaggio nella coda di polling è a volte accompagnato dall'invio di una mail nelle circostanze in cui deve essere eseguita una notifica diretta ad un attore che non può accedere alla coda di polling come nel caso di una notifica ad un Registrant o nel caso di notifiche riguardanti la soglia di credito di un Registrar che interessano prevalentemente le figure amministrative piuttosto che tecniche.

4. I controlli periodici

Come è facile immaginare dalla tipologie dei vari MDB utilizzati e illustrati precedentemente, i controlli periodici del sistema possono essere suddivisi in due categorie:

1. Controlli che vengono attivati a seguito della ricezione di messaggi inseriti in specifiche code
2. Controlli che vengono attivati a precise istanze temporali che possono corrispondere:
 - a. Alle scadenze dei vari stati di un dominio
 - b. Alle scadenze di altri intervalli di tempo

4.1 Controllo a seguito della verifica della configurazione DNS

Alla prima categoria appartiene il solo controllo eseguito all'atto della ricezione sulla coda **dns.validation.responses** di un messaggio riguardante l'avvenuta verifica con successo o con fallimento della configurazione DNS associata ad un dominio.

Una nuova configurazione DNS di un dominio viene verificata in due occasioni:

1. Al momento della creazione o riassegnazione di un nuovo nome a dominio cioè quando il dominio è nello stato inactive/dnsHold e non esiste alcuna configurazione DNS precedentemente validata con successo
2. Al momento della richiesta di cambio di configurazione DNS per un dominio che è già delegato cioè quando il dominio è nello stato pendingUpdate ed è quindi già presente una configurazione DNS validata con successo

Il controllo esegue in entrambi i casi pressoché le stesse azioni. Infatti:

1. In caso di verifica terminata con fallimento, non viene eseguita alcuna transizione di stato e, qualora la risposta sia relativa alla prima verifica effettuata, viene inserito nella coda di polling del Registrar un messaggio contenente il report dettagliato dell'errore riscontrato
2. In caso di verifica terminata con successo, si esegue una transizione verso lo stato ok (compatibilmente con le varie proibizioni imposte sul dominio e/o con la presenza degli stati autoRenewPeriod e/o challenged) e si aggiorna la configurazione DNS associata al dominio. Tale

nuova configurazione DNS sarà resa operativa nella zona .it con il successivo turno di generazione della zona stessa. Nel caso in cui il dominio è nello stato inactive/dnsHold viene anche inviata una mail al Registrant per informarlo che il dominio risulta delegato.

4.2 Controlli a seguito della scadenza di un periodo temporale

Nell'implementazione del Registro sono presenti vari controlli che vengono innescati da cron trigger corrispondenti a precisi cron expressions.

I controlli sono divise **funzionalmente** in:

1. Controlli per effettuare le varie transizioni di stato dovute alla scadenza degli stati dei domini
2. Controlli per il calcolo delle varie soglie di credito per i Registrar
3. Controlli per la cancellazione definitiva dei domini contestualmente al drop time
4. Controlli per il re-invio periodico delle richieste di verifica della configurazione DNS che hanno avuto esito negativo
5. Controlli per il recupero di domini messi nello stato inactive/notRenewed perché il Registrar non disponeva di credito sufficiente ad assicurare il loro rinnovo
6. Controlli per la verifica di consistenza su domini dopo operazioni del Registro
7. Controllo per la cancellazione dei messaggi obsoleti
8. Controllo per rimozione dei contatti non più utilizzati
9. Controllo per azzerare il limite dei comandi inviati dai vari Registrar ai due server EPP (epp.nic.it ed epp-deleted.nic.it)

Per ciascuno di questi gruppi funzionali verranno identificati più avanti nel documento i vari **gruppi temporali** corrispondenti a precise cron expression.

L'attivazione delle vari controlli non è fatta direttamente ma è intermediata da un file di configurazione che per ogni gruppo di controlli e per ciascun singolo controllo riporta due parametri:

1. **active**: esprime il fatto che un gruppo di controlli o un singolo controllo nonostante siano stati attivati da un cron trigger debbano essere veramente eseguiti oppure no
2. **ignorePeriods**: esprime il fatto che per un gruppo di controlli o un singolo controllo che dipendono da una scadenza temporale (ad esempio la scadenza di uno stato) si debba realmente considerare tale scadenza o no. Ad esempio: nella versione di questi controlli per l'ambiente di test pubblico, si deve prevedere che lo stato redemptionPeriod non scada dopo 30 giorni dall'invio della richiesta EPP Delete Domain ma alla fine del giorno stesso in cui la richiesta è stata inviata. Ciò per velocizzare, ai fini di test, la creazione di domini, la loro rimozione e successiva ri-registrazione.

