

Edoardo Bolzani

La guerra dei 2 mondi

Google contro Microsoft, una sfida globale.
Con le applicazioni SaaS al centro.



Prefazione

Dopo aver pubblicato “[Come i servizi SaaS \(Software as a Service\) cambiano il modo di utilizzare il PC, il Software e il nostro Business](#)”, dove illustravo come il SaaS (Software as a Service) sta cambiando il modo di utilizzare il PC, il Software e il nostro Business, è iniziata la più grande guerra mai vista nel Web.

E’ una guerra senza quartiere, che non fa prigionieri, è una guerra di religione : Microsoft contro Google.

L’occasione per scatenarla è stata l’annuncio (3/9/2008) del browser Chrome di Google che, lo hanno già confermato in tanti, è uno strumento di navigazione che fa della velocità una sua prerogativa e se un domani il suo motore verrà potenziato dal supporto delle applet e della Virtual Machine Java ecco che le preoccupazioni in casa Microsoft diventano in effetti più che lecite.

Grazie al volano del nuovo browser di Google, il datato linguaggio Java Script può finalmente vivere la stagione della sua evoluzione mentre Silverlight (*), rimarca qualcuno, scommetterà per contro molte delle sue "fiches" sulle licenze d'uso del software in ambito aziendale.

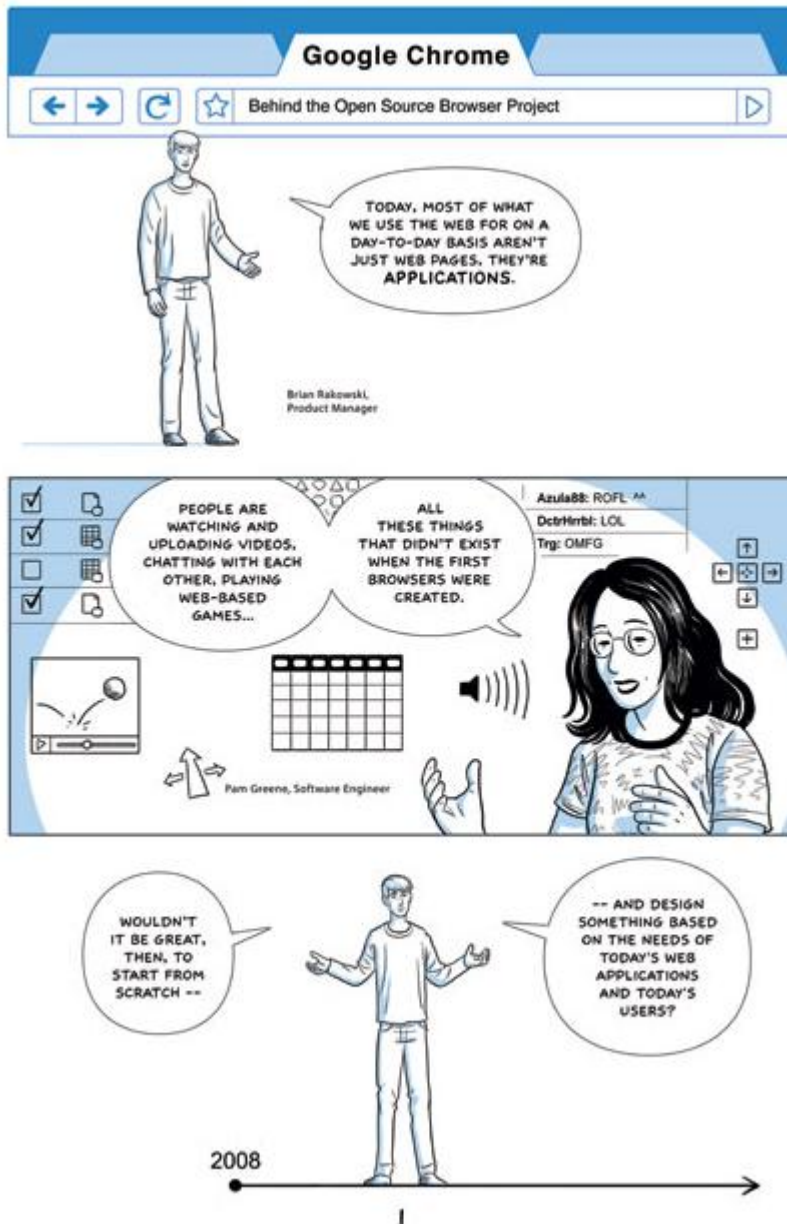
(*) Dal sito ufficiale di [Microsoft](#) “Microsoft Silverlight is a cross-browser, cross-platform, and cross-device plug-in for delivering the next generation of .NET based media experiences and rich interactive applications for the Web. ”.

