

IT TRANSFORMATION FLY TOGETHER

Gianni Canal, Marco Daccò, Simone De Rose, Giuseppe Parlati

La trasformazione digitale di TIM passa attraverso il ridisegno della Customer Experience ed una nuova architettura a microservizi. La prima rappresenta la finestra dalla quale i clienti percepiscono la trasformazione che sta avvenendo, la seconda rappresenta il cuore della trasformazione, il passaggio da un IT basato sui processi ad uno orientato ai dati, un nuovo IT future proof agile, flessibile, scalabile e “aperto” all’ecosistema digitale.

La Digital Service Delivery Platform

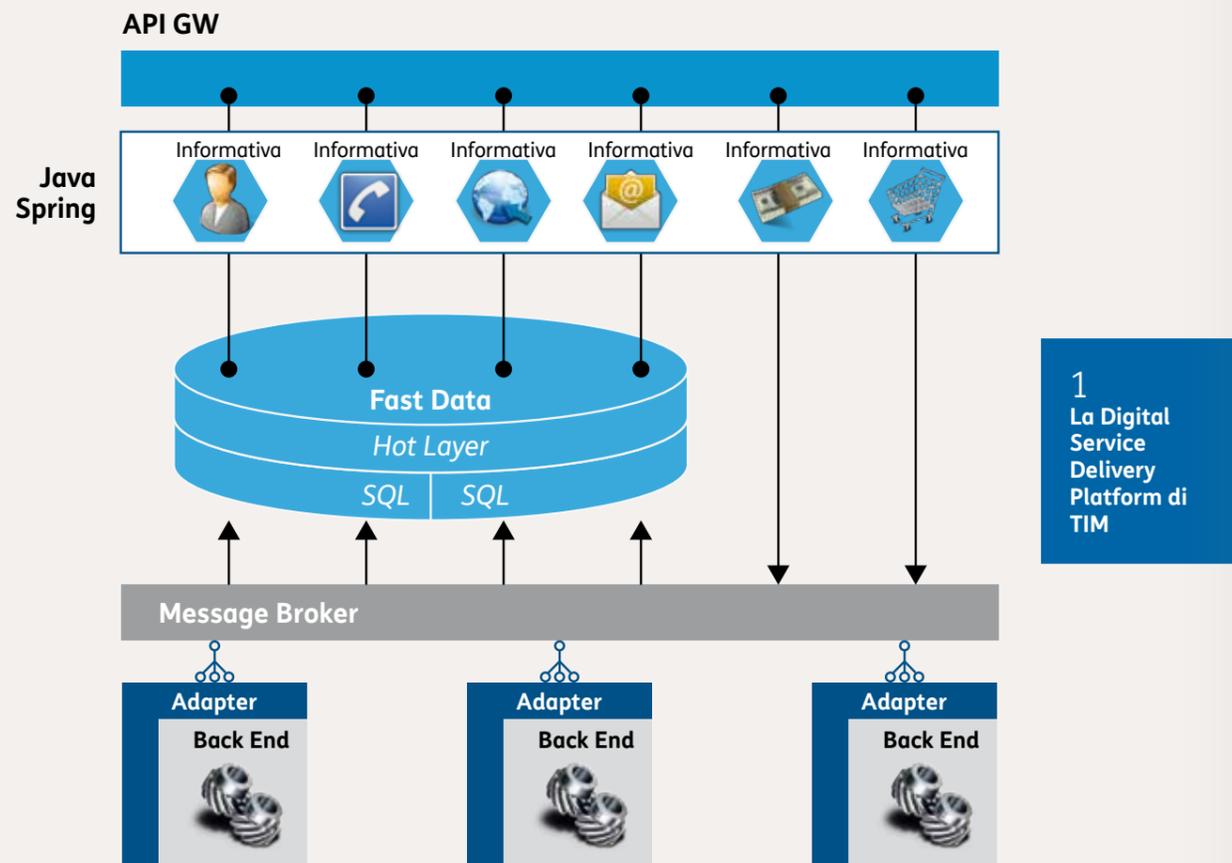
TIM ha avviato una trasformazione del dominio BSS per abilitare, in modo flessibile ed efficace, l’evoluzione del modello di business in linea con gli obiettivi di commercializzazione e gestione dei servizi Digitali. Per raggiungere tale obiettivo è stata identificata l’architettura

target della Digital Service Delivery Platform (SDP) che ha lo scopo di:

- Massimizzare la Customer Experience verso i canali digitali (Portali Web e App) – cfr paragrafo 2
- Ridurre il time to market per supportare l’ampliamento del portafoglio di offerta con nuovi servizi digitali

- Supportare Modelli di Business nuovi: Full Digital Provider, API Economy, Platform economy
- Abilitare la Data Monetization attraverso l’interazione con un ecosistema di terze parti

Il modello Digital Service Delivery Platform (SDP) si poggia su alcuni pilastri fondamentali descritti di seguito.



Performance, scalabilità e resilienza

Si rende necessario disporre di una infrastruttura che garantisca le alte performance (transazioni sub-second) tipiche dei contesti digitali, ma soprattutto di gestire picchi elevati di richieste in modo immediato ed adattivo per poter rispondere a comportamenti non sempre prevedibili tipici dei contesti web (es. fenomeni virali oppure adesioni massive a offerte last minute). Per questo, nell'ambito del programma FLY TOGETHER, TIM ha avviato la rea-

lizzazione di una "Digital Platform" basata su tecnologie come Paas e Container che supportano e garantiscono scalabilità orizzontale e su soluzioni di Inmemory Database capaci di garantire elevatissime performance di accesso ai dati anche in condizioni di accessi concorrenti multipli (FAST DATA). Il Fast Data nasce per esporre ai canali le entità di business di cui sono owner i componenti Legacy tramite API REST, sgravandoli dagli accessi in consultazione.

La sua infrastruttura si basa su uno stack tecnologico eterogeneo che

indirizza in modo puntuale i requisiti prestazionali nella gestione dei dati di business:

- un primo layer è dedicato a quello che sono i dati HOT che adotta una soluzione InMemory Grid per garantire accessi dell'ordine dei millisecondi e latenza tra produzione e consumo a "zero latency";
- un secondo layer è invece dedicato alla tipologia di dati WARM per i quali le prestazioni minime possono essere anche dell'ordine di una decina di millisecondi che utilizza sia un Database re-

lazionale (SQL) sia uno Document Based (NSQL).