I valori dei parametri applicati alla singolo controllo di un gruppo sovrascrivono quelli applicati al gruppo. I parametri active e ignorePeriods devono essere sempre presenti nella specifica del gruppo mentre, nella specifica della singolo controllo, deve essere presente active ma può mancare ignorePeriods. In questo caso si assume che il valore di ignorePeriods è false.

Il file di configurazione descrive anche l'ordine con cui i controlli appartenenti allo stesso gruppo vengono sequenzialmente chiamati.

Di seguito è riportato un estratto del file di configurazione per il gruppo corrispondente alle transizioni di stato (Fig. 6)

Vediamo adesso nel dettaglio le funzionalità di ciascun gruppo di controlli.

4.2.1 Controlli per le transizioni di stato

Cron expression: 0 30 0 * * ?

Il gruppo di controlli che riguarda le transizioni di stato viene attivato alle ore 0.30 di ogni giorno e si occupa di effettuare le transizioni di stato per tutti i domini per i quali risulta superato il periodo di permanenza in un certo stato.

A questo gruppo appartiene anche il controllo che effettua la transizione di stato per tutti i domini che sono scaduti.

Nella tabella di seguito sono riportati i periodi temporali associati agli stati che vengono presi in considerazione dai controlli di questo gruppo.

Lo stato pendingTransfer si intende scaduto alla fine del giorno in cui è iniziato il trasferimento.

Periodo	Descrizione	Unità	Valore
DOMAIN_EXPIRE	Scadenza di un dominio	anno	1
PENDING_UPDATE_PERIOD	Periodo massimo di permanenza nello stato pendingUpdate	giorni	5
REDEMPTION_PERIOD	Periodo massimo di permanenza nello stato pendingDelete/redemptionPeriod	giorni	30
PENDING_TRANSFER_PERIOD	Periodo massimo di permanenza nello stato pendingTransfer	giorni	0
AUTO_RENEW_PERIOD	Periodo di grace dopo il rinnovo automatico del nome a dominio	giorni	15
REVOKED_PERIOD	Periodo massimo di permanenza nello stato inactive/revoked	giorni	30
TO_BE_REASSIGNED_PERIOD	Periodo massimo di permanenza nello stato inactive/toBeReassigned	giorni	30
NO_REGISTRAR_PERIOD	Periodo massimo di permanenza nello stato inactive/noRegistrar	giorni	60
NOT_RENEWED_PERIOD	Periodo massimo di permanenza nello stato inactive/notRenewed	giorni	30

Una transizione di stato in generale prevede che, a seguito di un'operazione richiesta dal Registrar o a seguito di un'operazione richiesta dal Registro o all'accadimento di un preciso evento, il dominio transiti da uno stato di partenza ad uno stato di arrivo e che contestualmente a tale transizione possano essere eseguite delle azioni. Le azioni più comuni sono:

1. l'inserimento di un messaggio nella coda di polling del Registrar che gestisce il dominio
2. l'invio di una mail al Registrant del dominio

Lo stato di partenza non è l'unica cosa che condiziona la scelta dello stato di arrivo. Altre condizioni che concorrono a determinare lo stato di arrivo possono essere:

1. Lo stato che il dominio aveva precedentemente allo stato di partenza
2. La presenza di altri stati associati a quello di partenza come lo stato autoRenewPeriod o challenged

3. La presenza di proibizioni imposte dal Registrar o dal Registro sul dominio
4. Il fatto che il dominio sia scaduto o no

In base a quanto detto, è stata tentata una rappresentazione generalizzata delle transizioni di stato a seguito di operazioni EPP ed eventi non EPP che è riportata in un file di configurazione con il seguente contenuto:

1. Una prima parte utilizzata solo dai server EPP in cui sono riportate, per ogni stato, la lista delle operazioni permesse e/o proibite con l'eventuale errore da generare nel caso in cui un'operazione sia stata richiesta in uno stato nel quale non è consentita
2. Una seconda parte in cui per ogni stato di partenza sono elencate i possibili stati di arrivo, in base alle altre condizioni presenti, e le eventuali azioni da intraprendere delle due tipologie sopra descritte

Di seguito (Fig. 8) è riportato un estratto del file di configurazione delle transizioni di stato in cui sono evidenziate rispettivamente l'insieme de comandi EPP ammessi nello stato pendingTransfer e le due transizioni dallo stato pendingTransfer (senza altri stati o proibizioni aggiuntive) agli stati ok ed inactive/dnsHold in base al fatto che il dominio ha associato correntemente dei name server o no. Anche questo file di configurazione è caricato nell'applicazione tramite la tecnologia XStream.

```

<state-properties>
  <state-commands>
    . . . . .
    <state-command>
      <state>pendingTransfer</state>
      <commands type="allowed">
        <command>DOMAIN_TRANSFER_CANCEL</command>
        <command>DOMAIN_TRANSFER_APPROVE</command>
        <command>DOMAIN_TRANSFER_REJECT</command>
        <command>DOMAIN_TRANSFER_TRADE_CANCEL</command>
        <command>DOMAIN_TRANSFER_TRADE_APPROVE</command>
        <command>DOMAIN_TRANSFER_TRADE_REJECT</command>
      </commands>
      <error-reason>DOMAIN_STATE_PENDING_TRANSFER</error-reason>
    </state-command>
    . . . . .
  <state-commands>
  . . . . .
  <state-transitions>
    . . . . .
    <state-transition>
      <from-state>pendingTransfer</from-state>
      <conditioned-state-group>
        <condition name="DomainHasNoNameServers" expectedValue="false"/>
        <to-state>ok</to-state>
      </conditioned-state-group>
      <conditioned-state-group>
        <condition name="DomainHasNoNameServers" expectedValue="true"/>
        <to-state>inactive</to-state>
        <to-state>dnsHold</to-state>
      </conditioned-state-group>
      <event>PENDING_TRANSFER_EXPIRED</event>
      <actor>registry</actor>
      <polling-msg-to-Registrar>DOMAIN_TRANSFER_EXPIRED</polling-msg-to-
Registrar>
    </state-transition>
    . . . . .
  </state-transitions>
  . . . . .
</state-properties>

```

Fig.8 – Estratto del file di configurazione sulle transizioni di stato: esempio di operazioni permesse e transizioni di stato per lo stato pendingTransfer

Altre azioni sono legate ai concetti di addebito e fatturazione di un'operazione:

1. **Addebito**, si intende l'azione di prelevare, dal credito del Registrar, il costo di un'operazione comprensivo dell'eventuale I.V.A. Al fine di tenere costantemente aggiornato l'ammontare del credito del Registrar, le operazioni fatturabili vengono immediatamente addebitate. Le operazioni fatturabili sono:
 - a. *Registrazione*
 - b. *Richiesta di trasferimento* (anche detta cambio Registrar)
 - c. *Recupero successivo a richiesta di cancellazione*
 - d. *Rinnovo automatico*

Le prime tre sono conseguenti ad operazioni eseguite dal Registrar tramite i server EPP, mentre la quarta è eseguita dal Registro.

2. **Fatturazione**, si intende l'azione di rendere disponibile il suddetto costo in modo da essere riportato in una successiva fattura da inviare al Registrar. La fattura contiene tutte le operazioni effettuate dal Registrar a fronte di un determinato versamento.

Questa differenziazione si è resa necessaria dal momento in cui la fatturazione, rispetto all'addebito, può avvenire in modo non contestuale. Nei casi di rinnovo automatico o trasferimento, per esempio, addebito e fatturazione avvengono a distanza di tempo:

1. Nel caso di **rinnovo automatico**, se il nome a dominio si trova in uno stato che lo consenta, l'addebito avviene all'atto della scadenza del nome a dominio. La fatturazione, invece, avviene alla scadenza del grace period, detto autoRenewPeriod: se durante tale periodo, il nome a dominio viene trasferito ad altro Registrar o cancellato, il costo del rinnovo viene ri-accreditato al Registrar e il rinnovo non viene fatturato.
2. Nel caso di **trasferimento**, l'addebito dell'operazione avviene all'atto della richiesta di modifica del Registrar. La fatturazione avviene soltanto nel momento in cui l'operazione termina positivamente. Se l'operazione termina negativamente, il costo del trasferimento viene ri-accreditato al Registrar che l'ha richiesta e l'operazione non viene fatturata.

