

## SCHEDA PROGETTO

**Responsabile del progetto e dell'esecuzione del contratto:**

Prof. Massimo Maresca (Professore ordinario – Settore Scientifico Disciplinare - IINF-05/A – Sistemi di elaborazione delle informazioni)

**Obiettivo del progetto:**

Il progetto SEBISS (SELF service Business Intelligence over the Spreadsheet Space) CUP: G39J23000920009 - finanziato nell'ambito del POR FESR della Regione Liguria, trae la sua motivazione dal fatto che l'industria, ed in particolare l'industria ligure, sta vivendo una significativa transizione digitale che la sta trasformando in modo radicale. L'aspetto specifico della transizione digitale a cui il progetto si propone di dare un contributo è quella della gestione delle attività aziendali. Il progetto poggia sulla classificazione delle attività dell'azienda in due grandi aree, quella dell'Operation, governata dagli ERP ed in generale dagli Enterprise Information Systems, che conoscono ed orchestrano i processi di business, e quella dell'Intelligence, governata dalle persone, che pensano, valutano, definiscono strategie etc., tipicamente avvalendosi del supporto di strumenti di produttività individuale (documenti testuali, fogli di calcolo, etc.). Il progetto ha una significativa ricaduta per quanto riguarda la "valorizzazione delle persone nei processi produttivi", espressamente citata tra le traiettorie del Polo Sosia, supportando la sincronizzazione degli strumenti di produttività individuale (il foglio di calcolo in particolare) con le basi dati aziendali gestite dagli ERP, in modo da garantire che le attività dell'Intelligence, siano costantemente allineate con quelle dell'Operation. Tutto ciò in perfetta sicurezza, il che potenzia il ruolo di coloro che svolgono attività di Intelligence.

Dal punto di vista scientifico e tecnologico, le tematiche progettuali sono parte dell'evoluzione verso il Cloud e verso l'Edge Computing, in particolare per quanto riguarda il requisito della "sovereignty", oggi centrale nella programmazione europea, come appare dalle iniziative degli International Data Spaces (<https://internationaldataspace.org/>) e delle Federated Cloud Infrastructure (<https://www.data-infrastructure.eu/GAIA/>). La tecnologia sulla quale il progetto si basa, è un esempio di realizzazione del concetto di International Data Space, in cui i dati condivisi rimangono sempre all'interno dei domini amministrativi delle aziende che li possiedono, escono dagli stessi solo cifrati e sono decifrabili solo dai destinatari che hanno diritto di accedere ad essi. Il fatto che il progetto sia perfettamente inquadrato nell'iniziativa International Data Space può far prevedere importanti ricadute del progetto sul Polo e sul territorio ligure. A tale proposito il DIBRIS ha in corso iniziative di ricerca e sviluppo formalizzate con Leonardo SpA, con Confcommercio Salute e con altri soggetti proprio sulle tematiche oggetto del progetto.

Il principale aspetto innovativo del progetto deriva dal fatto che lo stesso propone una piattaforma avanzata per la condivisione di dati che si pone come riferimento a livello internazionale. Le caratteristiche innovative più significative del progetto, meglio espresse nel seguito del documento, sono le seguenti:

- La scalabilità, al centro dell'evoluzione scientifica e tecnologica, che verrà perseguita attraverso l'uso di architetture virtualizzate basate su Microservice cooperanti, che nascono e muoiono dinamicamente.
- La federazione, che consiste nel realizzare il sistema attraverso un insieme di Server coordinati, associati a diversi Domini, che supportano servizi sia a livello Intra Dominio sia a livello Inter Dominio in condizioni di sicurezza.

Il progetto si pone diversi obiettivi sia dal punto di vista tecnologico che dal punto di vista del miglioramento delle conoscenze.

Dal punto di vista del miglioramento delle conoscenze il progetto va ad esplorare le modalità di collaborazione, accesso a dati in tempo reale e più in generale "Cooperative Business Intelligence". Studia quindi da un lato le modalità attraverso le quali oggi le attività di business e di intelligence collaborativa hanno luogo nei diversi domini, dall'altro le tecnologie di supporto, ed infine l'influenza che tali tecnologie hanno sulle modalità di collaborazione.

L'obiettivo è quello di utilizzare questa conoscenza per proporre ed affermare nuove forme di collaborazione basate sulla sincronizzazione di tabelle supportata dalla piattaforma.

