Windows Azure – la storia

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

* 1. 

*Settembre, 2012*

In questo articolo verranno discussi i seguenti argomenti:

* + Storia di Windows Azure

## Sommario

* + [Origini](#_Origini)
  + [Linea del tempo](#_Linea_del_tempo)
  + [L’ecosistema Azure](#_L’ecosistema_Azure)
  + [Evoluzione dei pattern architetturali](#_Evoluzione_dei_pattern)
  + [Conclusioni](#_Conclusioni)

In questo articolo parleremo della storia di Windows Azure, partendo da un solido background informativo sul Cloud, trattato nell’articolo “Il Cloud Computing – origini”, per arrivare a quello che Azure rappresenta oggi, nel panorama internazionale dei prodotti cloud di massa.

## Origini

Penso che l’idea di puntare su una piattaforma di Cloud Computing di massa, come molte delle buone idee di Microsoft, sia nata in primis da esigenze interne. Spesso è capitato, nella storia dei prodotti Microsoft, di vederne di molto buoni e molto aderenti alle necessità dei “clienti” (aziende, utenti): prodotti come Exchange, Office, Active Directory e le ultime soluzioni di Unified Communications sono esempi eccellenti.

Proprio da questi esempi, il cui sviluppo è stato aiutato molto spesso da un intensivo utilizzo interno a Microsoft Corporation, partiamo per definire la necessità di una piattaforma di Cloud.

Nel primo lustro del nuovo secolo Microsoft ha consolidato numerosi prodotti online e ampliato la sua base di utenti fino ad essere uno dei player più importanti al mondo sul web. Questo, come già visto per Google e Amazon, ha fatto emergere sempre di più la domanda di datacenter interni e utilizzabili in modo elastico, riducendo i costi, i consumi, l’inquinamento e, perchè no, incentivando una palestra interna anche sul fronte della gestione di complessi sistemi IT.

E così, mentre Exchange Online faceva enormi passi avanti, Bing si consolidava, Office Live Meeting esplodeva (in termini di consenso) e l’immortale Windows Live ID continuava la crescita lineare sempra avuta, Microsoft ha deciso di debuttare sullo scenario del Cloud di massa.

* 1. **Nota:**
  2. Si fa sempre riferimento al cloud di massa, noto anche come Consumer Cloud, in netta contrapposizione al cloud business già presente da moltissimi anni in varie forme (es. Akamai, etc).

## Linea del tempo

Windows Azure nasce nei laboratori MS con il codename di Red Dog, per quanto ne sappiamo, tempo prima dell’annuncio ufficiale di Ray Ozzie nell’Ottobre 2008 (durante la PDC di quell’anno).

L’idea alla base, oltre la crescente offerta dei competitors come Google (e il suo App Engine), era quella di creare siti e applicazioni web che potessero scalare senza doversi preoccupare dell’infrastruttura hardware a supportarlo. Così, nell’Ottobre 2008, Microsoft lancia una piccola ma sostanziosa CTP per gli appassionati del nuovo giocattolo.

* 1. **Note:**
  2. Un primordiale video satirico a questo indirizzo (<http://www.youtube.com/watch?v=NZO9YE1ZvqE>) spiega la “semplice” necessità dietro al nuovo progetto.

Avendo avuto modo di sperimentare Azure in profondità sin dalla prima CTP, ho potuto vedere l’evoluzione della piattaforma sia nella sua fattispecie tecnica che nella sua veste commerciale e di offerta complessiva. Sebbene oggi ci siano molti servizi sotto il grande cappello del nome Azure, all’inizio del cammino potevamo comunque notare già buona parte dei tasselli fondamentali, semmai con nomi diversi:

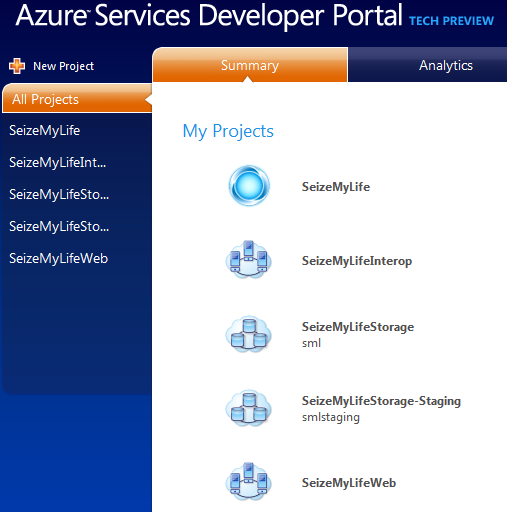
* + Windows Azure Storage
  + Windows Azure Hosted Services

Microsoft voleva già dare una base completa per permettere ai clienti di fare girare una applicazione sul cloud e, in effetti, tramite un hosted service (già differenziato in Web Role e Worker Role) più una serie di strategie per la gestione dello storage (nei già presenti Blobs, Tables e Queues) era già possibile implementare in modo efficace.