Vediamo in sintesi quali sono le azioni per ogni singolo controllo di questo gruppo:

1. Scadenza di un nome a dominio

Alla scadenza di un dominio si hanno sostanzialmente tre situazioni:

- a. Il Registrar che gestisce il dominio ha sufficiente credito per rinnovare il dominio. In questo caso, viene addebitato al Registrar il costo del rinnovo ed il dominio entra nello stato di ok/autoRenewPeriod. Viene inserito nella coda di polling, un messaggio sull'avvenuto passaggio di stato.
- b. Il Registrar che gestisce il dominio non ha sufficiente credito per rinnovare il dominio. In questo caso il dominio entra nello stato di inactive/notRenewed. Viene inserito nella coda di polling un messaggio sull'avvenuto passaggio di stato.
- c. Il Registrar che gestisce il dominio non ha più un contratto attivo con il registro cioè il dominio si trova nello stato ok/noRegistrar. In questo caso il dominio entra nello stato inactive/noRegistrar. Viene inviata una notifica al Registrant che il dominio è entrato in questo stato.

Ovviamente, non tutti i domini scaduti sono presi in considerazione. Vengono scartati i domini che sono nei seguenti stati:

- a. autoRenewPeriod (combinato con altri stati)
- b. pendingDelete/pendingDelete e pendingDelete/redemptionPeriod
- c. pendingTransfer/bulk
- d. inactive/revoked
- e. inactive/toBeReassigned
- f. inactive/notRenewed
- g. inactive/noRegistrar

2. Scadenza dello stato di pendingUpdate

Lo stato pendingUpdate inizia allorché un Registrar invia una richiesta di cambio della configurazione DNS per un proprio dominio al quale è associata una configurazione DNS precedentemente validata. Quindi durante lo stato di pendingUpdate, un dominio ha associato due diverse configurazioni DNS:

- quella vecchia, già validata precedentemente
- quella nuova, in corso di validazione

Alla scadenza dello stato di pendingUpdate, la nuova configurazione DNS viene dismessa ed il dominio ritorna nello stato in cui era precedentemente all'invio della richiesta di cambio name server con associata la sola vecchia configurazione.

E' da sottolineare che, durante il periodo di pendingUpdate, il domino risulta comunque delegato nella zona .it con la vecchia configurazione DNS.

Viene inserito un messaggio di notifica sul cambiamento di stato del dominio nella coda di polling del Registrar.

3. Scadenza dello stato pendingTransfer

La scadenza dello stato pendingTransfer permette ad un trasferimento di concludersi con successo. Il trasferimento consiste in pratica nel passaggio di un nome a dominio da un Registrar chiamato cedente ad un altro Registrar chiamato cessionario. In pratica, la scadenza dello stato pendingTransfer corrisponde all'operazione EPP Domain Transfer con opzione op="approve" inoltrata dal Registrar cedente. Durante tale passaggio è necessario duplicare i contatti Registrant e Admin (se diverso da Registrant) del Registrar cedente ad assegnarli al Registrar cessionario. I contatti Tech non vengono duplicati e vengono rimpiazzati con il duplicato del contatto Registrant.

La duplicazione del contatto Registrant non è necessaria qualora la richiesta di trasferimento contenga l'opzione **trade** (non è il caso contemplato dalla fig.7) la quale permette di effettuare contestualmente il cambio Registrar e Registrant.

La data di scadenza del dominio viene aggiornata e l'operazione di trasferimento, addebitata al Registrar cessionario all'atto della richiesta, viene fatturata.

Vengono inseriti due messaggi di notifica del completamento del trasferimento nelle code di polling del Registrar cedente e del Registrar cessionario.

Nel caso in cui il trasferimento sia iniziato durante l'autoRenewPeriod allora è necessario riaccreditare il costo del rinnovo automatico al Registrar cedente.

Nel caso in cui il trasferimento è di tipo trade, viene inviata una mail al vecchio ed al nuovo Registrant per notificare l'avvenuto cambio di Registrant.