**Oggetto della prestazione:**

Attività di consulenza avente per oggetto lo "Sviluppo di un Sistema Scalabile e Federato a supporto della Condivisione di Dati Tabellari" necessaria per il progetto SEBISS, (SELF service Business Intelligence over the Spreadsheet Space) finanziato nell'ambito del POR FESR della Regione Liguria – CUP: G39J23000920009

**Descrizione dettagliata della prestazione:**

Per poter descrivere le attività richieste è necessario inquadrare il funzionamento del sistema nella sua situazione attuale. In esso operano tre componenti, chiamati rispettivamente Client, Notify e Server.

Il Client è costituito da un sistema basato su Foglio di Calcolo, tipicamente Excel, ma può anche corrispondere ad un sistema software di tipo generale, che interagisce con gli altri componenti attraverso una libreria (SDK) e/o attraverso chiamate a Web Service REST.

Il Notify è un Componente Software, tipicamente ma non necessariamente localizzato nello stesso sistema di calcolo nel quale risiede il Client, al quale il Client stesso demanda le funzioni di interazione con il Server. Esiste quindi una comunicazione tra Client e Notify.

Il Server è il Componente Software localizzato "in the Cloud" e cioè in un sistema di calcolo esterno, che gestisce le interazioni tra diversi Client, attraverso i rispettivi Notify.

Nella situazione attuale il Client effettua una richiesta di servizio al Server attraverso il Notify e rimane in attesa (stato di Wait) fino al soddisfacimento della richiesta stessa. Il Notify riceve la richiesta, la inoltra al server, attende l'esito dell'elaborazione e quindi la risposta del Server, ed a sua volta risponde al Client, finalizzando la transazione. L'attività richiesta ha per oggetto l'introduzione al supporto della scalabilità, e quindi la realizzazione di un sistema scalabile completo e pronto per l'utilizzazione, l'introduzione al supporto della federazione, e quindi la realizzazione di un sistema costituito da più server completo e dimostrabile, e la realizzazione finale del sistema federato e scalabile derivante dall'integrazione dei due componenti sopramenzionati, anche in questo caso completo e pronto per l'uso.

Referente per lo svolgimento delle attività sarà il Prof. Massimo Maresca, al quale il prestatore si riferirà per ogni necessità collegata al rapporto, anche in ordine alle modalità di svolgimento dell'attività stessa.

L'attività verrà articolata in tre fasi consecutive corrispondenti alle tre sottoattività sopramenzionate. La prima fase viene attivata alla stipula del contratto mentre le fasi successive potranno essere avviate al completamento della precedente e di concerto con il Responsabile Scientifico del DIBRIS che valuterà la necessità di procedere in base allo stato e alle necessità delle attività di ricerca in cui l'incarico si colloca.

**3.1. Fase 1: Scalabilità**

L'introduzione della scalabilità richiede lo sviluppo di nuovi componenti sia sul lato periferico sia sul lato centrale. In particolare, sul lato periferico dovrà essere introdotto un sistema di disaccoppiamento tra le richieste degli utenti, oggi effettuate eseguite in modo sincrono dal server, ed il servizio. L'organizzazione del software dovrà quindi prevedere che il componente Client riceva una conferma dal sistema Server non più come ora al soddisfacimento della richiesta di servizio ma all'accettazione della stessa. La richiesta entrerà quindi in uno stato Pending, in attesa del soddisfacimento, ma sarà possibile persistere lo stato e spegnere il sistema, nonché svolgere altre attività, mentre la richiesta è in tale stato. Non potranno invece essere attivate richieste che riguardino lo stesso contesto di quella nello stato Pending.

Il componente Notify si farà quindi carico dell'invio della conferma al componente Client e del colloquio con il componente Server per l'effettiva esecuzione del servizio. Non appena riceverà da quest'ultimo la conferma dell'avvenuta esecuzione del servizio trasmetterà la notifica al componente Client, che farà passare la richiesta dallo stato Pending allo stato Done.

Di fatto quindi il componente Notify svolgerà il ruolo di disaccoppiatore della sincronia tra Cliente Server, garantendo che non vengano perse richieste, e cioè che una richiesta venga posta nello stato Done solo al termine del suo soddisfacimento finale e che non accadano situazioni di stallo, in cui i componenti risultano tutti inattivi in presenza di richieste nello stato Pending.

