

**UNIVERSITÀ DEGLI STUDI DI MILANO**  
POLO DIDATTICO E DI RICERCA DI CREMA

---



# Virtual Private Jet

Progetto di

**INGEGNERIA DEI PROCESSI AZIENDALI**

**Studente:**

Casazza Marco 771605

Lena Cota Guido 773358

**Docente:**

Damiani Ernesto

## Descrizione offerta e obiettivi

---

Il servizio **Virtual Private Jet (VPJ)** si pone come alternativa economica al noleggio di aerei privati per la copertura di tratte nazionali. Il servizio offre:

- viaggio in *first class* su aereo di linea, con tempi competitivi a quelli offerti da un aereo privato;
- trasporto in elicottero dall'aeroporto di arrivo alla destinazione finale;
- check-in agevolato alla partenza e tragitto dall'aereo all'elicottero in minicar, entrambi garantiti dal personale di terra della VPJ. Per minicar si intende una macchina elettrica adibita al trasporto all'interno dello spazio aeroportuale.

Il **business goal** del servizio è acquisire e fidelizzare l'utenza sottraendola a quella delle compagnie di aerei privati, da considerare come competitors principali. Gli obiettivi di *renew* sono considerati minoritari poiché si tratta di un servizio nuovo, dunque si è scelto di dare priorità al consolidamento della propria posizione sul mercato.

I principali valori aggiunti dell'offerta VPJ rispetto a quella con cui si pone in concorrenza sono:

- *costo del servizio*, prezzi più convenienti;
- *qualità del servizio*, compensa la comodità e il prestigio di un volo su aereo privato con altri servizi aggiuntivi (elicottero, minicar ed assistenza personale).

Come ulteriore accorgimento per aumentare la *customer satisfaction* e la propria competitività, è previsto uno sconto sull'offerta presentata se questa non rispetta alcune condizioni (note all'utente) sui tempi di viaggio.

# Utenza

---

I segmenti di utenza a cui si rivolge il servizio **Virtual Private Jet** sono singoli privati di fascia economica medio-alta, aziende e organizzazioni.

I canali attraverso cui il cliente può richiedere il servizio sono:

- applicazioni per dispositivi portatili, rilasciate per i maggiori sistemi operativi mobili (*Android, BlackberryOS, iOS, Meego, Windows Phone 7, ...*);
- Internet browser.

Prerequisito per l'accesso al servizio è la registrazione del cliente presso uno degli uffici VPJ sparsi sul territorio nazionale o via web.

# Partner

---

Il servizio prevede il coinvolgimento e la stipula di contratti con numerosi partner esterni.

## **Compagnie aeree:**

- riserva di posti in prima classe fino a un'ora prima dalla partenza, così da aumentare le probabilità di trovare posti disponibili per i clienti;
- accesso al **CRS** (*Computer Reservation System*), che fornisce un'interfaccia per la ricerca e la prenotazione dei voli.

## **Società noleggio elicotteri:**

- noleggio elicottero per trasporto urbano;
- accesso a interfacce simili al CRS delle compagnie aeree per la riserva e la prenotazione dei veicoli.

