

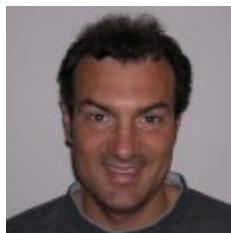
Accedere ad applicazioni z/OS tramite iPhone!

Fabio Riva
Certified Senior IT Architect
z/Series Technical Presales
IBM Software Group

Document Version: 2.5.3

Per commenti o richiesta di maggiori informazioni:

email: fabio.riva@it.ibm.com



Fabio Riva
IBM Certified IT Architect
z/Series SW Technical Sales



IBM Italia
Circonvallazione Idroscalo
20090 Segrate (Milano) Italy

Office: +39 02.596.29347
Fax: +39 02.596.29347
Mobile: +39 335 7368769
Email: fabio.riva@it.ibm.com

Sommario

| | |
|--|-----------|
| 1. INTRODUZIONE..... | 5 |
| 1.1 DA COSA NASCE QUESTO DOCUMENTO..... | 5 |
| 2. PRENDIAMO UN'APPLICAZIONE ESISTENTE | 6 |
| 2.1 DISEGNO ARCHITETTURALE DELL'APPLICAZIONE ESISTENTE..... | 6 |
| 2.2 PROVIAMO A UTILIZZARE L'APPLICAZIONE | 7 |
| 3. COME RENDERLA ACCESSIBILE ANCHE DA ALTRI AMBIENTI..... | 9 |
| 3.1 DISEGNO ARCHITETTURALE DELL'APPLICAZIONE CON MODIFICHE | 9 |
| 3.2 ACCEDIAMO DA UN COMUNE PC CON INTERNET EXPLORER..... | 10 |
| 4. ACCEDIAMO CON IPHONE | 11 |
| 4.1 PRIMO PASSO: REPERIMENTO DEL DEVICE. | 11 |
| 4.2 SECONDO PASSO: CONTROLLO FIRMWARE | 12 |
| 4.3 TERZO PASSO: SETUP DI RETE/COLLEGAMENTO WiFi..... | 12 |
| 4.4 QUARTO PASSO: COLLEGAMENTO AL WEB SERVICE | 13 |
| 4.5 QUINTO PASSO: OTTENIMENTO DEI DATI | 13 |
| 5. FACCIAMO UN'ALTRA DEMO..... | 14 |
| 5.1 ACCEDIAMO AL SERVER TCIM | 15 |
| 6. CONCLUSIONE..... | 17 |

1. Introduzione

1.1 Da cosa nasce questo documento

Negli ultimi anni la tendenza che si credeva ormai inesorabile verso la sostituzione dei sistemi mainframe con sistemi dipartimentali (conosciuti anche come *'informatica distribuita'*) si è interrotta, anzi potremmo dire che i segnali di inversione della tendenza sono sempre più forti. Nell'era di internet, del social networking, del consolidamento e della virtualizzazione, si è scoperto che i mainframe sono in fondo le piattaforme più economiche, sicure, scalabili ed affidabili. Ci si è resi conto, inoltre, che i cari vecchi mainframe consumano meno energia rispetto ai loro discendenti, ossia che a parità di lavoro svolto, il vecchio dinosauro divora molto meno di uno stuolo di PC in parallelo.

Gli sforzi che sono stati fatti nella piattaforma mainframe per l'apertura al software open, al mondo Unix, all'integrazione con reti e dati non proprietari, non credo abbiano eguali nel campo dell'informatica. Chi avrebbe mai immaginato una ventina di anni fa che il sistema operativo MVS potesse essere certificato Unix? Eppure oggi le applicazioni standard Unix possono girare nativamente su z/OS, anzi la componente Unix è stata integrata nella base del sistema operativo. Alcune parti dello stesso sistema operativo sono addirittura Unix (la parte TCP/IP del Communication Server, ad esempio).

Ebbene, nonostante questo sforzo, nonostante l'impegno e gli investimenti legati al rinnovamento fatto, ci sono ancora persone che ritengono questa macchina 'superata' o 'vecchia'. Che parlano ancora solo di video con fosfori verdi. Che non la prendono nemmeno in considerazione o che pensano che le nuove tecnologie informatiche non possano interagire con il mondo z/Series.

Ecco, la molla principale che mi ha spinto a scrivere questo documento è forse stata proprio questa. Dimostrare il contrario, far vedere che anche un nuovissimo gioiellino tecnologico, una cosa di tendenza, insomma un oggetto di culto popolare come l'iPhone sia in grado di interagire con una vecchia applicazione MVS. Un'applicazione magari scritta 30 anni fa in linguaggio Cobol. Sarebbe una bella cosa riuscire a fare questo, dimostrerebbe in modo concreto quanto vengono salvaguardati gli investimenti in area mainframe.