4. Scadenza dello stato autoRenewPeriod

La scadenza dello stato autoRenewPeriod consiste essenzialmente nelle seguenti azioni:

- a. Aggiornamento della data di scadenza del dominio eccetto nel caso in cui il dominio è nello stato pendingTransfer/autoRenewPeriod
- b. Fatturazione dell'operazione di rinnovo, addebitata all'atto della scadenza del dominio
- c. Uscita dallo stato autoRenewPeriod (cioè lo stato viene rimosso dalla lista degli stati associati al dominio)
- d. Inserimento nella coda di polling del Registrar di un messaggio sull'avvenuto passaggio di stato

Nel caso in cui il dominio fosse nello stato di pendingTransfer/autoRenewPeriod, al Registrar cedente viene riaccreditato il costo del rinnovo.

5. Scadenza dello stato pendingDelete/redemptionPeriod, inactive/notRenewed, inactive/revoked, inactive/noRegistrar

In tutti questi casi si agisce nello stesso modo:

- a. Se il dominio non è nello stato challenged, il dominio passa nello stato pendingDelete/pendingDelete
- b. Se il dominio è nello stato challenged, il dominio passa nello stato inactive/toBeReassigned e quindi passa sotto la gestione del Registro

Viene inserito nella coda di polling del Registrar un messaggio sull'avvenuto passaggio di stato.

6. Scadenza dello stato inactive/toBeReassigned

Il dominio passa nello stato pendingDelete/pendingDelete. Viene inserito nella coda di polling del Registrar un messaggio sull'avvenuto passaggio di stato.

4.2.2 Controlli per il calcolo delle varie soglie di credito per i Registrar

Cron expression1: 0 15 1,3,5,9,11,13,15,17,19,21,23 * * ?

Cron expression2: 0 15 7 * * ?

Questo gruppo funzionale di controlli prevede due diversi gruppi temporali in quanto per la sola attivazione delle ore 7 e 15 minuti di ogni giorno è previsto che venga eventualmente inviata una mail ai contatti di tipo billing che ciascun Registrar ha dichiarato nel portale RAIN-NG.

Per comprendere meglio le azioni di questo gruppo di controlli occorre spiegare il concetto di soglia. Nel sistema sono implementate attualmente due diverse soglie:

- **Soglia**

È il limite minimo di credito necessario per assicurare il mantenimento dei domini di prossima scadenza ed in base a tale valore ed al valore corrente del credito aggiorna lo stato associato ad un Registrar. Gli stati associati ad un Registrar sono:

- a. **normalBalance:** il Registrar può richiedere qualsiasi operazione
- b. **lowCredit:** il credito del Registrar è al di sotto della soglia
- c. **outOfFunds:** il credito del Registrar è vicino allo zero (in pratica è minore del costo di una qualsiasi operazione) e quindi al Registrar sono permesse le sole operazioni non fatturabili

Il numero dei giorni da considerare per calcolare la soglia è variato nel corso degli anni passando dall'iniziale 15 all'attuale 4. Anche il significato della soglia è cambiato. Fino all'entrata in vigore del contratto Registrar del 1 Maggio 2016, un Registrar che aveva il credito sotto la soglia non poteva effettuare operazioni fatturabili e quindi ci si riferiva a tale soglia con il termine di "soglia bloccante". Successivamente al 1 Maggio 2016 è stato rimosso questo vincolo ed è stata eliminata anche la "soglia di avviso" (una soglia calcolata nello stesso modo ma su un arco temporale di 21 giorni) che aveva lo scopo di avvisare i Registrar dell'imminente raggiungimento della "soglia bloccante". Di fatto l'attuale soglia ha sostituito la vecchia "soglia di avviso". Il Registro ha fatto questa scelta allo scopo di agevolare la normale operatività dei Registrar.

La soglia viene calcolata abbastanza spesso durante la giornata, cioè ogni due ore, perché il credito di un Registrar può variare molto in base alle operazioni che richiede al server EPP. Ciò può avvenire sia verso il basso, qualora il Registrar esegua parecchie operazioni che generano addebito, sia verso l'alto se il Registrar richiama ad esempio parecchie cancellazioni per domini in autoRenewPeriod. In questo caso, infatti il sistema provvede a rifondere il Registrar del costo del rinnovo automatico.