Indice

Un po' di storia	Pag. 4
11 milioni di utenti!	Pag. 6
Piattaforma mobile	Pag. 7
La fine dell'incubo	Pag. 9
La percezione dell'utente	Pag. 12
PaaS (Platform as a Service)	Pag. 14

Un po' di storia

E' la sera del 3 settembre 2008 quando diventa disponibile il download del browser Chrome di Google dopo una brevissima campagna pubblicitaria dovuta ad una fuga di notizie.



E' capitato che un fumetto di 38 pagine, realizzato da un affermato fumettista statunitense (Scott McCloud) per spiegare Google Chrome è uscito su di un blog.

Il marketing Google voleva che fosse recapitato nelle caselle di posta (non di posta elettronica!) delle più influenti personalità del Web.

Dal fumetto è chiaro che per Google il Web non sono "pagine" ma "applicazioni" !

Quali applicazioni?
Quelle SaaS come Google Apps!

Nell'articolata disfida sui prodotti di produttività personale, quelli cioè che si utilizzano per scrivere ed elaborare documenti, entrambe le contendenti hanno motivo di sorridere.

In casa Microsoft si dicono a ragione soddisfatti per i risultati ottenuti da Office Live Workspace, visto e considerato che la piattaforma collaborativa on line che Redmond ha contrapposto alla tanto celebrata Google Docs ha raggiunto il milione di utenti.

Un successo (relativo) atteso materializzatosi, dicono in Microsoft, soprattutto grazie alle funzionalità collaborative della suite (che non è una versione uno a uno di quella tradizionale off line).

In casa Google, invece, la quota un milione di utenti business è stata centrata con le Google Apps, il pacchetto rivale di Office, rispetto al quale vanta meno attributi ma costa anche - la Premier edition, quella Standard è gratuita e ha all'attivo circa 10 milioni di utenti - solo 50 dollari di licenza all'anno.

Anche in questo caso si tratta un numero importante, seppur infinitamente inferiore a quello che può rivendicare Microsoft con il suo Office.

Ed è importante per vari motivi.

Perché fra gli utenti ci sono realtà come la Valeo (settore auto motive) con 32 mila sottoscrizioni e la District of Columbia (38mila dipendenti), oltre alle numerosissime università che hanno sposato il verbo di Google Docs.

E perché porta nelle case di Mountain View utili e qualche milione di dollari (quattro nel 2007, contro i 12,2 miliardi di dollari generati da Office) da investire in nuovi prodotti.

Come il browser Chrome, altra freccia - al pari di Google Gears e Google Friend Connect - con il quale armare un arco che potrebbe dare dolori al colosso di Redmond all'insegna di una piattaforma operativa Web based completamente open source.

Che dovrebbe essere la manna degli sviluppatori, anche di quelli che Microsoft vuole a tutti i costi tenere agganciati alla propria galassia.

11 milioni di utenti!

Un milione “business” e dieci “standard” sono un numero impressionante per una applicazione SaaS, specie se confrontati con i 43.600 di Salesforce.com, che viene considerata la soluzione SaaS numero 1.

Office e CRM sono due applicazioni diverse?

Per l’utente si, ma da un punto di vista di uno sviluppatore no : fanno cose diverse, ma per chi le ha sviluppate sono migliaia di righe di codice che un utente può usare per i propri scopi.

Sono righe di codice che gestiscono un data base, l’applicazione vera e propria e le maschere per imputare i dati.

Una architettura a 3 livelli costituita dal data base server, l’application server e il web server.

Per Microsoft è l’architettura .Net, coi suoi linguaggi Visual Basic e C#; per Google è l’architettura Open Source col linguaggio Java.

Ma Java non è utilizzata solo da Google e dai ragazzi dell’universo Open Source, è anche il linguaggio di riferimento dei “nemici” di Microsoft : IBM, Oracle, Sun, Bea, Salesforce, Novell,tutti insieme appassionatamente contro Microsoft che prevede un sistema operativo (Windows) su ogni dispositivo.