2. Prendiamo un'applicazione esistente

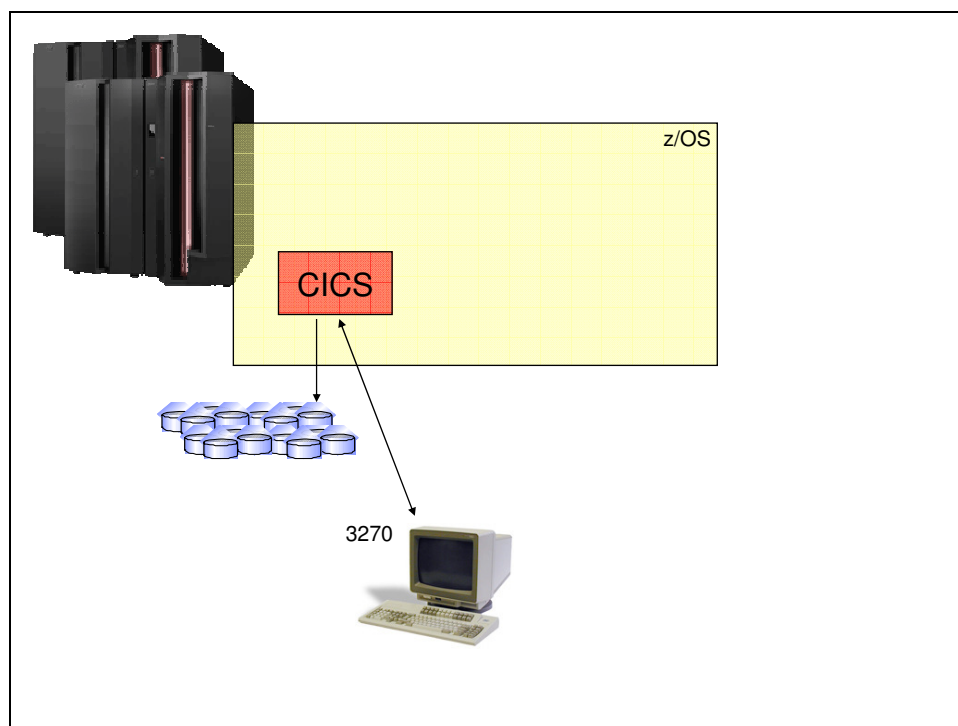
La prima cosa da fare per costruire una dimostrazione come quella che mi prefiggo di preparare consiste nella scelta dell'applicazione. Deve essere un'applicazione adatta allo scopo che abbiamo ipotizzato. Deve essere un'applicazione scritta in modo tradizionale, in modo che possiamo dimostrare la possibilità di rendere facilmente 'moderne' anche applicazioni datate.

Dopo un pò di ricerche ho trovato una cosa che può fare al caso mio: un'applicazione locale scritta dal mio collega Giancarlo Rochester, che viene da lui utilizzata anche per fare demo ai clienti. Si tratta di un'applicazione CICS basata su programmi scritti in Cobol, che interagiscono con una base dati VSAM. Direi che potrebbe esser un buon esempio per la nostra dimostrazione.

2.1 Disegno architetturale dell'applicazione esistente

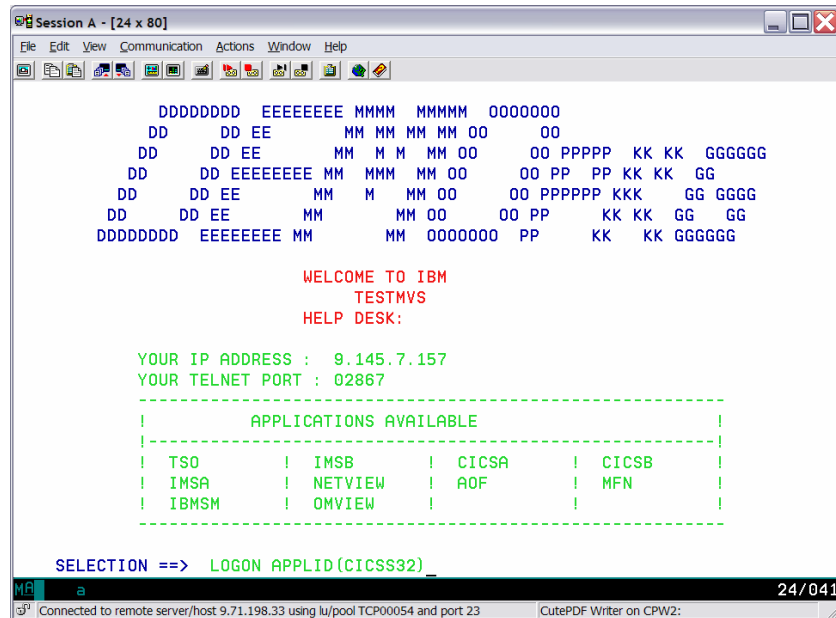
Di seguito allego uno schema architetturale (molto stilizzato) dell'applicazione e del contesto in cui viene utilizzata. Dato che la mia intenzione era di utilizzare un'applicazione vecchio stile per dimostrare la riusabilità degli asset informatici mainframe... mi sembra che sia proprio riuscito a trovare quello che cercavo. L'applicazione viene invocata da un terminale 3270, e una volta connessi al CICS si può lanciare la transazione che accederà alla base dati (dataset VSAM) e ci restituirà i valori di cui abbiamo bisogno.