Decoupling

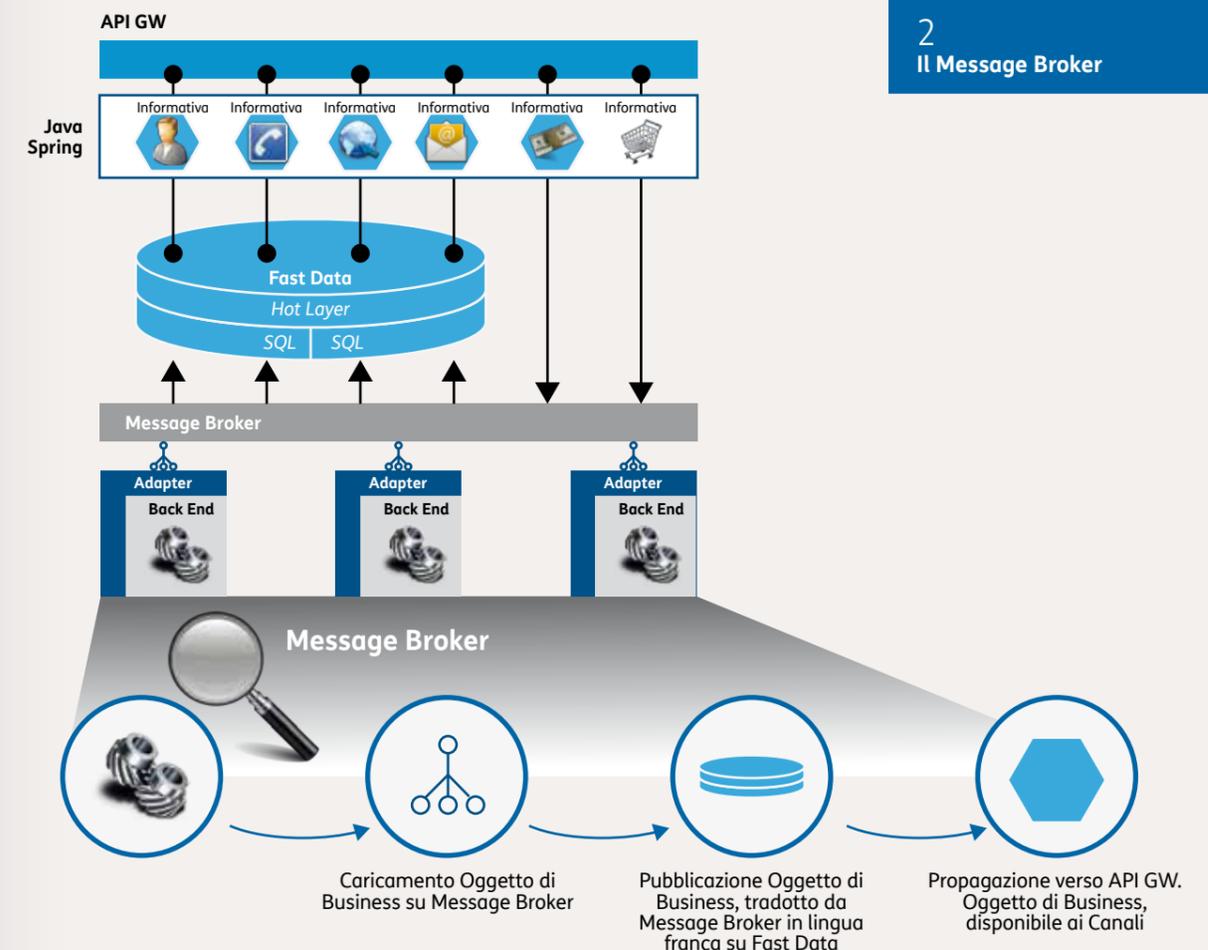
Il secondo pattern utilizzato nella Digital Platform di TIM implementa il concetto di **disaccoppiamento/decoupling** dalla infrastruttura legacy BSS attraverso un paradigma di tipo "coregrafato". Al fine di non avere sul percorso critico la vecchia infrastruttura BSS nell'accesso ai dati di business, si è adottato un modello di cache applicativa che viene alimentata attraverso

un meccanismo di code (**Message Broker**) dai vari sistemi legacy owner del dato. Ogni volta che su un sistema CRM viene aggiornato un dato del cliente oppure su un sistema di fatturazione viene prodotta una fattura, tale evento viene propagato verso la cache applicativa (Inmemory) che poi lo mette a disposizione dei canali consumatori (web, app, terze parti) con prestazioni superiori a qualsiasi infrastruttura tradizionale. Questo meccanismo serve sia le transazioni di tipo "informativo" sia quelle di tipo "dispositivo", ossia quelle in cui si inviano i comandi di "scrittura" verso il back end.

Microservizi

Un ulteriore elemento innovativo è l'adozione di un modello di sviluppo basato sui **microservizi**.

I microservizi sono delle applicazioni software "piccole" ed autonome che interagiscono tra di loro e che hanno come finalità quella di fare una "sola" cosa e di farla bene; sono a tutti gli effetti dei sistemi distribuiti. Martin Fowler, padre di questa metodologia/stile di coding afferma che "Lo stile architetturale a microservizi è un approccio allo sviluppo di una singola applicazione come insieme di piccoli servizi, ciascuno dei quali viene eseguito da un proprio proces-



IL CATALOGO A BUILDING BLOCK

Marilena Di Berardino, Vittorio Panariello

Tra i servizi principali della piattaforma Digital c'è il Catalogo TIM strutturato a Building Block che contengono gli elementi base delle offerte che si vogliono vendere ed i pattern di vendita (regole di gestione). Questi possono essere combinati per comporre il tipo di offerta e di servizio con relative regole tecniche.