Per Google il sistema operativo vero e proprio è il browser, ecco perché tanto “casino” all’uscita di Chrome che si presuppone verrà usato da quei 11 milioni di utenti già utilizzatori di applicazioni Google.

A questo punto su Chrome potranno “girare” applicazioni che non gireranno così bene su Explorer e la guerra diventerà totale : o con me o contro di me!

Piattaforma mobile

La Guerra delle piattaforme (sistemi operativi) mobili è la più cruenta di tutte le guerre : Microsoft con Windows Mobile e Google con Andrid, per non parlare di iPhone OS della Apple, Symbian che appartiene ad un gruppo di aziende (Nokia, Ericsson, Panasonic, Siemens AG, Samsung e Sony Ericsson) ed infine SavaJe Os il sistema operativo basato su tecnologia Java e utilizzato da LG Elettronics.

All'inizio di settembre è in vendita per 719 Euro nel negozio online di Nokia il suo N96 : display da 2,8 pollici, altoparlanti stereo 3D integrati, 16 Gbyte di memoria interna, modulo A-Gps con navigatore (Nokia Maps) e guida vocale, lettore multimediale, fotocamera da 5.0 megapixel e il chipset Dvb-H per ricevere i canali della televisione digitale terrestre. L'accesso privilegiato ai servizi di Ovi fanno invece pendant con le capacità di comunicazione wireless (via reti Wi-Fi o 3G Hsdpa 3G).

Letta così assomiglia molto di più alla scheda tecnica di un mini computer da tasca che non a quella di un cellulare.

Stiamo perciò parlando di piattaforme che stanno cambiando le dimensioni, i contenuti e il modo stesso di usare i nuovi PC, che diventano degli Handset.

In questo settore la novità più grande è Android : una piattaforma open source per telefoni cellulari e handset, basata sul sistema operativo Linux e sviluppata dall'Open Handset Alliance (abbreviato OHA, è un accordo di differenti compagnie con Google come capofila, HTC, Intel, Motorola, Qualcomm, T-Mobile, e NVIDIA il cui obiettivo è sviluppare "open standards" per dispositivi mobili.).

La piattaforma è basata sul sistema operativo derivato da Linux con una serie di librerie dedicate come il database SQLite o SGL e OpenGL dedicate alla grafica, un "application framework", la Dalvik virtual machine (una Java virtual machine modificata) come ambiente di runtime e una serie di applicazioni preinstallate come un browser, una rubrica o un calendario.

A dimostrazione che sono visti come dispositivi per navigare e lavorare appena è uscito l'iPhone aziende come Salesforce.com e Netsuite.com si sono affrettate a dare la notizia che la loro applicazione SaaS funzionava perfettamente su questo telefonino della Apple.

Per gli sviluppatori di applicazioni su dispositivi “mobili” l'unico modo di raggiungere tutto il mercato è di pensare in modo “cross-platform” e sfortunatamente l'unico possibilità è il web browser. Questo è l'unica “piattaforma” che è disponibile su tutti gli smartphome e gli handset.

Perché “sfortunatamente”? Perché l'applicazione funziona bene solo se la connessione è disponibile e veloce e questo dipende dall'operatore del network.

Questo è ciò che angoscia gli sviluppatori di applicazioni mobili. Come risultato, se vogliono avere successo nel mercato mobile, debbono scrivere le applicazioni in modo che possano essere scaricate e memorizzate localmente sui dispositivi handset, fuori dai web browser.

Queste applicazioni debbono essere sviluppate specificatamente per i vari dispositivi. Dal punto di vista degli sviluppatori dover mantenere molte versioni della stessa applicazione è un incubo.

La fine dell'incubo

La combinazione 1-2-3 di Google con Gears, Android e Chrome è un messaggio agli sviluppatori web : l'incubo è finito, si possono sviluppare applicazioni web in grado di fornire accesso off-line ad alcuni servizi che solitamente funzionano on-line.

Gears è un'altra tecnologia open source di Google che, per le applicazioni web che la supportano, fa in modo che il web browser pensi di avere una connessione anche quando non c'è (ad esempio in aereo).

Senza una connessione molte applicazioni web (ad esempio la web e-mail) smettono di funzionare perché non riescono a scambiare dati e istruzioni col web server.