- **Soglia personalizzata**

A differenza della precedente non è calcolata automaticamente dal sistema ma è impostata dal Registrar nel portale RAIN-NG

L'unico controllo appartenente a questo gruppo esegue i seguenti passi:

- a. Invocazione del controllo per l'eventuale recupero dei domini in inactive/notRenewed
- b. Ricerca dei Registrar per i quali il credito risulta inferiore alla soglia personalizzata. Nel caso in cui il credito risulti inferiore alla soglia personalizzata, viene inserito un messaggio nella coda di polling e viene inviata una mail nel caso in cui l'attivazione è quella delle ore 7 e 15 minuti

- c. Calcolo della soglia di credito per ogni Registrar ed in base a questa ed al valore del credito aggiornamento eventualmente dello stato del Registrar. Per tutti i Registrar che hanno variato il proprio stato, il sistema invia un messaggio nella coda di polling e invia una mail nel caso in cui l'attivazione fosse quella delle ore 7 e 15 minuti

4.2.3 Controlli per la cancellazione definitiva dei domini

Cron expression1: 0 0 9,16 * * ?

Cron expression2: 0 5 9,16 * * ?

Cron expression3: 0 15/20 * * * ?

Ognuna delle tre attivazioni corrisponde ad un gruppo che contiene un solo controllo:

- a. Alla prima cron expression corrisponde il controllo che rende disponibili in corrispondenza delle ore 9:00:00 e 16:00:00 i nomi associati ai domini la cui cancellazione è prevista dal drop time.
- b. Alla seconda cron expression corrisponde il controllo che parte esattamente 5 minuti dopo la prima e completa la transizione di stato da pendingDelete/pendingDelete a deleted. I domini che devono essere cancellati non sono rimossi definitivamente dal database del Registro ma permangono con lo stato deleted. Ciò perché per ogni nome a dominio deve essere comunque conservata la sua storia dalla registrazione alla rimozione dalla lista dei domini registrati. Per cercare di limitare al massimo i conflitti di concorrenza fra il controllo che effettua il drop time e le operazioni di ri-registrazione indirizzate dai Registrar verso il server epp-deleted.nic.it, è stata appunto implementata una cancellazione in due passi successivi.
- c. Alla terza cron expression corrisponde il controllo che verifica ogni 20 minuti che nessun nome a dominio con lo stato pendingDelete/pendingDelete sia stato messo nello stato challenged. In questo caso, il nome a dominio è tolto dalla lista del drop time e viene messo nello stato inactive/toBeReassigned.

4.2.4 Controlli per il re-invio delle richieste di verifica della configurazione DNS

Cron expression1: 0 10/30 * * * ?

Cron expression2: 0 10 9 * * ?

I controlli di questo gruppo vengono attivati con due cron expression:

- a. La prima cron expression è relativa ad un controllo che è attivato ai minuti 10 e 30 di ogni ora e ricerca quali sono le richieste di verifica della configurazione DNS per i domini che si trovano nello stato pendingUpdate e inactive/dnsHold da meno di un mese.
- b. La seconda cron expression è relativa ad un controllo che è attivato alle sole ore 9.10 di ogni giorno e ricerca quali sono le richieste di verifica della configurazione DNS per i domini che si trovano nello stato inactive/dnsHold da più di un mese.

Questa suddivisione si è resa necessaria da quando è stata eliminata la scadenza dello stato inactive/dnsHold.

In entrambe i controlli le richieste trovate vengono re-inviate nella coda **dns.validation.requests** in attesa che vengano elaborate dal tool di verifica della configurazione DNS. Le risposte vengono inserite dal tool nella coda **dns.validation.responses** menzionata precedentemente.

4.2.5 Controlli per il recupero di domini nello stato inactive/notRenewed

Cron expression1: 0 20 0/2 * * ?

Cron expression2: 0 50 * * * ?

Questo gruppo funzionale prevede due diverse attivazioni rese necessarie dal fatto che nelle ore dispari lo stesso controllo viene chiamato all'interno del calcolo della soglia. La sommatoria delle due attivazioni corrisponde, in pratica, ad eseguire questo controllo ai minuti 20 e 50 di ogni ora.