La modalità di accesso utilizzata è proprio quella tipica della transazione 3270, infatti tutta la logica e' nel server mainframe z/OS, mentre il terminale mostra solo dati alfanumerici e serve solo per input/output. Il flusso di dati e' quello tipico del terminale 3270. E' proprio l'applicazione giusta per gli scopi che ci siamo prefissi.



2.2 Proviamo a utilizzare l'applicazione

Di seguito allego alcune schermate che dimostrano come il colloquio con l'applicazione CICS venga fatto in modalità 3270. Innanzi tutto facciamo partire l'emulatore software 3270 (di terminali coassiali non ne ho più trovati in giro...) che si collegherà via telnet 3270 al server z/OS (nel nostro caso, quello del centro GYM z/OS di Segrate). Per rendere più visibili le schermate, ho messo lo sfondo bianco alla schermata dell'emulatore 3270.



```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
DDDDDDDD EEEEEEEE MMMM MMMM 0000000
DD DD EE MM MM MM MM OO OO
DD DD EE MM M M MM OO OO PPPP KK KK GGGGG
DD DD EEEEEEEE MM MMM MM OO OO PP PP KK KK GG
DD DD EE MM M MM OO OO PPPPP KKK GG GGGG
DD DD EE MM MM OO OO PP KK KK GG GG
DDDDDDDD EEEEEEEE MM MM 0000000 PP KK KK GGGGG

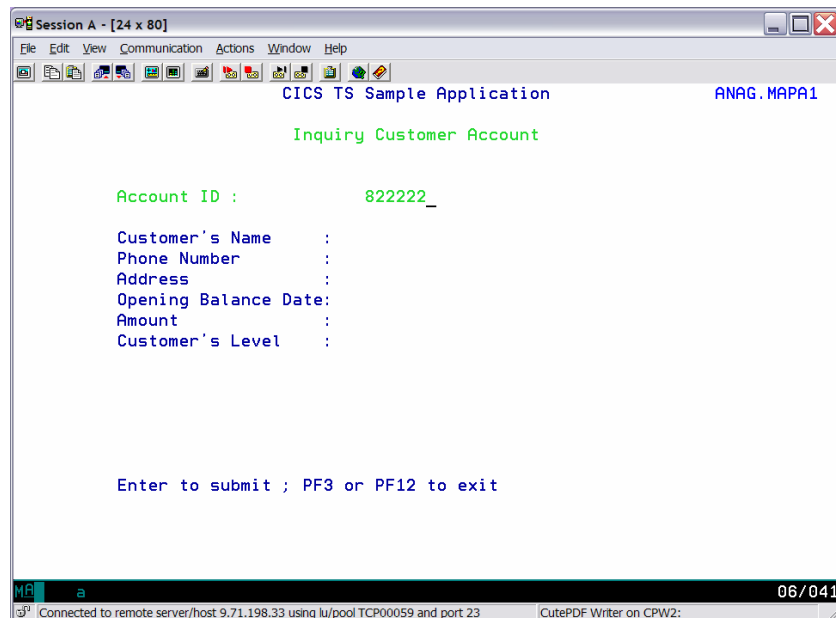
WELCOME TO IBM
TESTMVS
HELP DESK:

YOUR IP ADDRESS : 9.145.7.157
YOUR TELNET PORT : 02867

! APPLICATIONS AVAILABLE !
! TSO ! IMSB ! CICSA ! CICSB !
! IMSA ! NETVIEW ! ADF ! MFN !
! IBMSM ! OMVIEW ! ! !

SELECTION ==> LOGON APPLID(CICSS32)_
MA a 24/041
Connected to remote server/host 9.71.198.33 using lu/pool TCP00054 and port 23 CUTEPDF WRITER ON CPW2:
```

Una volta collegati al Communication Server dello z/OS, facciamo logon all'applicazione CICS tramite il comando di LOGON APPLID(...). Questo ci permette di essere in linea direttamente con il sottosistema CICS e poter invocare la transazione che a noi serve.



```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
CICS TS Sample Application ANAG.MAPA1
Inquiry Customer Account

Account ID : 822222_

Customer's Name :
Phone Number :
Address :
Opening Balance Date:
Amount :
Customer's Level :

Enter to submit ; PF3 or PF12 to exit

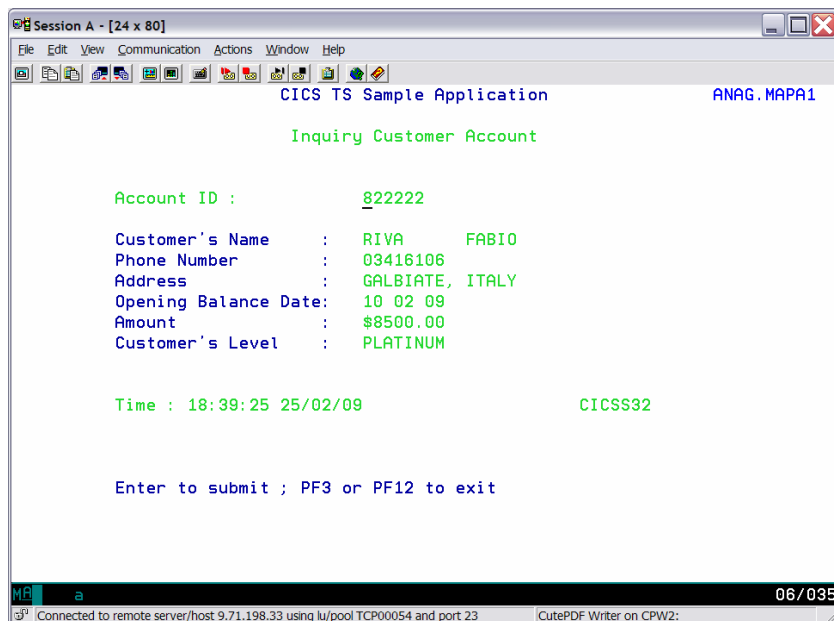
MA a 06/041
Connected to remote server/host 9.71.198.33 using lu/pool TCP00059 and port 23 CUTEPDF WRITER ON CPW2:
```