Gears installa un database engine, basato su SQLite, sul sistema del client per archiviare in locale i dati. Le pagine così abilitate da Google Gears utilizzano i dati della cache locale al posto dei servizi on-line.

Usando Google Gears, un'applicazione web può periodicamente sincronizzare i dati nella cache locale con il servizio on-line. Se non è disponibile una connessione in rete, la sincronizzazione sarà differita fino a quando la connessione non verrà stabilita nuovamente. Quindi Google Gears consente alle applicazioni web di lavorare anche quando l'accesso ai servizi della rete è assente.

Android, come tutti sanno, è il sistema operativo open source per gli handset che Google sta lanciando sul mercato. Si pensa che il primo Android handset -- HTC Dream -- sia disponibile in USA il 23 settembre 2008 e che l'operatore del network sia T-Mobile.

Chrome è il nome del web browser lanciato da Google in settembre. Per il momento è disponibile solo per Windows ma presto sarà utilizzabile su Mac, Linux e ovviamente su Android.

Il browser di Android è costruito con la stessa tecnologia di Chrome ma disegnato per gli schermi più piccoli degli Handset.

Il numero delle applicazioni web che supportano Gears sta silenziosamente crescendo (a metà settembre sembra che siano > 1.700 e saranno esposti e venduti in Android Market).

Zoho è giusto un esempio di un competitor di Google, nel mondo delle suite di office nel web, che usa Gears così che i clienti Zoho possono accedere alle sue applicazioni anche quando non sono connessi ai web server di Zoho. In settembre MySpace ha annunciato che offrirà l'accesso offline ai suoi membri usando Google Gears.

In verità il problema dell'offline è uno degli argomenti chiave delle applicazioni installate localmente come Microsoft Office, contro le applicazioni web come Google Apps.

Gears è riconosciuta come la tecnologia che fa cambiare il gioco.

Nel mondo dei desktop e dei PC notebook la combinazione di browser, di applicazioni basate sul browser e di una tecnologia come Gears è una possibilità credibile di affrancarsi da Windows e MS Office e Steve Bass di PC World [ha scritto](#) un articolo "Chrome: Google's Windows Killer" : "Questo è un diretto attacco a Microsoft, e io penso che Microsoft sia preoccupato. Questo perchè un piccolo kernel sul tuo sistema locale ti può permettere di lavorare senza Windows".

Senza Windows significa anche senza Microsoft Office.

Ma la sola esistenza di una tecnologia che potenzialmente può cambiare il gioco, come Gear, non è sufficiente per cambiare davvero il gioco.

Sono gli utenti a decidere, se hanno la percezione che la tecnologia è migliore la usano e decretano il successo.

Gear non è una applicazione, è qualcosa di non strutturato, quando Gears è in uso lavora in modo silente dietro le quinte, dentro la "trama del tessuto", ma, sfortunatamente, non per default.

Dal sito ufficiale Google Gear "*General Info: Where can I use Gears? Gears works with a select group of sites that are specifically designed for compatibility. When you visit a site that's configured to work with Gears, you should see a message stating "The website below wants to use Gears." If you'd like to allow the use of Gears on this site, simply check the box next to "I trust this site. Allow it to use Gears." and click the Allow button.*

Note: Once you've allowed a site to use Gears, you won't see this message again on that site."



Gear vuole solo far girare queste applicazioni in un certo modo (il modo offline) che, per alcune applicazioni come Google RSS Reader, che supporta Gear, non viene invocato a meno che l'utente lo invochi esplicitamente come nella figura a sinistra.

Ma la notizia peggiore (nei termini di mettere una tecnologia come Geas nella trama del tessuto) è quella che l'opzione di invocare il modo "Gears-driven offline" in una applicazione web come Google Reader non è disponibile senza che Gears sia installato localmente.

La trappola è che Gears non è la tecnologia più facile al mondo da utilizzare, oggi per avere le funzioni di Gears in uno dei tre browser più utilizzati (Explorer, FireFox e Safari) l'utente deve aggiungerlo di sua iniziativa : scaricarlo ed installarlo.

Concludendo : se Gears è installato allora le applicazioni che usano Gears avvisano che è possibile andare offline, se non è installato queste applicazioni non hanno possibilità di funzionare in modalità offline.

Nel mondo Google tutto questo è facilitato perchè quando scarichi Chrome installi anche Gears e sei sicuro che è installato bene e che tutto è coerente!