Il controllo rileva quali sono i domini nello stato inactive/notRenewed che possono essere rinnovati perché il Registrar ha provveduto a rifondere il suo credito. Il controllo, ovviamente, effettua l'operazione di rinnovo e cambiamento di stato fin tanto che risulta esserci credito sufficiente. Quindi è possibile che non tutti i domini non rinnovati per mancanza di credito possano essere recuperati.

Per tutti quei domini che sono stati recuperati oltre l'autoRenewPeriod, in aggiunta all'addebito, si procede alla fatturazione e all'aggiornamento della data di scadenza. Viene inserito nella coda di polling dei Registrar un messaggio che notifica il cambiamento di stato per ogni dominio recuperato.

4.2.6 Controlli per la verifica di consistenza su domini dopo operazioni del Registro

Cron expression1: 0 7/15 6-20 ? * MON-FRI

Questo gruppo funzionale, che corrisponde ad un'unica attivazione temporale, comprende tre controlli che vengono attivati ogni 15 minuti a partire dal minuto 7 dalle ore 6 alle ore 20 da Lunedì al Venerdì:

1. Un controllo che ricerca i domini oggetto di un bulk transfer completata con successo ed iniziata in autoRenewPeriod. In base al numero dei domini ottenuti per ogni bulk transfer, viene calcolato l'intero ammontare da riaccreditare al Registrar cedente come somma dei costi dei singoli rinnovi. Viene inserito nella coda di polling di ciascun Registrar cedente individuato un messaggio che riporta l'identificatore della bulk transfer, il numero dei domini trasferiti in autoRenewPeriod e l'ammontare riaccreditato
2. Un controllo che ricerca i domini oggetto di una revoca avvenuta in autoRenewPeriod. Viene inserito nella coda di polling di ciascun Registrar individuato un messaggio che riporta il nome del dominio, la data in cui era stato effettuato il rinnovo ed il costo del rinnovo riaccreditato.
3. Un controllo che ricerca i domini oggetto di una cancellazione effettuata dal Registro in autoRenewPeriod. Viene inserito nella coda di polling di ciascun Registrar individuato un messaggio che riporta il nome del dominio, la data in cui era stato effettuato il rinnovo ed il costo del rinnovo riaccreditato.

L'attivazione prende in considerazione i soli giorni feriali perché le situazioni a cui le tre controlli si rivolgono scaturiscono da operazioni effettuate da operatori del Registro.

4.2.7 Controllo per la cancellazione dei messaggi obsoleti

Cron expression: 0 20 3 1,15 * ?

L'unico controllo appartenente a questo gruppo viene attivato alle ore 3:20 del giorno 1 e 15 di ogni mese ed ha il compito di rimuovere dalla coda di polling tutti i messaggi presenti da più di 60 giorni.

4.2.8 Controllo per rimozione dei contatti non più utilizzati

Cron expression: 0 10 3 1,15 * ?

L'unico controllo appartenente a questo gruppo viene attivato alle ore 3:10 del giorno 1 e 15 di ogni mese ed ha il compito di rimuovere i contatti duplicati non utilizzati da più di 60 giorni. I contatti duplicati sono quei contatti che sono stati creati dal sistema a seguito del completamento con successo di un'operazione di trasferimento. I contatti duplicati sono facilmente identificabili perché iniziano con la stringa "DUP" la quale non può essere utilizzata come prefisso per la creazione dei contatti operata dai Registrar.

4.2.9 Controllo per azzerare il limite dei comandi EPP

Cron expression: 0 0 23 * * ?

Anche questo gruppo comprende un unico controllo che viene attivato alle ore 23:00:00 di ogni giorno. Il controllo ha lo scopo di azzerare il conteggio dei comandi inviati a i server epp.nic.it ed epp-deleted.nic.it.

Nella tabella di seguito sono riportati i limiti dei comandi per i due server di produzione.

Parametro	Descrizione	Valore
MAX_CMD	Numero massimo di comandi Check Domain per Registrar che possono essere inviati al giorno al server epp.nic.it	20000
	Numero massimo di comandi Check Domain per Registrar che possono essere inviati al giorno al server epp-deleted.nic.it	3000
	Numero massimo di comandi Create Domain per Registrar che possono essere inviati al giorno al server epp-deleted.nic.it	

I numeri dei comandi da considerare inviati da ciascun Registrar ai due server di produzione sono conservati su un database Redis⁽⁸⁾. Redis è un database di strutture dati che consente la conservazione di varie tipologie di strutture dati come liste, insiemi, tabelle hash e insiemi ordinati. Fra le caratteristiche che vale la pena di elencare c'è la possibilità di configurare database come repliche master-slave e database in cluster.