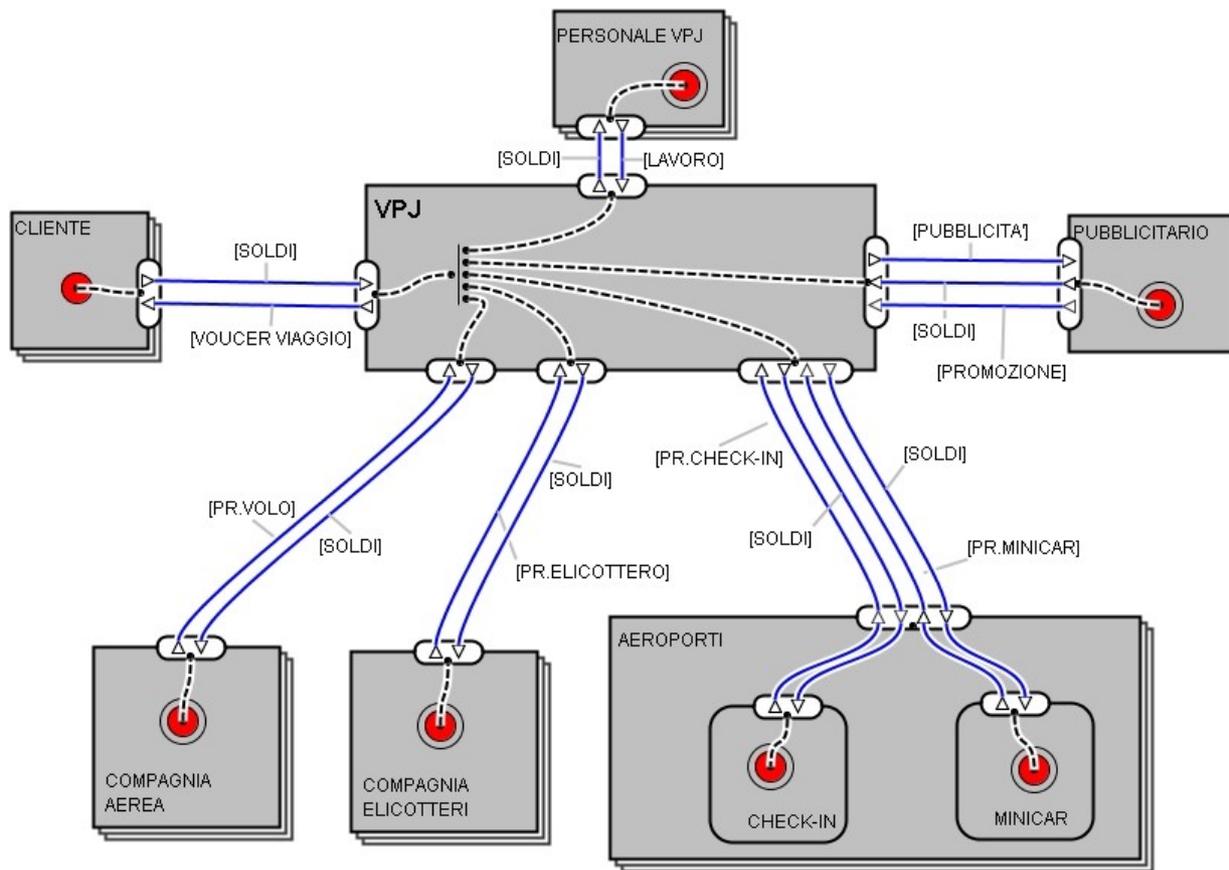
## **Società gestione aeroportuale:**

- noleggio sportello check-in, in cui il personale di terra della VPJ può accogliere i clienti e concludere le formalità burocratiche;
- noleggio minicar;
- accesso a interfacce simili al CRS delle compagnie aeree per la riserva e la prenotazione delle minicar.

## **Pubblicitario (concessionario):**

- vendita di spazi pubblicitari, come l'inclusione di banner all'interno dell'interfaccia del servizio VPJ;
- promozione dell'offerta Virtual Private Jet.

## e3 value model



Alcuni commenti:

- il prefisso *PR.* che compare in diversi *value object* è l'abbreviazione di "prenotazione";
- Il concetto di *singolo attore* è usato solo per VPJ e il concessionario Pubblicitario, mentre per tutti gli altri è stato usato il concetto di *segmento di mercato* (per sottolineare che il ruolo può essere assunto da diversi attori);
- i servizi offerti dalle società aeroportuali sono stati modellati come *value activity* per porre tutti i partner allo stesso livello di indipendenza;
- l'interfaccia tra VPJ e Pubblicitario mostra che gli spazi pubblicitari messi a disposizione della prima devono essere tanti e tali da poter garantire non solo la promozione del servizio VPJ ma anche un certo introito monetario.

## Ipotesi di lavoro

---

Alcune ipotesi di lavoro:

- il servizio non prevede il collegamento tra tutti gli aeroporti italiani, ma solo tra quelli che coprono le principali tratte di affari. Ad esempio: Milano Malpensa, Roma Fiumicino, Venezia San Marco, Palermo, Cagliari, Napoli Capodichino, Torino;
- le compagnie aeree con cui si hanno accordi sono *Alitalia* e *Lufthansa*, che utilizzano rispettivamente i CRS *Galileo* e *Amadeus*;
- la ricerca della disponibilità di un veicolo prevede la sua riserva temporanea, con la restituzione di un identificativo univoco che la individua. Ciò va incontro alle esigenze di realizzazione del servizio secondo una filosofia REST, seguendo la quale propagare lo stato della prenotazione sarebbe inutilmente dispendioso e poco conforme allo stile architetturale;
- le minicar e la loro gestione sono di competenza della società aeroportuale, che consente il noleggio dei veicoli ai partner autorizzati;
- dando al cliente la possibilità di indicare l'indirizzo di destinazione da raggiungere tramite elicottero si presuppone che questi indichi aree idonee all'atterraggio (sia da un punto di vista tecnico che di autorizzazioni);
- il costo del servizio viene direttamente addebitato sul conto corrente del cliente tramite *PayPal*. Gli estremi del conto e l'autorizzazione all'addebito diretto sono forniti dal cliente al momento della registrazione.