Però viene spontanea una domanda : perché Google non ha messo questa tecnologia "dentro" il suo Chrome, perché la rende disponibile per gli altri browser, come per il suo, come add-on?

Il perché è un fatto di marketing : deve dare a tutti gli utilizzatori delle sue Web Apps mobili, anche con altri browser la percezione che queste sono affidabili e che girano anche se non c'è connessione e questo senza obbligarli ad usare Chrome.

Chrome dà perciò la percezione di sicurezza perché Google è lo stesso fornitore del sistema operativo mobile, del browser e dell'add-on.

La percezione dell'utente

Google vince alla grande se l'utente percepisce che le applicazioni web mobile, con la sua tecnologia, sono più veloci e sicure.

Se poi riuscirà a far crescere il numero di coloro che hanno accesso alle Web Apps tramite un mobile browser, oltre agli 11 milioni già connessi, avrà stra-vinto.

Uno dei modi migliori per creare questa percezione è quello di spingere più sviluppatori di applicazioni mobili a sviluppare nel web invece che per una specifica piattaforma.

Come sappiamo questo è vincente per gli sviluppatori che possono così raggiungere un mercato più grande ed è vincente per gli utenti che non sono costretti ad acquistare una specifica piattaforma o network (ad esempio iPhone/AT&T) solo per aver accesso a certe applicazioni.

Google fornisce un "software development kit" per Android ma Google non ha interesse nel limitare il suo disegno solo ai telefonini e handset che utilizzano Android, deve continuare a dare agli utilizzatori la percezione di avere "scelto".

Per questo motivo Gears deve far parte del "tessuto" dei vari browser come i renderer HTML e i motori Javascript, per questo motivo Google ha reso Gears "open source" : per incoraggiare una adozione non forzata su qualsiasi altro browser.

Magari vi sono dei motivi tecnici per cui ha più senso che Gears sia un elemento fondamentale di un browser piuttosto che un add-on, si pensa infatti che così potrebbe offrire più stabilità e sicurezza, specie in un ambiente partizionato come Chrome.

Ma Chrome non permette solo di andare con qualche funzione offline.

In una intervista Larry Page, fondatore di Google, parlando di Chrome ha menzionato la necessità della velocità e in un ambiente mobile una tecnologia come Gears è in verità una "cache" e nel mondo della tecnologia una cache viene spesso associata con le performance.

Tecnologicamente parlando non c'è nessun motivo perché una tecnologia come Gears non possa lavorare in background tutto il tempo così che il browser possa trovare le cose che gli servono nella cache invece che nel lento e/o non connesso network.

Da adesso le due cose che interessano agli sviluppatori mobile, la velocità e le performance, sono tutte raggiungibili dalla presenza di una tecnologia standard di cache come Gears ed è importante che questa sia presente sulla maggioranza dei browser nei futuri handset.

In Android e nel suo “software development kit” tutto è pronto per sviluppare le prime applicazioni web mobile di nuova generazione.

PaaS (Platform as a Service)

Da Wikipedia “*Platform as a Service (PaaS) is an outgrowth of the Software as a Service application delivery model. The PaaS model makes all of the facilities required to support the end-to-end life cycle of building and delivering web applications and services entirely available from the Internet—with no software downloads or installation for developers, IT managers or end-users. It's also known as cloudware.*

PaaS offerings include workflow facilities for application design, application development, testing, deployment and hosting as well as application services such as team collaboration, web service integration and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation and developer community facilitation. These services are provisioned as an integrated solution over the web.”.

PaaS ha reinventato le funzioni tradizionali di sviluppo, realizzazione e distribuzione di qualsiasi applicazione aziendale offrendo una piattaforma come servizio, anche lei secondo la modalità SaaS.

Sviluppatori, clienti e partner possono affidarsi ad una piattaforma PaaS per creare con facilità una nuova generazione di applicazioni on-demand e renderle fruibili in tutto il mondo come servizio.

PaaS consente alle applicazioni di essere condivise, scambiate e installate con pochi click consentendo di trarre vantaggio dalle innovazioni della piattaforma.

Sviluppare e distribuire una applicazione SaaS non è facile : bisogna che sia sicura, affidabile, riservata, scalabile e facilmente utilizzabile. Poi bisogna darla ai clienti, sia in fase di test che di acquisto definitivo, bisogna fatturarla, bisogna vendere moduli aggiuntivi, bisogna fare i back-up, bisogna gestire i realease, bisogna permettere le integrazioni, bisogna cioè “gestire una piattaforma” che è una cosa più complessa di quella dello sviluppo di una applicazione.