La transazione che a noi serve si chiama MILA. Questa transazione accede al dataset VSAM che, nel nostro caso, funge da base dati. In input alla schermata della transazione MILA si passa l'Account Id, e di ritorno vengono riempiti tutti i campi info relativi all'utente collegato all'account.

Accedere ad applicazioni z/OS... tramite iPhone!

Per compiere questa demo ho chiesto che mi venga definito un account, e mi è stato assegnato l'account id **822222**. Una volta sottomessa la transazione, il sistema dovrebbe ritornarmi i valori relativi al mio conto corrente, ossia **Customer's Name, Phone Number, Address, Opening Balance Date, Amount** e **Customer's Level**.

Ecco la schermata risultante dopo l'esecuzione della transazione MILA. Il sottosistema CICS ha recuperato le informazioni legate al mio codice di account id dalla base dati e ha riempito i campi.



Abbiamo dimostrato che l'applicazione tradizionale è esistente e utilizzabile. Prima di procedere con l'accesso tramite iPhone bisogna però fare ancora un pò di lavoro. Infatti allo stato attuale l'applicazione tradizionale può essere invocata direttamente solo con il protocollo 3270.

C'è ancora qualcosa da fare per 'modernizzare' la nostra vecchia applicazione. Possiamo utilizzare una tecnologia che ormai è consolidata e facilmente realizzabile... una tecnologia che è conosciuta con il nome di **web services**.

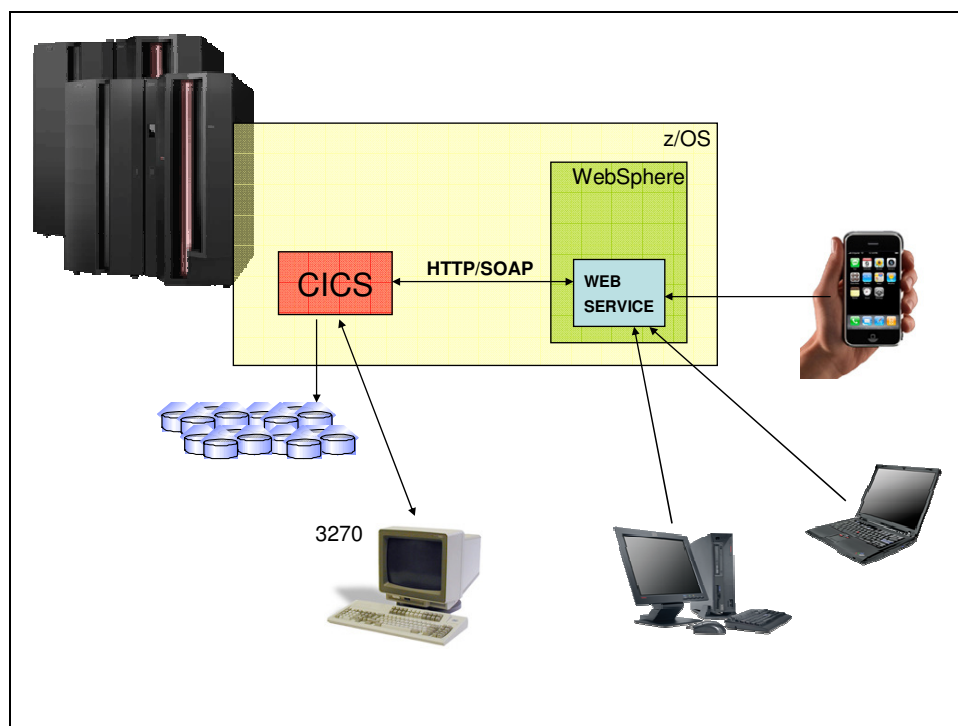
3. Come renderla accessibile anche da altri ambienti

Secondo quanto detto precedentemente, la tecnologia **Web Services** permetterebbe l'apertura della nostra applicazione al mondo esterno. Ma cosa sono i Web Services? Secondo la definizione fornita da Wikipedia (che è tratta a sua volta dalla definizione fornita dal World Wide Web Consortium W3C) un *Web Service* è un sistema software progettato per supportare l'interoperabilità tra diversi elaboratori su di una medesima rete.