# Progettazione servizio REST

---

La consegna del progetto richiede la progettazione del servizio secondo lo stile architetturale REST, quindi sottoforma di applicazione web con modello di tipo client-server.

La metodologia seguita è caratterizzata da cinque fasi:

1. identificare le risorse da mostrare come servizi;
2. modellare le relazioni tra le risorse, facendo attenzione a non creare "isole" (rappresentazioni scollegate o irraggiungibili dalle altre);
3. definire un piano delle URI per indirizzare le risorse;
4. definire l'interfaccia comune tra client e server attraverso le operazioni CRUD del protocollo HTTP;
5. progettare e documentare la rappresentazione del formato delle richieste e delle risposte dell'intero servizio.

## 1. Identificare le risorse

Le risorse individuate sono le seguenti:

- *aeroporti*, l'insieme degli aeroporti serviti dalla società;
- *tratte*, l'insieme delle tratte servite, ottenute come prodotto cartesiano delle risorse *aeroporti*;
- *viaggi*, l'insieme dei viaggi, ognuno dei quali contiene una serie di informazioni che lo caratterizzano univocamente (codici dei veicoli, dati del viaggiatore, date e ore di partenza, ...);
- *utenti*, l'insieme dei clienti registrati presso il nostro servizio;
- *prenotazioni*, l'insieme delle prenotazioni effettuate dagli utenti.

## 2. Relazioni tra le risorse

Le istanze degli insiemi di risorse *aeroporti*, *tratte* e *utenti* sono recuperate da un database in modo statico e non sono altro che i nomi di aeroporti, tratte ed utenti. Le istanze degli insiemi *viaggi* e *prenotazioni* vengono invece generate in modo dinamico (lato server) come *id* numerico univoco.

La risorsa *prenotazioni* è una sotto-risorsa di *utenti*, e riferenzia l'istanza corrispondente della risorsa *viaggi*.

## 3. Piano delle URI

Assumendo che l'Host del servizio Virtual Private Jet sia **<http://www.vpj.it>**, lo schema delle URI delle risorse sarà il seguente:

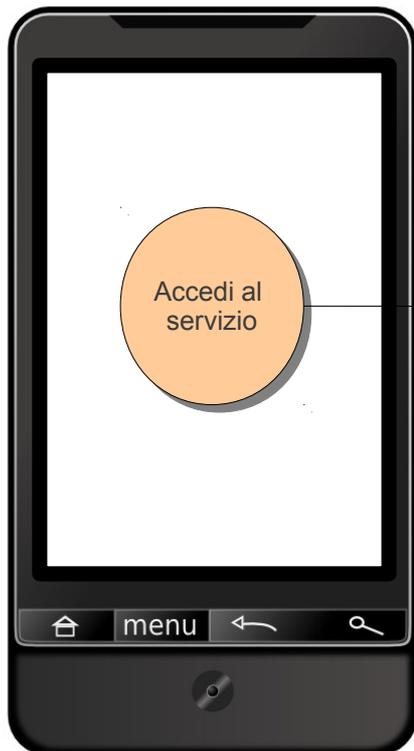
- *aeroporti*            /*aeroporti*  
                          /*aeroporti*/[nomeAeroporto]
- *tratte*                /*tratte*  
                          /*tratte*/[nomeAeroporto]-[nomeAeroporto]
- *viaggi*               /*viaggi*  
                          /*viaggi*/[id]
- *utenti*                /*utenti*  
                          /*utenti*/[user]
- *prenotazioni*       /*utenti*/[user]/*prenotazioni*  
                          /*utenti*/[user]/*prenotazioni*/[id]

## 4. Interfaccia comune

Una volta note le risorse e il modo in cui vengono indirizzate resta da definire il comportamento delle operazioni CRUD supportate dal protocollo HTTP.

Nei prossimi sotto paragrafi si affiancheranno dettagli sull'interfaccia comune e mock-up dell'applicazione lato client per dispositivi portatili.