Perciò una PaaS deve contenere :

1. servizi per lo sviluppo, test, distribuzione, host e manutenzione delle applicazioni
2. tools per creare delle interfacce utente disponibili nel web
3. una architettura multi-tenant
4. integrazioni con data base e web services
5. supporto per lo sviluppo in collaborazione fra team diversi
6. modelli di fatturazione

Tutte queste funzioni possono essere fatte in casa ma costano, sia in termini di sviluppo che di manutenzione, moltissimo.

Siamo così arrivati alle applicazioni SaaS per sviluppare e gestire il business SaaS e molto facilmente accadrà quanto è successo durante la “corsa all’oro” : chi ha fatto i soldi non sono stati i cercatori (sviluppatori di applicazioni SaaS) ma chi ha fornito loro picconi, badili e setacci (PaaS).

Di PaaS per le applicazioni web ce ne sono ormai molte, la più famosa è Force.com l’On-Demand Platform di Salesforce.com.

Quella di Microsoft è in via di sviluppo : Microsoft xRM.

Quale PaaS per le applicazioni web mobile?

Il 29 maggio 2008 Google ha lanciato la sua prima developer conference a pagamento : Google I/O.

Il messaggio nella presentazione era intitolato “Client, Connectivity, and the Cloud” condotto da Vic Gundrotta (Google Engineering Vice President) che ha spiegato che l’intento principale di Google è permettere agli sviluppatori :

1. *Access the cloud more easily (with Google App Engine);*
2. *Leverage more power from the browser (with Google Web Toolkit);*
3. *Maintain pervasive connectivity in their applications (with Android and Gears).*

La PaaS per gli sviluppatori è Google App Engine : Google manterrà le loro applicazioni sui suoi server gratis e pagheranno solo quando l'applicazione supererà i 500 MB di memoria persistente o la CPU e la banda supereranno quella necessaria a vedere 5 milioni di pagine al mese.

Google ha annunciato la possibilità di iscriversi a tutti gli sviluppatori interessati che dovranno, a questo punto, solo sviluppare la loro web applicazione sulle loro macchine locali, scaricarla su Google e consegnare la URL all'utente per l'utilizzo.

Oggi l'App Engine Runtime usa solo Python (*), se uno non è un programmatore Python o impara questo linguaggio o aspetta nuovi release con altri linguaggi e runtime.

Altre mobile PaaS : Yahoo! BrowserPlus RIA (Rich Internet Application) e la Adobe AIR development platforms.

Da eWeek.com “*BrowserPlus is designed to be run from the browser, while AIR lets users download Web-powered applications and run them on their desktop. Currently, AIR provides a complete runtime of APIs, analogous to an operating system or virtual runtime. BrowserPlus also differentiates from Gears in the flexibility quotient because users grab Gears in one big download, while BrowserPlus calls Web services as Web pages need them. Gears also runs in the browser space, which can cause browser crashes. BrowserPlus runs mostly in its own process space to ensure more stability.*”.

(*) **Da Wikipedia** : Python è un potente linguaggio di programmazione interpretato creato da Guido van Rossum. Prende il nome dalla serie televisiva *Monty Python's Flying Circus*. Python è un linguaggio multi-paradigma. Infatti permette in modo agevole di scrivere programmi seguendo il paradigma object oriented, oppure la programmazione strutturata, oppure la programmazione funzionale. Il controllo dei tipi è forte (strong typing) e viene eseguito runtime (dynamic typing). Usa un garbage collector per la gestione automatica della memoria. La grande quantità di librerie disponibili e la facilità con cui questo linguaggio permette di scrivere software modulare favoriscono anche lo sviluppo di applicazioni molto complesse.

L'autore

Edoardo Bolzani è stato un tecnico IT e un amministratore di aziende IT di successo. Dal 1998 si occupa di Applicazioni nel Web, di Web marketing e di e-Business. Nel 2001 ha portato Salesforce.com in Italia.

Appassionato di SaaS gestisce il sito www.saasitaly.com e fa consulenze per le aziende SaaS più innovative sul mercato nazionale.

e.bolzani@gmail.com