Il controllo comunica con il server Redis attraverso un client Redis per Java detto Jedis. Le classi ed i metodi del jar Jedis permettono di aprire una connessione con il server Redis ed inviare comandi.

Il controllo invia il comando che chiede al server Redis di azzerare alle ore 23:59:59.999 le strutture dati che contengono le informazioni relative al numero dei comandi inviati da ciascun Registrar per ciascuno dei due server di produzione.

5. Conclusioni

Il sistema sincrono di gestione dei nomi a dominio del Registro Italiano implementa una serie di procedure che operano in modalità asincrona.

L'introduzione di tale procedure, se in linea generale, è utile per evitare situazioni in cui l'elaborazione sia bloccata in attesa del completamento di certe azioni, è ancora di più giustificata nei sistemi che si occupano della gestione dei nomi a dominio sia per la natura intrinseca di certe operazioni (vedi il trasferimento di un dominio) sia per le condizioni dettate dallo specifico regolamento del Top Level Domain (in questo caso del .it).

L'adozione della tecnologia EJB può essere senz'altro di aiuto nell'implementazione di tutte quelle situazioni che richiedano una comunicazione asincrona fra due agenti software come pure l'attivazione di precise procedure alla scadenza di particolari periodi temporali.

Gli EJB sono le componenti più importanti della piattaforma J2EE dedicata allo sviluppo di applicazione distribuite e supportano quella che viene chiamata **programmazione ad eventi**. Inizialmente, sotto questa categoria, ricadevano le sole applicazioni che prevedevano un'interazione con un utente "umano" (come una qualsiasi applicazione web basata su un'interfaccia grafica) ma, nel corso degli anni, il linguaggio Java si è arricchito di nuove tecnologie per consentire l'attivazione di porzioni di codice attraverso chiamate da remoto o attraverso la gestione di eventi (ricezioni di messaggi, attivazione di job da parte di uno scheduler).

Nell'architettura del sistema sincro, oltre che nei controlli periodici, è stato fatto ampio ricorso all'utilizzo delle code gestite dal MOM ActiveMQ per implementare un certo disaccoppiamento fra le varie funzionalità, qualora ciò fosse corretto da un punto di vista concettuale ma anche vantaggioso da un punto di vista dell'affidabilità e delle prestazioni. Ciò ha reso così di fatto la stessa architettura maggiormente service-oriented.

Una prospettiva interessante per quanto riguarda gli sviluppi futuri è perseguire questa filosofia cercando di ampliare l'insieme dei servizi disponibili (intesi sia come servizi sincroni che asincroni) ottenuti isolando un certo numero di funzionalità non banali richiamate o innescate all'occorrenza tramite EJB o tramite Web Services.

Bibliografia

1. Hollenbeck, S. (2009, August). *Extensible Provisioning Protocol (EPP)*. Tratto da <https://tools.ietf.org/html/rfc5730>.
2. Registro .it. (2014, Novembre). *Gestione delle operazioni sui nomi a dominio nel ccTLD .it*. Tratto da http://www.nic.it/sites/default/files/docs/Linee_Guida_Tecniche_Sincrone_v2.1_0.pdf.
3. Oracle. (s.d.). *Enterprise JavaBeans Technology*. Tratto da <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>.
4. Oracle. (s.d.). *Java Message Service Concepts*. Tratto da <https://docs.oracle.com/javaee/6/tutorial/doc/bncdq.html>.
5. The Apache Software Foundation. (s.d.). *ActiveMQ*. Tratto da <http://activemq.apache.org/>.
6. Terracotta, I. (s.d.). *QUARTZ Job Scheduler*. Tratto da <https://quartz-scheduler.org/>.
7. XStream Development Team. (s.d.). *XStream*. Tratto da <http://x-stream.github.io/>.
8. Salvatore Sanfilippo, P. N. (s.d.). *Redis*. Tratto da <http://redis.io/>.