L'uso di Web Services permette quindi di espandere enormemente l'accesso alla nostra applicazione tradizionale. Infatti tramite un'architettura basata sui Web Services è possibile far comunicare tra loro applicazioni software scritte in diversi linguaggi di programmazione, e addirittura implementate su diverse piattaforme hardware.

L'uso di Web Services permette di offrire un'interfaccia software (descritta in un formato standard come WSDL o Web Services Description Language) tramite la quale altri sistemi possono interagire con il Web Service stesso attivando le operazioni descritte nell'interfaccia tramite appositi "messaggi" inclusi in una "busta" (SOAP). Il protocollo di comunicazione è di solito il protocollo HTTP e i messaggi sono formattati secondo lo standard XML.

3.1 Disegno architetturale dell'applicazione con modifiche



Come si può vedere nella figura precedente, nello schema architetturale è stato introdotto uno strato software aggiuntivo che mi permetterà di 'esternalizzare' la mia applicazione. Anche in questo caso ho sfruttato l'ambiente demo del mio collega Giancarlo Rochester, infatti la struttura Web Services era già stata creata da lui in precedenza, quindi ho evitato di fare direttamente il lavoro di installazione e personalizzazione.

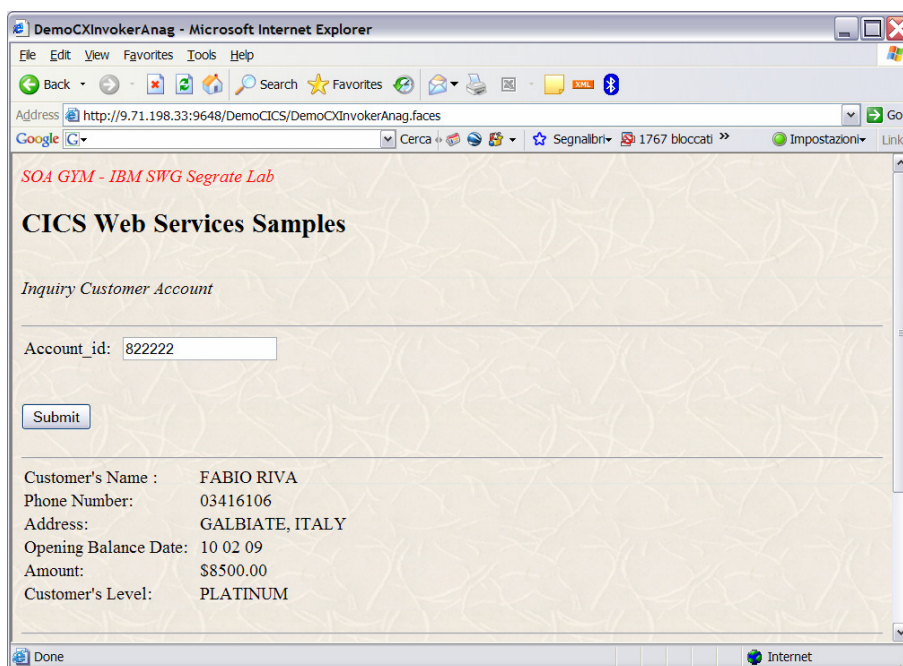
In questo mio documento tralascierò, per ovvi motivi di spazio, di trattare nel dettaglio le attività che sono state fatte per implementare questa ‘esternalizzazione’. Mi permetto però di citare due redbook che mi sono sembrati molto esaurienti sull’argomento e che spiegano con dovizia di particolari le possibilità offerte dai sottosistemi z/OS:

- *Integrating Back-end Systems with WebSphere Application Server on z/OS through Web Services (SG24-7548-00)*
- *Implementing CICS Web Services (SG24-7657-00)*

Tornando al nostro esempio, l’esternalizzazione mi permette di fare in modo che l’applicazione possa essere invocata anche in altri modi oltre al protocollo 3270, che resta in ogni caso attivo e funzionante. In aggiunta alla metodologia esistente, ora posso invocare l’applicazione tramite una *servlet* WebSphere che chiama l’applicazione CICS/Cobol e mi ritorna i dati richiesti.

3.2 Accediamo da un comune PC con internet explorer

Per come è stata realizzata, la servlet Web Service mi permette di chiamare l’applicazione CICS da un qualsiasi browser internet, tramite il protocollo HTTP. Questo rende l’accesso alla nostra applicazione *platform-independent*. Questo fatto può essere dimostrato facilmente accedendoci da un qualsiasi browser internet:



Con l’introduzione dei web services abbiamo aperto la strada per accedere alla nostra vecchia applicazione CICS/Cobol da un qualsiasi dispositivo che permetta il colloquio HTTP con browser internet... perciò questa modalità può essere usata anche con l’iPhone, tramite il browser Safari e il collegamento di rete WiFi. Dimostriamolo procedendo con la demo.

4. Accediamo con iPhone

Nel capitolo precedente abbiamo visto come fare per mettere a disposizione del mondo esterno le applicazioni pre-esistenti. Con le modifiche indicate, è ora possibile accedere all'applicazione CICS/Cobol tradizionale da un qualsiasi browser. Abbiamo pure verificato che il processo funziona, accedendo al servizio tramite un comune internet browser di un PC. Quello che resta da fare è collegarsi con l'iPhone e dimostrare che il processo funziona.