I principali vantaggi dell'utilizzo di un catalogo strutturato a Building Block sono:

- potenziamento della vendita su più canali (multicanalità);
- riduzione del time to market (da mesi a pochi giorni) per il lancio di offerte, personalizzazioni e pacchetti creativi senza impatto sui sistemi di backend;
- configurazione di nuove offerte direttamente dal Marketing.

Inoltre il catalogo TIM ha un motore a regole flessibile con il quale si possono configurare i modelli di pricing e le leve commerciali (ad esempio promozioni, sconti, offerte 3x2); questo permette di avere un tailoring delle offerte su misura del cliente.

Infine la componente Carrello fornisce un'esperienza di acquisto evoluta attraverso lo strumento della multicanalità, potendo servire contemporaneamente canali attended e unattended, con la possibilità di far selezionare offerte, aggiungere servizi e prodotti e poi procedere all'acquisto utilizzando i più comuni metodi di pagamento ■



ANATOMIA DI UNA API

Elisabetta Persello, Domenico Colia

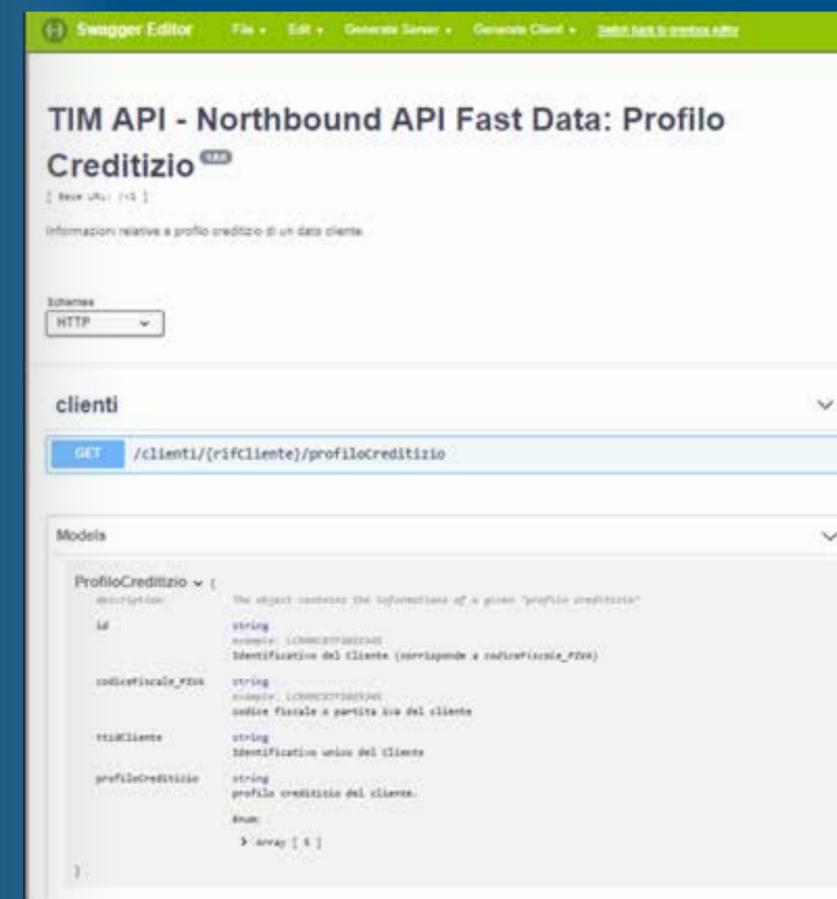
TIM avvia la trasformazione digitale attraverso la realizzazione di una nuova piattaforma capace di abilitare innovativi scenari di business. Il cuore della piattaforma è costituito da un'architettura a microservizi invocabili tramite delle API RESTful, la lingua franca di internet, e disponibile ai diversi canali ed in futuro anche ai propri partner commerciali.