## 4.1 Accesso al servizio



**URL:** `http://www.vpj.it`

**Pagina cachabile**

**GET /tratte**

Host: `www.vpj.it`

## 4.2 Selezione tratta e creazione del viaggio



**URL:** `http://www.vpj.it/tratte`

**Pagina cachabile**

**GET /tratte/[partenza-arrivo]**

Host: `www.vpj.it`

Il server **VPJ** crea una nuova istanza della risorsa *viaggi* (`/viaggi/[id]`), in cui inserisce le informazioni del volo.

Il client viene infine rediretto all'indirizzo:

`www.vpj.it/viaggi/[id]`

### 4.3 Conferma o rifiuta offerta



**URL:** `http://www.vpj.it/viaggi/[id]`

**Pagina non cachabile**

**DELETE** `/viaggi/[id]`

Host: `www.vpj.it`

Il client viene infine rediretto su: `www.vpj.it`

**GET** `/viaggi/[id]/form-dati-viaggio`

Host: `www.vpj.it`

### 4.4 Aggiorna dati viaggio



**URL:** `http://www.vpj.it/viaggi/[id]/form-dati-viaggio`

**Pagina non cachabile**

**PUT** `/viaggi/[id]`

Host: `www.vpj.it`

Aggiorna le informazioni della risorsa *viaggi* associata al cliente.

Il server crea una nuova istanza di *prenotazioni* (`/utenti/[user]/prenotazioni/[id]`), che riferenzia il viaggio appena aggiornato e su cui il client viene rediretto.