4.1 Primo passo: reperimento del device.

Non essendo in possesso di un iPhone, per realizzare la demo userò un iPod Touch. L'iPod Touch è il fratellino minore dell'iPhone, in quanto manca del chip telefono e della fotocamera. Le funzionalità principali che a noi servono però sono presenti (WiFi e browser internet), quindi il device può essere usato tranquillamente per soddisfare i nostri scopi.

Nella figura seguente ho riportato l'immagine della schermata iniziale del mio iPod. Dicevamo che per la nostra demo abbiamo bisogno di rete e browser internet... verificiamo se i prerequisiti sono soddisfatti:

- *Rete*: in alto a sinistra c'è il simbolo che indica la connessione WiFi attiva, quindi abbiamo l'indicazione se il nostro dispositivo è connesso in rete o meno. Come in un normale telefonino, il numero di tacche indica la potenza del segnale di rete, in questo caso rete WiFi.
- *Browser*: la prima icona è relativa al programma che utilizzeremo per la connessione: il browser internet Safari. È integrato nell'iPod/iPhone e non necessita di particolari personalizzazioni.

Bene...abbiamo a disposizione tutto ciò che ci serve, quindi possiamo procedere con la demo.



4.2 Secondo passo: controllo firmware

Il firmware del device dev'essere a livello 2.0 o superiore, in quanto è necessario un protocollo di comunicazione sicura WiFi chiamato WPA2 Enterprise che e' stato introdotto con questa versione di firmware. L'aggiornamento firmware viene fatto in modo standard dall'applicazione iTunes. Per questa prova, ho installato il firmware versione 2.2.1, ossia il livello più alto disponibile alla data.



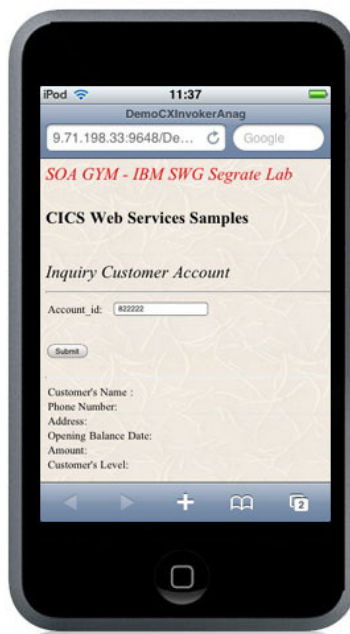
4.3 Terzo passo: setup di rete/collegamento WiFi

Le modalità di connessione alla rete sono dipendenti dall'installazione aziendale. Per l'accesso alla rete IBM, ho seguito le info a <http://w3.tap.ibm.com/w3ki/display/iphone/Connect+to+IBM+Network>



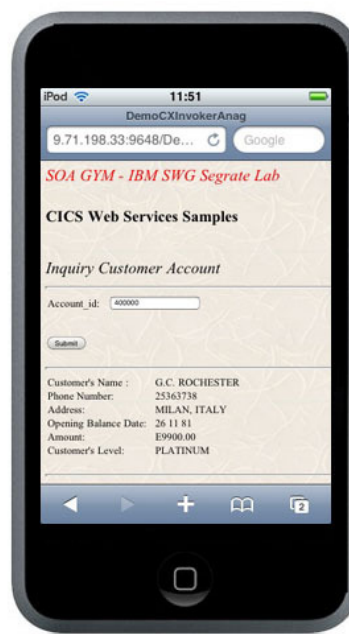
4.4 Quarto passo: collegamento al Web Service

A questo punto il nostro iPod è connesso alla intranet aziendale e quindi possiamo collegarci alla servlet del Web Service. Questo viene fatto fornendo al browser Safari l'indirizzo HTTP relativo. Nel nostro caso, l'indirizzo e' `http://9.71.198.33:9648/DemoCICS/DemoCXInvokerAnag.faces`. Una volta dato l'enter, ci compare la schermata HTML relativa. Il collegamento è attivo.