Quale miglior modo di progettare le API RESTful con un framework opensource come Swagger?

Realizzato dall'azienda americana Smartbear e rilasciato con licenza Apache2.0,

Swagger è diventato in breve tempo uno standard de facto del mercato. Swagger mette a disposizione un ampio insieme di strumenti che accompagna l'intero ciclo di vita delle API, dalla progettazione, alla documentazione al test e al deploy, diffondendo la conoscenza delle API, e accelerando drasticamente lo sviluppo dei diversi team. Le specifiche Swagger, grazie alla vasta adozione da parte del mercato, sono evolute nelle OpenAPI Specification 3.0, adottate e sponsorizzate da varie aziende quali Google, IBM, Microsoft e la stessa Smartbear.

La trasformazione digitale passa anche da qui, razionalizzando e riorganizzando il proprio business ■



IL MONITORAGGIO DEI MICROSERVIZI

Francesco Fusco, Patrizia Pisano

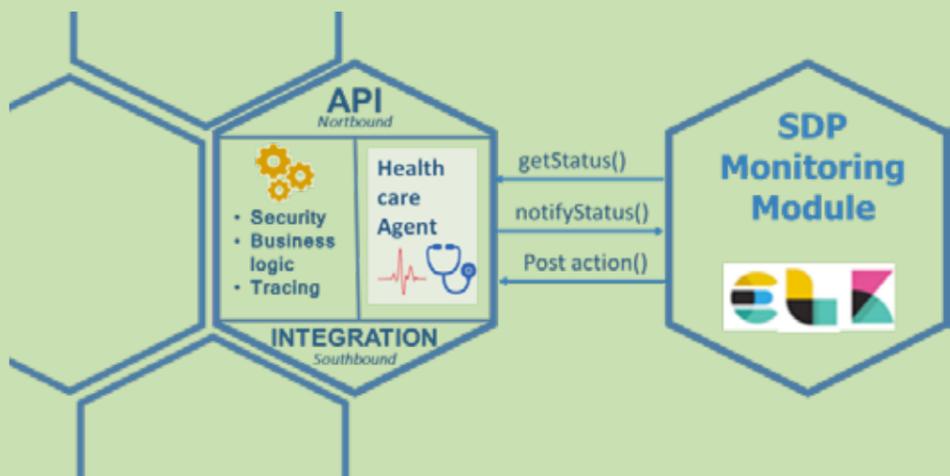
Per verificare lo stato di salute della Service Delivery Platform e consentire alle funzioni di GA di intervenire prontamente sia per risolvere guasti che per incrementare le prestazioni del sistema in caso di necessità, è stato progettato un modulo funzionale di monitoraggio che controlla costantemente lo stato di salute di tutte le componenti. Tale modulo è basato su prodotti open source, ampiamente descritti in letteratura e largamente utilizzati: lo stack tecnologico ELK costituito da un database NoSql (Elasticsearch), un collettore real time (Logstash) e un motore grafico (Kibana). E' un componente funzionale preso oramai come riferimento dai più importanti player di tecnologie digitali; diventato uno standard de facto per chi non vuole abbracciare soluzioni basate completamente o in parte su prodotti commerciali.

Specifici "agent" attivati sui server di SDP, ogni giorno scambiano con il modulo di monitoraggio, centinaia

di milioni di informazioni riguardanti il funzionamento delle componenti hardware e software come memoria, cpu, stato delle code del message broker, stato dei processi e dei microservizi, etc. Tali dati subiscono processi di elaborazione real time (filtraggio, normalizzazione, correlazione e aggregazione) e vengono memorizzati sul database Elasticsearch, un database specializzato per ricerche testuali.

Grazie al motore grafico di Kibana, il modulo di monitoring è stato arricchito con dashboard specifiche per visualizzare le metriche ed i KPI definiti per SDP. Tali dashboard si aggiornano di continuo per fornire costantemente la fotografia dello stato corrente di SDP. Una componente di alerting genera eventuali allarmi e li notifica agli operatori di esercizio via email ed sms. In futuro tali notifiche verranno inviate su una mobile app dedicata, al fine di consentire una maggiore interazione uomo-macchina.