Una cosa che ho personalmente apprezzato all’inizio del cammino della CTP era la impossibilità di fare girare il nostro codice in Full-Trust, permettendo quindi solo un “Azure Partial Trust” molto limitativo e decisamente isolato. L’approccio aveva un grande fascino: permetteva sì poche uscite di rotta, ma allo stesso tempo incanalava la soluzione in un modello di sviluppo fatto da best-practice che spesso sia gli sviluppatori che le case produttrici di software, dimenticano.

Proprio sotto grande richiesta di queste ultime, che si vedevano troncare il 90% del loro codice verso il FileSystem, il registro, la rete, etc, il team di Azure ha introdotto successivamente il supporto al Full-Trust, introduzione che ha dato il via ad una serie di future introduzioni volte a IaaS-izzare il PaaS tanto caro. Così, nel Marzo 2009, dopo soli 4 mesi di CTP, Microsoft ha allentanto le briglie di Azure permettendo l’esecuzione del codice Nativo e del FullTrust per le applicazioni web: a cascata ne è nato il supporto nativo per PHP tramite FastCGI. Ultimo grande allentamento della logica restrittiva del PaaS lo avremo in seguito con l’introduzione del supporto a Remote Desktop per le istanze di calcolo.

E per quanto riguarda gli SDK? All’inizio “avevamo” a disposizione ben poco, probabilmente già tanto ma rispetto alla piena integrazione di oggi, il processo di sviluppo ne era un pò rallentato. Il portale era sconnesso da Visual Studio e il publishing si faceva a mano caricando CSPKG e CSCFG: incollo qui uno screenshot dell’epoca per il suo intrinseco valore “storico”:

* 1. 

Tolte le icone futuristiche, il portale permetteva poco più di un upload dei pacchetti, degli eventuali certificati SSL più la gestione di base dei servizi (Start/Stop/Upgrade/Delete/Swap), enfatizzando molto la funzionalità di VIP-Swap Staging-Produzione presente anche oggi.

L’SDK per Visual Studio era composto dal già ben funzionante Development Fabric, che permetteva l’esecuzione e il debug della soluzione in locale. Da Maggio 2009, dopo 3 mesi dall’apertura di Azure verso nuovi scenari (aka FullTrust), è stato introdotto l’SDK per PHP che andrà poi a finalizzarsi in un PlugIn per Eclipse (nell’Ottobre del 2009) sia per PHP che per Java.

Il supporto a Java è sempre stato oggetto di particolari critiche, da parte mia in primis, per la natura poco chiara del come implementarlo su un PaaS fondamentalmente .NET. Il supporto alle tecnologie non nativamente supportate da IIS è infatti anche oggi ottenuto utilizzando dei derivati software, comunemente chiamati acceleratori (ma noti anche sotto altri nomi) che una volta installati sul ruolo di Azure avrebbero innescato una serie di operazioni autoinstallanti al fine di creare la topologia corretta.

Ad esempio, l’acceleratore per Java, consisteva in un maxi-pacchetto in cui, una volta scritta l’applicazione (per esempio Servlet/JSP) essa veniva decorata anche di tutto l’ambiente di runtime (Tomcat, Librerie, etc) più un software che, una volta caricato il tutto sul PaaS, avrebbe scompattato, installato e configurato tutta la grande macchina. Ho sempre rinnegato questa soluzione indicandola come uso “improprio” del PaaS motivo per cui, anche da parte degli utenti, non è stata una strada più di tanto perseguita.

La data storica di inizio di Azure la possiamo ri-delineare verso l’inzio del 2010, quando Azure è entrato a fare parte dell’offerta commerciale di Microsoft, offrendo i suoi servizi a pagamento. I possessori di un abbonamento MSDN avrebbero avuto dei forti sconti (che tuttora hanno) e le macchine in CTP sarebbero state progressivamente rimosse. Di pari passo Microsoft ha incrementato ovviamente il suo pool di datacenters sparsi per il mondo fino ad arrivare a quelli oggi disponibili (più nodi in America, Europa, Asia).