4.5 Quinto passo: ottenimento dei dati

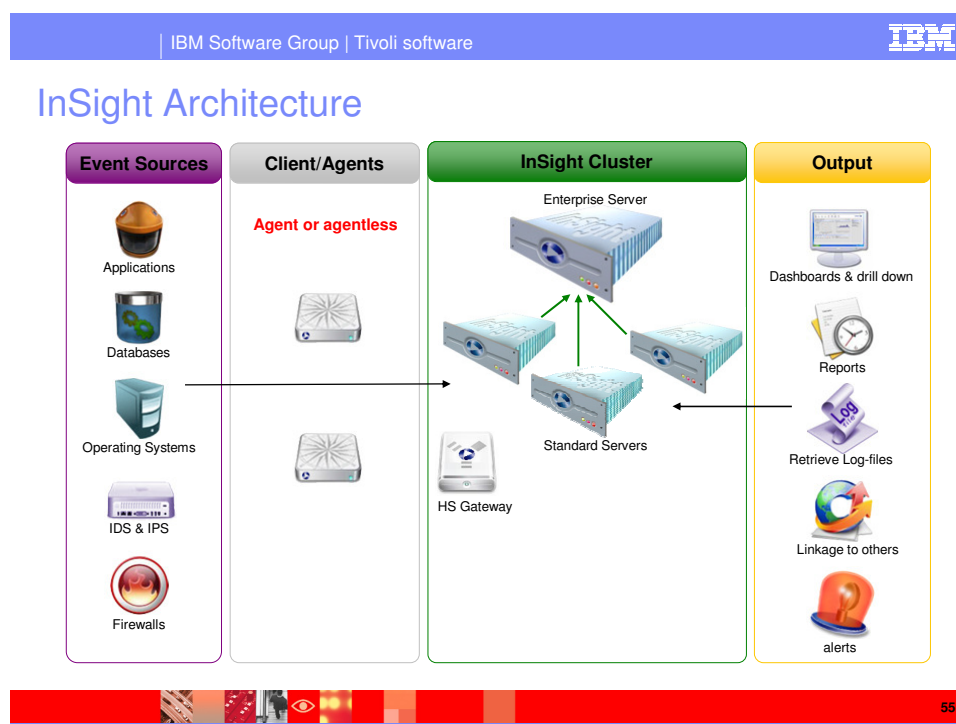
A questo punto proviamo a fare un paio di query per validare il flusso del processo. Nel mio caso userò il mio **Account Id** (822222) e quello di Rochester (400000). Come si può vedere dalle schermate seguenti, le transazioni sono state eseguite e abbiamo ottenuto i dati che volevamo.



5. Facciamo un'altra demo

Con la demo precedente abbiamo dimostrato che è possibile 'modernizzare' anche una vecchia applicazione mainframe e renderla accessibile da parte di device nuovissimi come l'iPhone. Per rendere questo aspetto ancora più interessante volevo ampliare il discorso, provando l'accesso ai prodotti software, in modo che si possa valutare quanto sia flessibile questo approccio.

Per dimostrare questo sfrutto l'occasione offerta da un prodotto al quale mi sono interessato in questo periodo a causa della nuova normativa sulla Data Privacy che coinvolge direttamente gli amministratori di sistema, i quali sono tenuti ad adottare misure specifiche per verificare le loro stesse attività sui dati e per la gestione dei log a livello Enterprise. Il prodotto in questione si chiama Tivoli Compliance Insight Manager (TCIM).



Il prodotto raccoglie eventi e log da varie fonti, li raccoglie in un suo database e permette di verificare la compliance rispetto alle più diffuse normative di sicurezza come la Sarbanes-Oxley (SOX), la normativa Health Insurance Portability and Accountability Act (HIPAA), International Organization for Standardization (ISO) 17799, GLBA e per la normativa Basel II.

Il mio obiettivo è quello di accedere tramite iPhone al server TCIM per ottenere i report di compliance, e fare poi il drill down fino ad arrivare all'evento di sicurezza che pregiudica la compliance.

Il prodotto TCIM non è un prodotto che gira su z/OS, però raccoglie anche dati z/OS e saranno questi che analizzerò. E il fatto di poter accedere con lo stesso strumento sia a un ambiente z/OS che a un ambiente non-z/OS dimostra la flessibilità della soluzione.

5.1 Accediamo al server TCIM

Per fare questa dimostrazione ho attivato una macchina TCIM che usiamo internamente per le demo del prodotto, e ho provato ad accederci tramite il browser Safari. Poichè il colloquio e' fatto con traffico HTTP e verranno utilizzate solo schermate HTML non prevedo problemi nell'accesso al server. Infatti riesco ad accedere subito al server TCIM ed a ottenere la vista della compliance di sicurezza a livello enterprise, come si può vedere nell'immagine seguente:

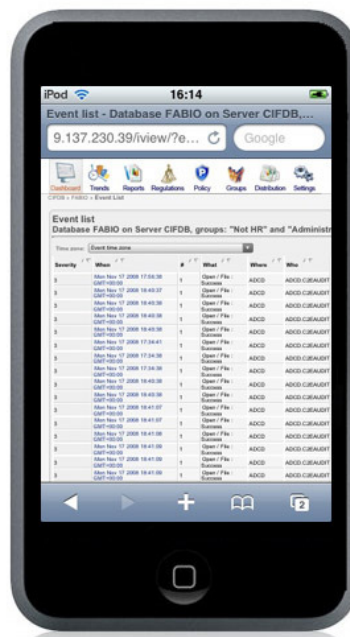


Ma la visione dell'intera enterprise, anche se molto importante, non ci basta. Vogliamo verificare la possibilità di fare drill-down e poter scendere più nel dettaglio. Per questo scegliamo un db dove sono stati precedentemente raccolti dati, e in questo caso scegliamo il db che ho creato per raccogliere i dati degli eventi z/OS, passati al TCIM dal prodotto zAudit. Per tenere isolati questi eventi ho creato un db con il mio nome, quindi facciamo zoom sul db FABIO.