I dati elaborati dal modulo di monitoraggio di SDP serviranno inoltre a ricostruire una visione più ampia della corretta erogazione dei servizi di business in cui SDP è coinvolta e questo verrà effettuato attraverso la standardizzazione di tracciati record compliant con gli standard più recenti (es. openTracing) e l'integrazione con piattaforme di monitoraggio dedicate all'E2E ■



so e comunica con un meccanismo snello, spesso una HTTP API". I vantaggi rispetto ad una architettura tradizionale (che potremmo definire Monolitica) sono evidenti:

- Si possono utilizzare **stack tecnologici eterogenei**. Non si è vincolati a fare tutto con la stessa tecnologia, ma liberi di scegliere di volta in volta la tecnologia migliore per la soluzione che si sta approntando.
- Favoriscono la **resilienza**: Se un componente della catena fallisce, resta un problema isolato, che non interviene a cascata su tutto il sistema. E abbiamo già parlato di quanto la resilienza sia un aspetto chiave.
- Favoriscono la **scalabilità**: con tanti servizi piccoli, è facile intervenire solo su alcuni di essi, senza necessariamente modificare nulla degli altri.
- Sono **semplici da rilasciare**: essendo isolati, è facile rilasciare

un singolo microservizio, senza dover interrompere tutti gli altri allo stesso tempo (no effetto domino).

- Si basano sulla **componibilità**: è un po' come giocare con il Lego. Possiamo ampliare la nostra soluzione un po' alla volta, senza dover rilasciare tutto subito e quindi sono particolarmente adatti ai modelli di sviluppo software incrementale.

Risulta evidente che la loro "atomicità" e autoconsistenza diventano elemento di enorme flessibilità e agilità rispetto ad architetture monolitiche tradizionali per le quali l'unico modo di poter scalare un'applicazione era quello di replicare l'intera applicazione con conseguente aumento di costi e risorse necessarie.

Tutte queste caratteristiche sono state un fattore differenziante nelle architetture di successo di alcuni dei principali Over The Top (OTT) mon-

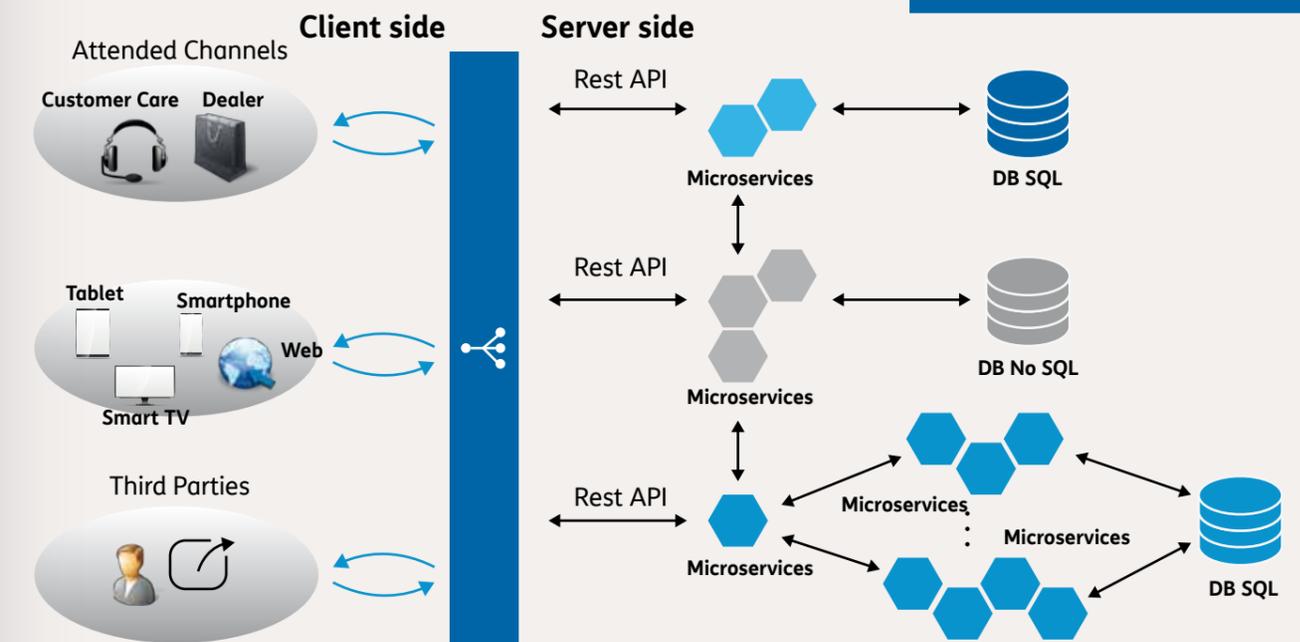
diali come Netflix, Amazon, Spotify e Zalando.