Sul lato centrale il componente Server dovrà essere strutturato in modo tale da trasformare l'esecuzione dei servizi richiesti in modo tale che vengano eseguiti asincronamente rispetto alle richieste. Cioè, le richieste dovranno essere analizzate e verificate ed in caso di richieste corrette dovranno essere messe in una Message Queue, pronte per essere eseguite. L'esecuzione effettiva del servizio avverrà su iniziativa del Server, e non più sincronamente con la richiesta, secondo politiche di scheduling definite dal Server stesso. Questo estrarrà i messaggi dalla coda nella misura che riterrà opportuno e li assegnerà a Thread di Esecuzione concorrenti che si faranno carico dell'esecuzione. Il sistema dovrà prevedere la sincronizzazione tra il sistema di ricezione, l'analisi e l'accodamento delle richieste ed i Thread di esecuzione.

Particolare cura andrà posta al disaccoppiamento tra ricezione/accodamento delle richieste ed esecuzione delle richieste stesse, in modo tale da assicurare che non vi siano Thread idle durante il soddisfacimento delle richieste. Si richiede l'applicazione di tecniche asincrone (e.g., NIO) per la gestione delle richieste.

Al termine di questa attività di sviluppo si richiede la consegna di un sistema scalabile funzionante, documentato e pronto per l'uso. Il Dibris si riserva di collaudare il sistema verificando tra l'altro, l'effettivo disaccoppiamento tra le richieste di esecuzione dei servizi da parte del Componente Client e l'esecuzione dei servizi stessi, l'assenza di stalli, il completamento delle richieste in modo ritardato, la presenza della Message Queue sul lato server, la terminazione dei Thread di accodamento delle richieste durante la fase di esecuzione delle stesse. Si riserva infine di verificare la completezza di quanto sviluppato per l'effettivo inserimento della scalabilità nel sistema oggi esistente.

### 3.2. Fase 2: Federazione

Il sistema oggi esistente, sul quale dovrà essere innestata la funzionalità della federazione, è bastato su un unico componente Server. Nel sistema federato da sviluppare ogni Dominio (e cioè ogni organizzazione) avrà un proprio Server attraverso il quale servirà le richieste di servizio dei diversi Client. Ovviamente dovrà esistere un meccanismo di gestione ed esecuzione delle richieste Inter Dominio, in modo tale da consentire la sincronizzazione dei dati tra utilizzatori appartenenti a Domini diversi.

Il nuovo Server dovrà permettere ai Client di condividere dati generici con altri Client all'interno del proprio Dominio. Ogni Client potrà comunicare con il Server del proprio Dominio per caricare o aggiornare dati che vuole condividere con altri o per scaricare dati che siano stati condivisi con lui.

Il sistema federato prevederà inoltre per ogni Client la possibilità di condividere dati generici con Client appartenenti ad altri Domini. Ogni Client potrà scaricare dati da un Server di diverso Dominio (ma non caricare né aggiornare), purché un Client appartenente a quel Dominio abbia condiviso con lui quei dati.

Sarà quindi necessario, prima di tutto, riprogettare il meccanismo di autenticazione e definire un meccanismo di autenticazione inter Dominio basato su token che permetta ad un Client a@A di Dominio A di leggere dei dati presenti sul Dominio B con lui condivisi e caricati da un Client del Dominio B.

Completata la fase di autenticazione, il sistema Server dovrà essere riorganizzato in modo tale da prevedere che le richieste sottoposte da ogni Cliente al Server del proprio Dominio vengano da queste gestite in modo autonomo se relative a sincronizzazioni con Client appartenenti allo stesso Dominio. Viceversa. Le richieste relative a sincronizzazioni con Client appartenenti a Domini diversi dovranno essere inviate ai Server di tali Domini per l'esecuzione. Si tratta quindi prima di tutto di identificare tali server a partire dall'identificatore dei Client (tipicamente l'indirizzo di posta elettronica), di verificare l'esistenza di tali server, di stabilire un contatto con essi, di trasmettere le richieste di servizio in modo affidabile e acknowledged e di verificare il loro soddisfacimento. I sistemi Server, diversi quindi per ogni Dominio, riceveranno richieste di servizio dai sistemi Client appartenenti al Dominio da essi gestito e da sistemi server di altri Domini in casi di richieste da parte di questi ultimi.