## L’ecosistema Azure

Cose che oggi reputiamo normali, anche solo nel 2008 erano estremamente innovative. Ad esempio l’eventualità di poter offrire un servizio di Single-Sign-On come AppFabric Access Control Service (più volte rinominato ed oggi Windows Azure Active Directory) era una delle chicche più gettonate. Introduceva in un mondo che cominciava a masticare seriamente i claims, un modello completamente claims-based e integrato in Windows Identity Foundation per autenticare-as-a-service gli utenti sulle nostre applicazioni. Non nego che fino ad allora avevo ignorato alla grande tutto WIF, reputandolo uno strumento di nicchia: oggi invece, il mondo cambia, non ne potremmo fare a meno (perlomeno per semplificare di moltissimo le operazioni banali).

Ed è così che, servizio dopo servizio, prodotto dopo prodotto uscito dalla coda delle Beta ed entrato in produzione, Azure ha cominciato ad offire un portafoglio di servizi diversificato per poter portare tutta l’infrastruttura IT sui datacenter distribuiti.

Ed è così che hanno visto la luce alcuni tools usciti dal SQL Azure Labs (Reporting, Data Sync, ..), altri usciti dall AppFabricLabs (Caching) e così via. Oggi il modello di sviluppo è rimasto lo stesso, proponendo servizi nuovi come Beta gratuite o quasi, per poi rilasciarli in GA (General Availability) dopo alcuni mesi di rodaggio

* 1. **Nota:**
  2. Al momento della scrittura di questo articolo, i servizi in “rodaggio” sono: Windows Azure Web Sites, Windows Azure IaaS, Media Services, Caching Preview (versione self-hosted).

## Evoluzione dei pattern architetturali

Nel cloud si è partiti a tutta velocità dichiarando cose spesso molto scostate dalla realtà. È il caso dei player non-tecnici sul mercato: alcune grandi società di ricerca e/o di consulenza e alcuni grandi player di IT hanno abusato di tutta la terminologia presente nell’ecosistema cloud, influenzando necessariamente il mercato.

Come capita spesso, chi le cose le ha fatte (leggesi Microsoft, Google, Amazon, per esempio) non si è sbilanciato a dare per morti 30 anni di informatica in un sol giorno in nome del nuovo impavido stendardo del Cloud Computing. Tuttavia, ripercorrendo la storia di Azure, abbiamo dovuto assistere a scelte di comunicazione spesso fatte per assecondare anche le posizioni più di moda in quel determinato momento.

Nel 2009 si diceva che nel giro di due anni una percentuale significativa delle applicazioni in-house (poi chiamate on-premises) sarebbero state portate sul Cloud; non si capiva di preciso se su uno IaaS, un PaaS o buttate via del tutto in favore di un SaaS: la cosa certa era che ognuno doveva affrettare la corsa agli armamenti, la stessa che ha portato al proliferare di falsi vendor di cloud, “smagnetizzati” dopo pochi mesi, una volta che il mercato ha cominciato ad essere più maturo e consapevole.

È così che Microsoft (non tutta però) ha spinto molta enfasi sulla migrazione delle proprie applicazioni in ottica completamente PaaS (fino al 2009), per poi addirittura spingere verso l’eliminazione di alcuni componenti onerosi in favore del SaaS (Office 365, Dynamics), passando dai modelli ibridi cloud/on-premises (conseguentemente alla presa di coscienza della enorme difficoltà di rendere PaaS-oriented soluzioni spesso obsolete) fino ad arrivare oggi e di nuovo alla promozione di un modello cloud full-stack in cui le lacune del PaaS possano essere colmate da una buona offerta di IaaS ed eventualmente da qualche servizio in SaaS.

Reputo corretto pensare che oggi una nuova realtà (startup soprattutto) non possa che adottare un sistema cloud completo; quello che non mi è ancora chiaro è se questa soluzione sia economicamente sostenibile sul lungo periodo, ovvero quando il break-even è superato e quando l’impresa in questione ha sviluppato e incanalato bene il proprio business.

## Conclusioni

La storia di Azure è stata una storia di tecnologia, di comunicazione e di grande dialogo con i partners/ISVs. La vera forza di Microsoft (ma anche parte della sua debolezza) è stata infatti aver seguito i consigli, suggerimenti e le richieste di chi con il cloud sarebbe dovuto andare a lavorare, stimolandone l’adozione, la consapevolezza e la conoscenza.

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

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