Organizzando quindi la propria piattaforma secondo un modello Microservices si disporrà di un elevato numero di servizi che erogano il proprio compito attraverso il loro contratto pubblico le API application Programming Interface.

Trovandosi di fronte a un numero decisamente elevato di API/Contratti la SDP di TIM è stata dotata di 2 componenti:

- il **Composite Layer**, ossia uno stack software che ha il compito di realizzare "servizi complessi" tipicamente orchestrando i singoli microservizi elementari offerti dalla piattaforma;
- l'**Exposure Layer** di tutte le API che vengono offerte ai canali consumatori.

3 Esempio di Microservizi



L'approccio Digitale alla Customer Experience

I casi di successo di aziende come Amazon, Netflix, Google, Starbucks, Disney mostrano un elemento comune rivelatosi fondamentale nella trasformazione: l'ossessione per i clienti. L'Esperienza Digitale che i clienti vivono quando si relazionano con la nostra azienda rappresenta l'unica finestra dalla quale percepire la trasformazione che sta avvenendo.

La Digital Experience (DX), oggi rappresenta il reale vantaggio competitivo dell'azienda, che non si riconosce più solo nel prodotto o nel prezzo; tutte le espressioni della DX devono essere realizzate all'insegna della Customer Centricity:

- **Customer Engagement**, sono le interazioni rilevanti e coinvolgenti nell'intero ciclo di vita del cliente. Essa deve essere innovativa e basata su una conoscenza assoluta del cliente

digitale ed always-on, ovvero vive nella «omni-canalità» ed è digitalmente e socialmente ovunque. Un Engagement interattivo e predittivo permette di impostare Campagne ed offerte ad hoc che permettono di coinvolgere il cliente e creare una relazione basata sull'interazione costante.

- **Customer Experience**, è “il modo in cui i clienti percepiscono l'interazione con la vostra azienda” [Forrester Research]. Non è legata al “cosa” ma al “come” e non si limita tutte le potenzialità dei device per offrire una esperienza multi-sensoriale (es microfono, audio, lettore impronte digitali, riconoscimento facciale ...) e le tecnologie innovative ormai mature e disponibili come l'AI per rispondere ma soprattutto per predire e la VR/AR per una Immersive Experience. Anche la personalizzazione è fondamentale per

cucire su misura una esperienza perfetta per il cliente; il cliente ormai se lo aspetta!

- **Customer Journey**, si intende l'itinerario che il cliente percorre nel tempo e nei diversi “ambientati” di contatto, siano essi offline che online. Quindi possiamo dire che rappresenta la “storia del legame” cliente-azienda. Esso ha l'obiettivo primario di fornire ai clienti uno Unique Journey basato su un Digital-Only OmniChannel.

Dietro questi concetti molto sofisticati si celano comunque dei principi base che sono fondamentali:

- **Semplicità**: Percorsi che rappresentano processi commerciali e spesso non tengono conto del fattore umano che dopo 3 click inizia a perdere interesse ed attenzione.
- **Intuitività**: i comportamenti richiesti agli utenti devono essere intuitivi, comportamenti digitali che devono assimilarsi ai comportamenti tipici della vita rea-

le. Ogni “sforzo” richiesto all'utente è una barriera alla fruibilità.

