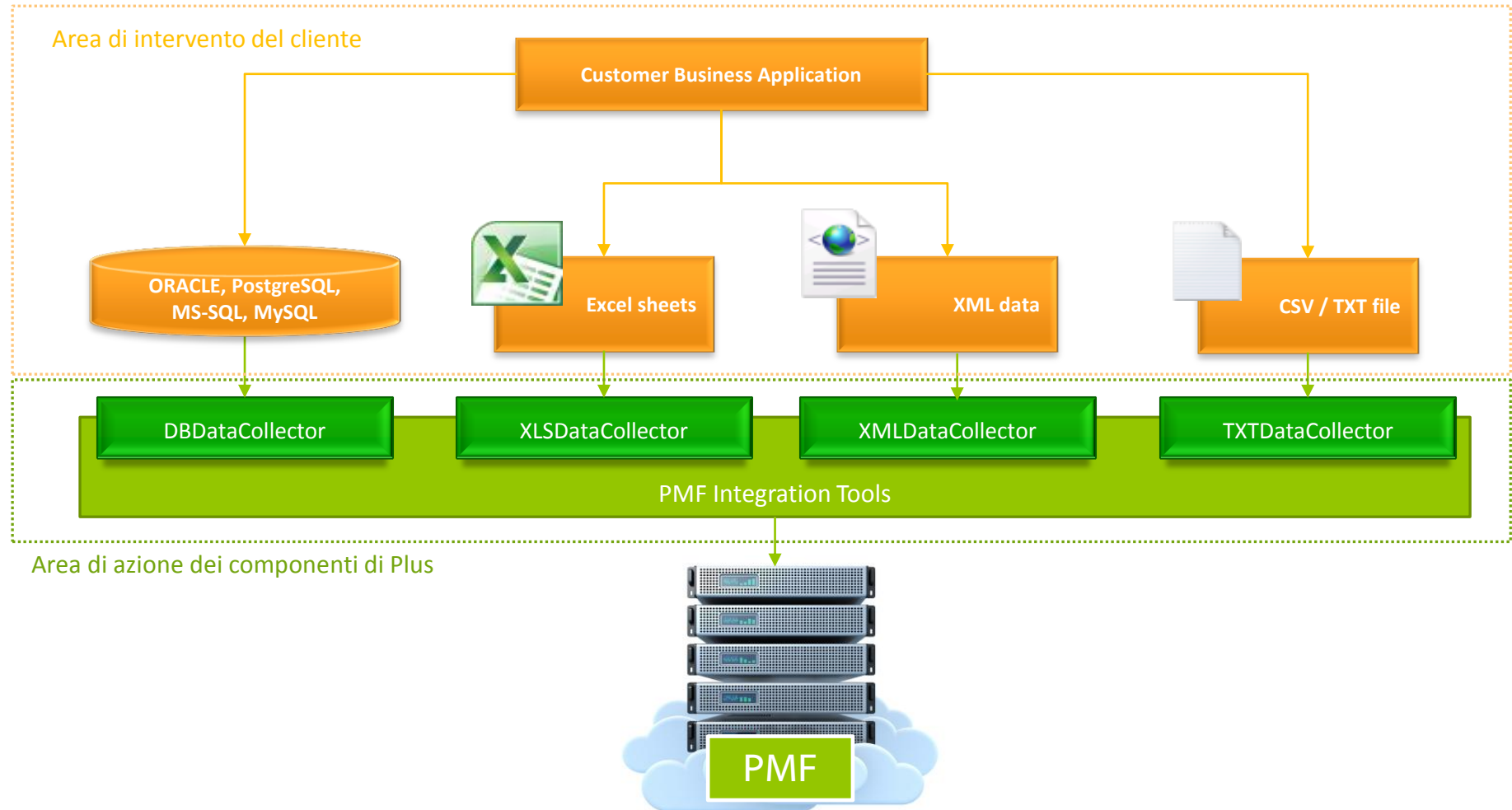


PMF

Integration tools

Soluzione integrata per la
comunicazione con le Customer
Business Applications

Architettura



Configurazione

La configurazione dei componenti è basata su semplici file di testo.

