Interoperabilità e migrazione verso il cloud – il caso DotNetLombardia

#### di [Roberto Freato](https://mvp.support.microsoft.com/profile=9F9B3C0A-2016-4034-ACD6-9CEDEE74FAF3)

* 1. 

*Marzo 2012*

In questo articolo verranno discussi i seguenti argomenti:

* + Interoperabilità
  + Soluzioni Ibride on-the-cloud
  + Metodologia ed approccio alla migrazione verso il cloud

E le seguenti tecnologie:

* + Windows Azure Compute
  + SQL Azure
  + Telligent Community
  + Amazon Web Services S3

La migrazione di ambienti e applicazioni esistenti verso il cloud è un tema caldissimo e su cui, molto probabilmente, i consulenti e le aziende IT fonderanno parte del loro business nei prossimi anni. Il motivo che spingerà questa fetta di operazioni IT a diventare un vero proprio business è dovuto in prima approssimazione a due grandi variabili:

* 1. Necessità di mantenere in parte le applicazioni on-premise
  2. Scetticismo verso l’affidamento totale dei propri sistemi/applicazioni al cloud
  3. Ottimizzazione dei costi

Chiaramente negli scenari reali che ho incontrato c’erano tipicamente tutti questi punti ben mescolati tra loro. Oggi quindi, a parità di offerta cloud, è interessante capire quale vendor punti di più sulla tematica che meglio sposa le problematiche sopra, ovvero: l’interoperabilità.

## Interoperabilità

L’interoperabilità è un punto cardine di molte strutture software complesse oggi implementate nella maggior parte delle realtà informatiche. Spesso si impiegano molte risorse e molto tempo per perseguire standard di interoperabilità molto alti, inseguendo la cosiddetta “compatibilità in avanti”, sfruttando le tonnellate di esperienza che gli esperti IT hanno accumulato negli anni d’oro dell’evoluzione informatica. È così nato un sano e costruttivo senso critico verso soluzioni troppo chiuse e con un rapporto costi-futuri/benefici troppo alto. Questi costi futuri, spesso occulti ad inizio adozione, hanno reso per anni le aziende vincolate ad un solo vendor per tutta una serie di prodotti: pena la perdita di funzionalità o la preclusione totale di un certo servizio aggiuntivo.

## Nel cloud computing

Ritornando al Cloud Computing, l’interoperabilità può essere espressa in termini di “possibilità di un sistema di Cloud di interagire con altri sistemi Cloud e non”; quindi in effetti, la possibilità di una soluzione di interfacciarsi concretamente con uno o più sistemi pre-esistenti legacy di un cliente, il quale in futuro vorrà magari utilizzare anche più sistemi Cloud diversi, ed integrarli tra loro. È quindi ovvio che, il grado di interoperabilità di una soluzione Cloud si formalizza nella presenza o meno di strumenti per operare in ingresso e in uscita: ovvero è necessario che la soluzione abbia un considerevole bagaglio di tecnologie abilitanti in modo che possa essere reperita e utilizzata da “fuori”, ma anche e soprattutto che possa a sua volta utilizzare in cascata altri sistemi Cloud connessi.

Si prenda per esempio un processo orientato ai servizi tale che, come in una semplice orchestrazione, vi sia una cascata di chiamate da un attore ad un altro. Ora, se un ipotetico scenario di questo tipo fosse riportato in un contesto Cloud, significherebbe avere una serie di provider Cloud abilitati ad essere orchestrati tra di loro, tali da creare un workflow omogeneo tra sistemi eterogenei. Se tale sistema già esistesse, saremmo già in condizioni ideali; in realtà la maturità del Cloud Computing è ancora abbastanza indietro e, come abbiamo visto nella prima parte di questa trattazione, sono delicati i rapporti tra le varie istituzioni, tant’è che oggi si è ancora in alto mare per quanto riguarda la definizione di standard che regolino il moderno Cloud Computing.

## Soluzioni ibride per forza maggiore

Benchè, come detto sopra, ci siano generalmente almeno 3 motivi per adottare soluzioni ibride, spesso accade che la tecnologia esistente limiti fortemente il passaggio totale al cloud oppure che un vendor non soddisfi completamente le necessità di una applicazione complessa e che quindi, per la stessa applicazione, si renda necessario utilizzare più vendor contemporaneamente.