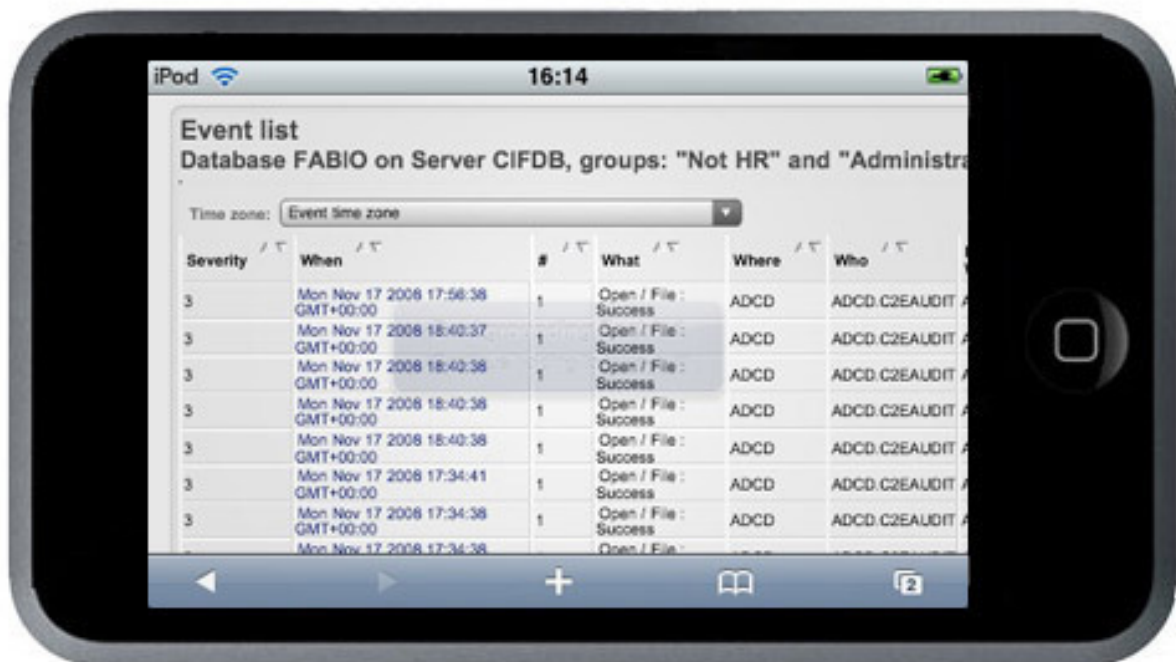


Accedere ad applicazioni z/OS... tramite iPhone!

Una volta aperta la vista sul db posso analizzare in generale la situazione di compliance di sicurezza per quanto riguarda questo genere di eventi. Ma io voglio ancora andare più in dettaglio, quindi scendo ancora di livello fino ad arrivare alla lista dei singoli eventi.



Forse questa lista è un pò poco leggibile.... ma qui entra in campo la tecnologia dell'iPhone. Posso fare zoom sui dati con un semplice movimento delle dita, oppure posso inclinare il device per avere una vista in orizzontale, sfruttando l'accelerometro che mi ruota automaticamente il display. Per una visualizzazione migliore, faccio lo zoom della foto che allego:



6. Conclusione

Siamo arrivati alla fine del documento, ed è quindi giunto il momento di tirare le conclusioni di quanto abbiamo fatto. In questo documento, più che sfruttare le possibilità offerte da novità tecnologiche, ho cercato di mostrare come è possibile integrare l'installato esistente senza dover per forza riscrivere codice per fare innovazione. Infatti, se avessi voluto porre l'accento sulla novità tecnologica, sarebbe forse stato più facile scrivere un programma client sull'iPhone che parlasse con una applicazione server... ma una demo del genere avrebbe solo fatto vedere le luci della novità senza far riflettere su i costi legati al cambiamento della mia applicazione.

Chi vive nel mondo reale dell'IT conosce benissimo i costi che sono legati allo sviluppo di nuovo codice e quanto questi sviluppi siano dispendiosi in termini di tempo e di denaro. Ma poi è davvero sempre necessario dover sviluppare nuovo codice applicativo? Se nella mia azienda ho già un processo in atto che funziona, in fondo perchè dovrei cambiarlo? Il discorso della riusabilità dell'esistente si sente da tanto tempo, ma mai come in questo periodo viene esaltata la sua attualità, forse anche per il momento di crisi economica che stiamo attraversando.

Con questo documento ho cercato di mostrare in modo pratico come è possibile modernizzare un'applicazione (anche datata) senza modificarla di molto. Anzi ...facendo coesistere il modo di lavorare tradizionale con il nuovo. Siamo stati in grado di accedere a una vecchia applicazione 3270 e un server di un prodotto rilasciato da un paio d'anni. Non male mi sembra, visto che non abbiamo modificato le modalità di lavoro in essere!

Insomma, ho cercato di mostrare il fatto che l'introdurre novità nell'ambiente informatico non significa necessariamente buttare via l'esistente, e che il mainframe sia ancora un componente fondamentale anche per gli sviluppi IT del futuro.