Qui di seguito un esempio dei parametri per il componente più complesso, il DBDataCollector.

nome parametro	Descrizione
DB_USERNAME	Nome utente di accesso al DB di transito
DB_PASSWORD	Password di accesso al DB di transito
DB_URL	URL per la connessione al DB, specifico per ciascun DBMS (es. jdbc:postgresql://localhost/mydb)
JDBC_DRIVER	Nome della classe del driver JDBC da utilizzare (es. org.postgresql.Driver)
PMF_USERNAME	Nome utente assegnato per collegarsi al server PMF
PMF_PASSWORD	Password dell'utente PMF
PMF_WSDL_URL	URL del web service di PMF (solitamente non è necessario modificare questo parametro)
SMTP_SERVER	Server SMTP tramite il quale spedire comunicazioni email con informazioni utili
FROM_ADDRESS	Mittente delle email
TO_ADDRESS	Destinatario delle email. Gli indirizzi devono essere separati da “;”
LOG_FILE_PATH	Percorso in cui scrivere il file di log del componente. Il file di log si chiama DBDataCollector.log
PMF_JOB_NAME	Nome del gruppo di messaggi che verrà creato sul server PMF
PMF_ALL_OR_NOHING	Se YES, procede con l'invio solo se tutti i messaggi superano la validazione del server PMF, in caso contrario annulla l'invio e registra sulla base dati l'esito negativo con una descrizione breve del problema. Se vale NO procede con l'invio dei soli messaggi che superano la validazione del server PMF; sulla base dati del Cliente viene comunque riportato quali messaggi non sono stati trasmessi e una descrizione breve del problema.

Strutture dati

Le strutture dati per l'esportazione vengono fornite da Plus al momento dell'attivazione del servizio.
Qui di seguito la struttura dati per il componente DBDataCollector per i sistemi ORACLE e PostgreSQL.

```
CREATE TABLE pmf_messages (  
  message_id NUMBER NOT NULL,  
  created timestamp NOT NULL ,  
  submitted timestamp ,  
  sender_name VARCHAR2(2048) NOT NULL,  
  recipient VARCHAR2(20) NOT NULL,  
  text_message VARCHAR2(700) NOT NULL,  
  job_type VARCHAR2(16) NOT NULL,  
  message_state VARCHAR2(512) DEFAULT 'READY',  
  message_error VARCHAR2(1024),  
  tag_1 VARCHAR2(64),  
  tag_2 VARCHAR2(64),  
  tag_3 VARCHAR2(64),  
  tag_4 VARCHAR2(64),  
  pmf_message_fk number,  
  pmf_job_fk number,  
  CONSTRAINT messages_pkey  
  PRIMARY KEY (message_id) );
```

ORACLE

```
CREATE TABLE pmf_messages (  
  message_id bigserial NOT NULL,  
  created timestamp without time zone NOT NULL,  
  submitted timestamp without time zone,  
  sender_name character varying(2048) NOT NULL,  
  recipient character varying(20) NOT NULL,  
  text_message character varying(700),  
  job_type character varying(16) NOT NULL,  
  message_state character varying(512),  
  message_error character varying(1024),  
  tag_1 character varying(64),  
  tag_2 character varying(64),  
  tag_3 character varying(64),  
  tag_4 character varying(64),  
  pmf_message_fk bigint,  
  pmf_job_fk bigint,  
  CONSTRAINT messages_pkey  
  PRIMARY KEY (message_id));
```

PostgreSQL

Cost model

PMF è un servizio cloud-based. Si paga per l'accesso al servizio e il cost-model è costituito dalle sole due voci:

- Canone annuo per l'accesso alla piattaforma che comprende l'utilizzo di tutti i tools di integrazione disponibili oggi ed in futuro.
- Costo dei messaggi inviati

Non è necessario utilizzare hardware dedicato. I componenti sono scritti in linguaggio Java e sono quindi eseguibili su sistemi Windows o Unix, sia server che workstation.

L'attivazione dei componenti può essere eseguita manualmente o programmata con i consueti strumenti di sistema (cron in Unix, Task Scheduler in Windows).