Al termine di questa attività di sviluppo si richiede la consegna di un sistema federato funzionante, documentato e pronto per l'uso. Il Dibris si riserva di collaudare il sistema verificando in particolare l'esecuzione delle richieste di servizio intra dominio ed interdominio e l'autenticazione distribuita che dovrà essere sicura. Si riserva infine di verificare la completezza di quanto sviluppato per l'effettiva migrazione del sistema attuale verso un sistema federato.

### 3.3. Fase 3: Integrazione

In questa fase si integreranno i due componenti sviluppati nelle due fasi precedenti per ottenere un sistema scalabile e federato.

Il sistema dovrà quindi prevedere la presenza di più Server, uno per ogni Dominio (o Organizzazione) che gestisca le richieste dei Client di quel Dominio, e dei meccanismi di autenticazione e di gestione delle richieste di servizio da parte di Client appartenenti a Domini diversi. Ciascun Server dovrà essere scalabile e quindi prevedere il disaccoppiamento tra le richieste di servizio e l'esecuzione dei servizi richiesti, in modo da consentire ai Client di entrare in uno stato Pending, opportunamente persistito, durante la fase di esecuzione del servizio per poi passare allo stato Done al termine dell'esecuzione.

Si dovrà avere cura dell'integrazione anche con il sistema Server pubblico, localizzato "in the Cloud", che gestisce i Client che non appartengono a domini privati, prevedendo che in assenza di un server di Dominio un Client venga associato al sistema Server pubblico.

Si richiede di sviluppare un sistema di monitoraggio e di log del funzionamento del sistema che ne consenta l'analisi.

Al termine di questa attività di sviluppo si richiede la consegna di un sistema federato e scalabile funzionante, documentato e pronto per l'uso. Il Dibris si riserva di collaudare il sistema verificando in particolare la scalabilità e la federazione nel loro funzionamento congiunto e la completezza dell'intero sistema e della sua documentazione. La verifica sarà effettuata anche attraverso il sistema di monitoraggio e log del funzionamento realizzato in questa fase di sviluppo.

Il Dibris si riserva infine di verificare la completezza di quanto sviluppato per l'effettiva migrazione del sistema attuale verso un sistema scalabile e federato.

#### **Competenze richieste al prestatore:**

- Diploma di Laurea quinquennale in Ingegneria Elettronica o Ingegneria Informatica conseguito ai sensi della normativa previgente al D.M. 3 Novembre 1999, no. 509 ovvero Laurea Specialistica (ex DM509/99) nelle classi di laurea CLS 32/S o 35/S ovvero Laurea Magistrale (ex DM 270/04) nelle classi di laurea LM-29 o LM-32
- Esperienza, anche in ambito accademico, in istituzioni o enti, pubblici o privati, anche a supporto di studi e ricerche nel settore di riferimento di almeno 3 anni, con particolare riferimento a sviluppo di applicazioni Web a livello professionale, sia in ambito industriale sia in ambito accademico sia a livello di consulenza;
- Conoscenze e competenze documentabili attraverso il curriculum ed acquisite tramite attività di ricerca o esperienze lavorative nei seguenti ambiti:
  - Linguaggi di programmazione Java e C#.
  - Conoscenza dei principali framework per la programmazione Web (p. es., Tomcat, Spring e RabbitMQ)
  - Conoscenza dei processi informativi in ambito industriale e/o finanziario e/o amministrativo e/o logistico.

#### **Durata del progetto:**

La prestazione deve essere conclusa entro 6 mesi

#### **Compenso:**

Compenso lordo per l'intero periodo contrattuale: euro 24.000,00 + IVA (se dovuta) e comprensivo di oneri previdenziali ed assistenziali a carico del prestatore, se dovuti;

Modalità di pagamento: l'attività è frazionata in 3 fasi, a ciascuna delle quali corrisponderà un pagamento di € 8.000,00 (ottomila/00) + iva e comprensivo di oneri previdenziali ed assistenziali a carico del prestatore (se dovuti).

**Natura Fiscale della prestazione:**

Prestazione unica ad esecuzione pressoché istantanea:

- lavoro autonomo – redditi diversi (art. 67, comma 1, lett. I, D.P.R. 917/86 TUIR);
- lavoro autonomo – redditi di lavoro autonomo- professionisti abituali (art. 53, comma 1, D.P.R. 917/86 TUIR);

Il Responsabile del progetto e dell'esecuzione del contratto

(prof. Massimo Maresca)

(Firmato Digitalmente)