- **Personalizzazione**: ogni persona ha una sua esperienza e modalità di interagire. Una persona anziana ha una gestualità molto differente da un teenager. Standardizzare le User Experience per qualsiasi utente è

probabilmente più efficiente ma sicuramente meno efficace.

- **Gradevolezza**: sicuramente la grafica ha un peso importante nell'agevolare la fruizione di un servizio dotato di GUI. I colori stimolano sensazioni ed emozioni e subliminalmente possono migliorare l'accettabilità. (i.e. una famosa maionese in

Francia è gialla - dà l'idea che sia fatta con le uova - mentre in America è bianca - dà l'idea che non contenga grassi).

- **Gratifica**: La complessità genera il rischio che l'utente non si senta adeguato ad un suo utilizzo causandone un allontanamento ■



Gianni Canal gianni.canal@telecomitalia.it

laureato in Informatica, è entrato nel gruppo TIM alla fine del 1992 dove, in ambito CSELT, si è occupato nei primi anni di Rete Intelligente. Successivamente ha coordinato un team di innovazione sulle piattaforme di servizio nell'ambito della rete mobile. Negli ultimi anni è stato responsabile di un gruppo di ingegneria Smart Pipe con il compito di innovare la monetizzazione di dati e asset di rete fino a divenire responsabile di Software Development Center in ambito TILAB. Attualmente in ambito TIM Information Technology è responsabile della funzione IT Architecture-Applications ■



Marco Daccò marco.dacco@telecomitalia.it

laureato in ingegneria gestionale, vanta una esperienza di 20 anni nel settore ICT per la industry delle Telecomunicazioni. Ha iniziato la sua esperienza lavorativa in Mars inc. in ambito Divisione logistica. A metà anni novanta ha partecipato allo start up della prima joint venture tra Telecom Italia e Bell Atlantic (oggi Verizon) per la realizzazione dell'attuale polo ICT di Trento. Ultimamente è stato responsabile dello start up della funzione Vertical IT Solution di Telecom Italia per i segmenti di sanità elettronica, infomobilità e Tourism. Attualmente in ambito TIM Information Technology è responsabile della funzione Digital Services Delivery Platforms ■



Simone De Rose simone.derose@telecomitalia.it

laureato in Ingegneria Elettronica presso l'Università degli studi La Sapienza di Roma. Nella sua carriera professionale ha partecipato nel 1998 allo start-up di WND Telecomunicazioni e dal 1993 ha collaborato a progetti di Ericsson AB e Motorola Inc. in qualità di progettista software. Dal 2001 è stato Responsabile per TIM e Telecom Italia come nell'ambito dell'Information Technology per lo sviluppo e la manutenzione dei sistemi di CRM Consumer, Business, e Wholesale e dei sistemi di Billing&Credit Management. Ha ricoperto anche il ruolo di responsabile delle Architetture e successivamente del Client Management. Attualmente in ambito TIM Information Technology è responsabile della funzione Digital Services Solutions ■



Giuseppe Parlati giuseppe.parlati@telecomitalia.it

laureato in Scienze dell'Informazione, si è occupato di ricerca scientifica in ambito Algoritmi Paralleli e Distribuiti Fault-Tolerant presso l'Università di Salerno e la Columbia University (NY). Ha pubblicato articoli scientifici su riviste e conferenze internazionali.

Nel gruppo TIM dal 1997 si è occupato dapprima di Architetture Applicative & Middleware per poi unirsi al gruppo di innovazione VAS. Negli ultimi anni si è occupato di progetti di Digital Transformation e di Innovazione. Ha conseguito le certificazioni PMP e SCRUM MASTER I ■



Customer Experience per OVER

Customer Experience per UNDER

Customer Experience per BIZ

Customer Experience per YOUNG