Il caso di studio che porto all’attenzione è quello di una migrazione su Windows Azure del portale della community lombarda DotNetLombardia.

## Migrazione di DotNetLombardia

Come molti sanno, DotNetLombardia (in seguito DNL) è una soluzione web basata su Community Server (oggi Telligent Community 5.6) che, secondo lo stesso supporto Telligent, non dovrebbe girare su Azure (per cui non è neanche ufficialmente supportato).

Questo perchè, in primis, il database utilizza features “non permesse” e, problema veramente gigantesco, non c’è accesso ai sorgenti per modificare le parti necessarie all’adattamento. Parto sempre dal presupposto che con l’accesso ai sorgenti tutte le applicazioni possano più o meno facilmente passare al cloud: la vera sfida è stato farlo in una applicazione senza sorgenti, a prova del fatto che Azure è una piattaforma di PaaS unica nel suo genere.

## SQL Azure

Per prima cosa era necessario migrare il Database (SQL Server) a SQL Azure. Si è tentato candidamente di fare un export/import su SQL Azure con i tools standard (SQL Server Management Studio) ma ovviamente (a conferma di quanto detto dal supporto) la cosa non ha funzionato.

A questo punto è stata necessaria una modifica di alcune chiamate non supportate nello script di generazione del database SQL Azure: modifiche T-SQL su porzioni incriminate verso qualche struttura consentita (in particolare il problema si presentava con le sp\_preparedocument e la chiamata a OPENXML, sostituite eccellentemente con nodes() e values()). Con un pò di fortuna e il supporto di alcuni membri della community (soprattutto per il T-SQL) ho fatto girare lo script e ho congelato la struttura DB.

## Windows Azure

Ho appreso anche io con un certo meravigliato stupore che Microsoft non millanta quando afferma che migrare a Windows Azure è semplice come fare l’upload della propria attuale soluzione: è avvenuto proprio questo. Ho fatto un ServiceConfiguration, un ServiceDefinition e ho usato manualmente il comando CSPack per creare il package dell’attuale applicazione web Telligent Community (un WebSite ASP.NET). Un punto a sfavore di questa soluzione è stato non poter esternalizzare i settings sul ServiceConfiguration, data la mancanza dei sorgenti dell’applicazione.

## File Storage

Il problema è che Telligent Community usa di default una cartella su FileSystem per salvare i file: immagini, allegati dei post, archivi, etc.. Questa cosa, sebbene sia possibile su Azure, sarebbe stata anti-cloud, poichè ci avrebbe reso dipendenti dalla situazione post-deploy ed invalidato completamente l’ottica Cloud che tanto si cercava di percorrere.

Mi sarebbe piaciuto concludere con l’esternalizzazione dell’accesso al file system su Azure Storage, scrivendo eventualmente un custom provider ad arte. Sebbene Telligent consenta la scrittura di tali provider, ne avrei dovuti scrivere troppi, quando di default Telligent ne ha implementati alcuni per l’esternalizzazione dello storage su un competitor di Microsoft: AWS S3, che ha aggiunto la chicca a questa soluzione di Cloud Integration.

## Conclusioni

In questo articolo abbiamo parlato di interoperabilità, soluzioni ibride e migrazione Cloud, analizzando i benefici derivanti dall’utilizzo congiunto di più vendor di cloud computing. Abbiamo portato inoltre ad analisi il caso eclatante di DotNetLombardia, che è tutt’ora attiva e online su Azure con un software chiuso portato al cloud.

## Riferimenti

Parte del materiale di questo articolo è già presente sul blog: <http://dotnetlombardia.org/b/rob/default.aspx>.

#### di Roberto Freato ([blog](http://dotnetlombardia.org/blogs/rob/default.aspx)) - Microsoft MVP

* 1. [*Altri articoli di Roberto Freato nella Libr*](http://sxp.microsoft.com/feeds/3.0/msdntn/TA_MSDN_ITA?contenttype=Article&author=Roberto%20Freato)*ary* [](http://sxp.microsoft.com/feeds/3.0/msdntn/TA_MSDN_ITA?contenttype=Article&author=Roberto%20Freato)