## 4.5 Riepilogo finale



**URL:** [http://www.vpj.it/\[user\]/prenotazioni/\[id\]](http://www.vpj.it/[user]/prenotazioni/[id])

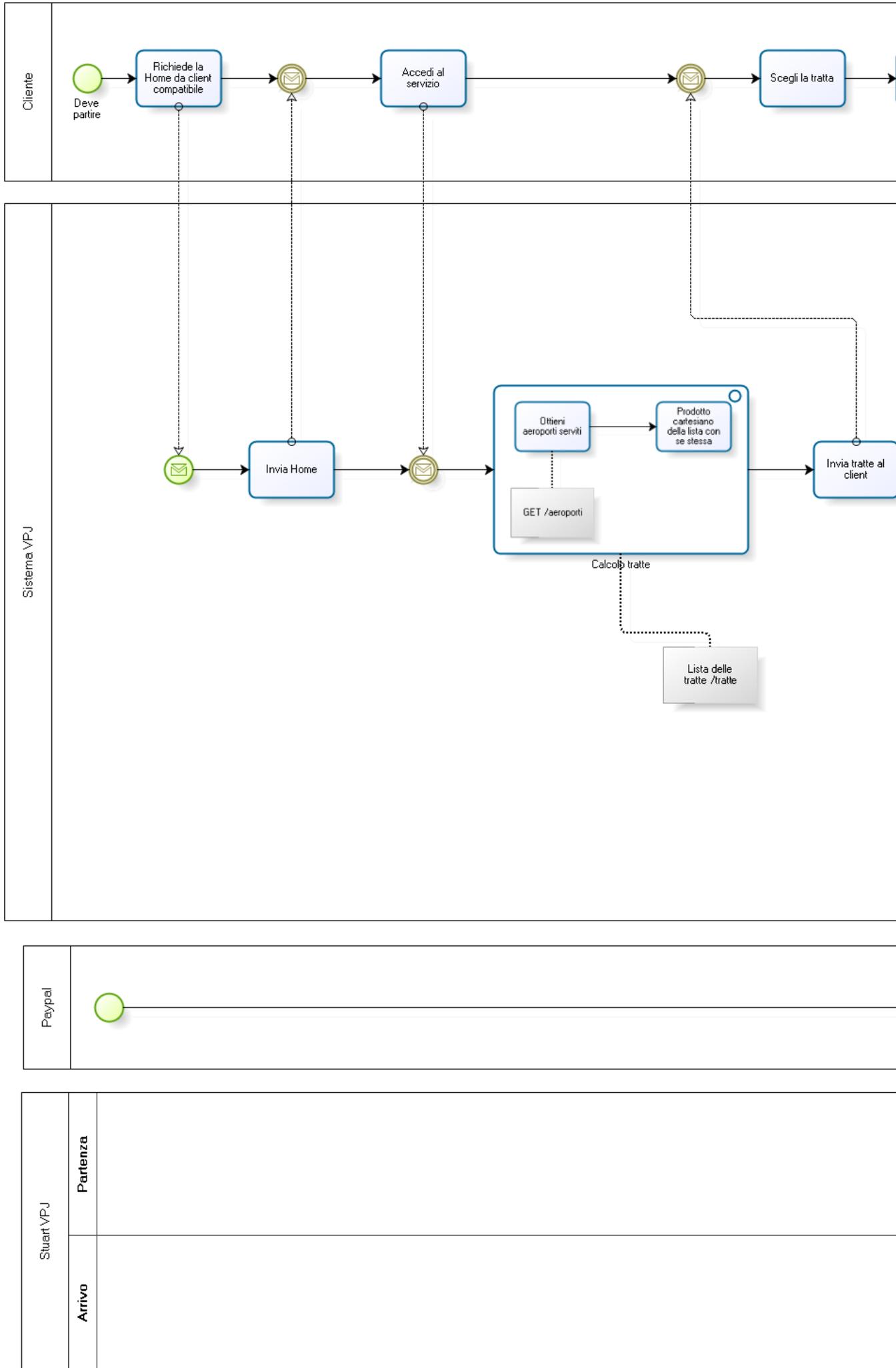
**Pagina cachabile**

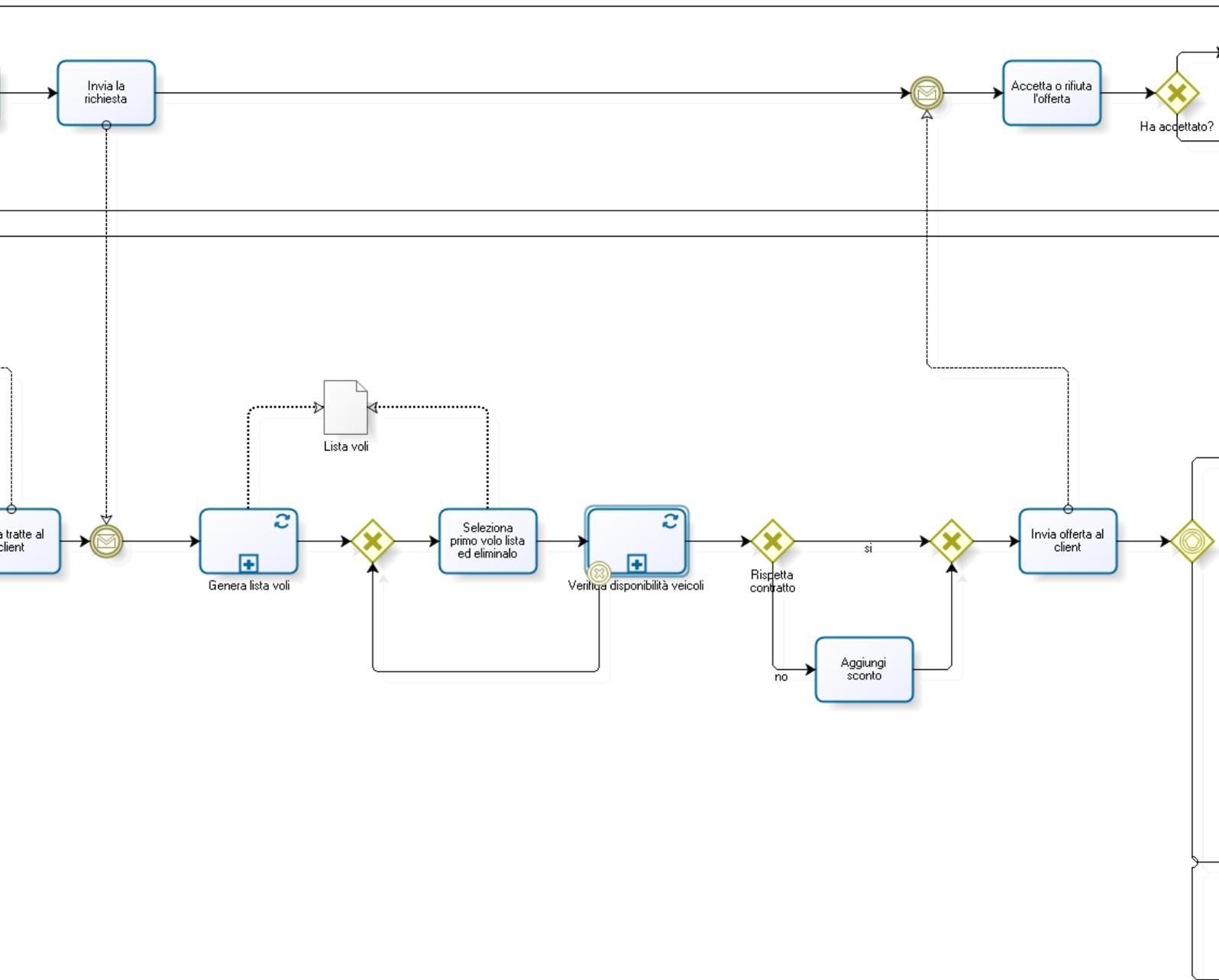
# BPMN

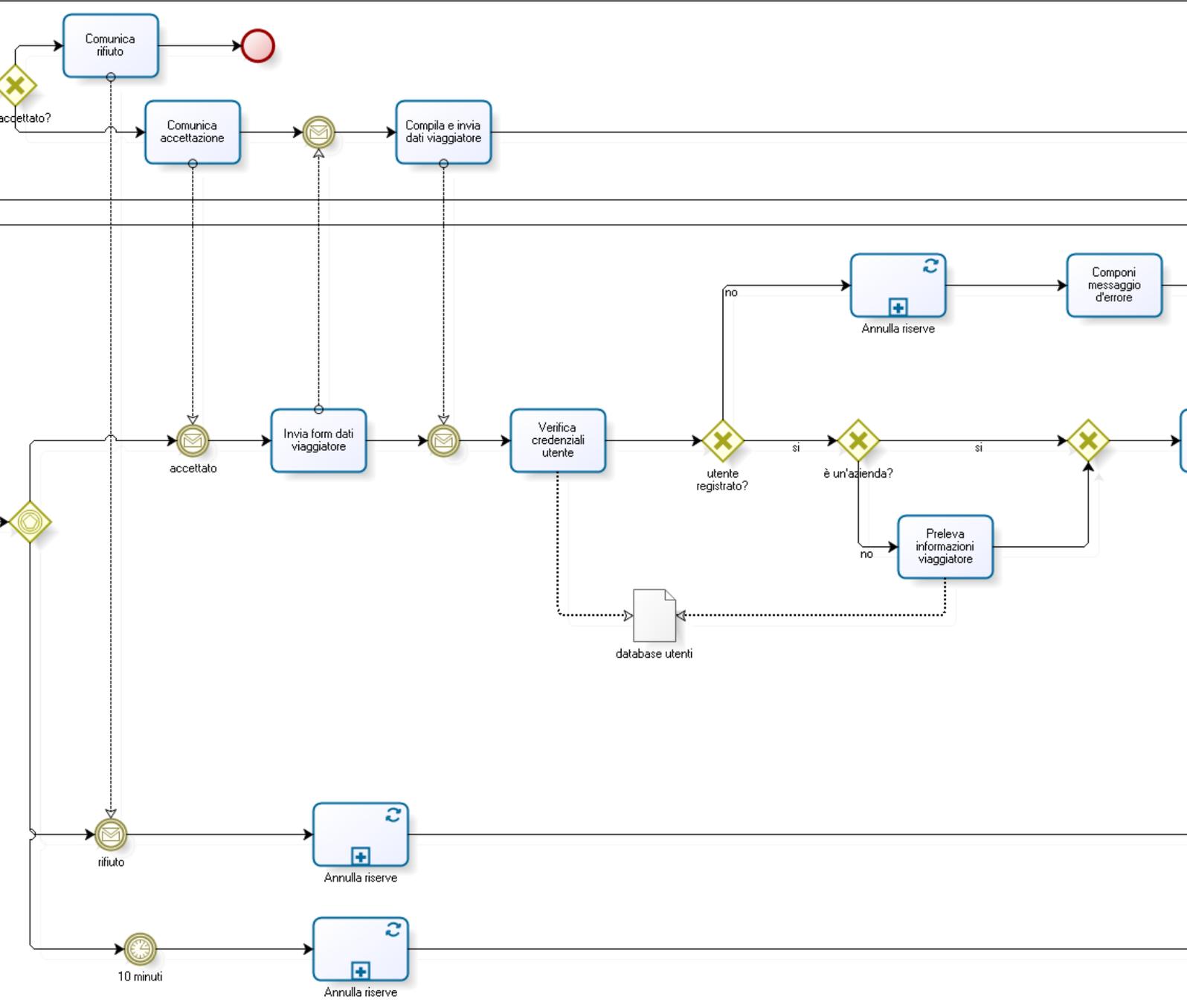
---

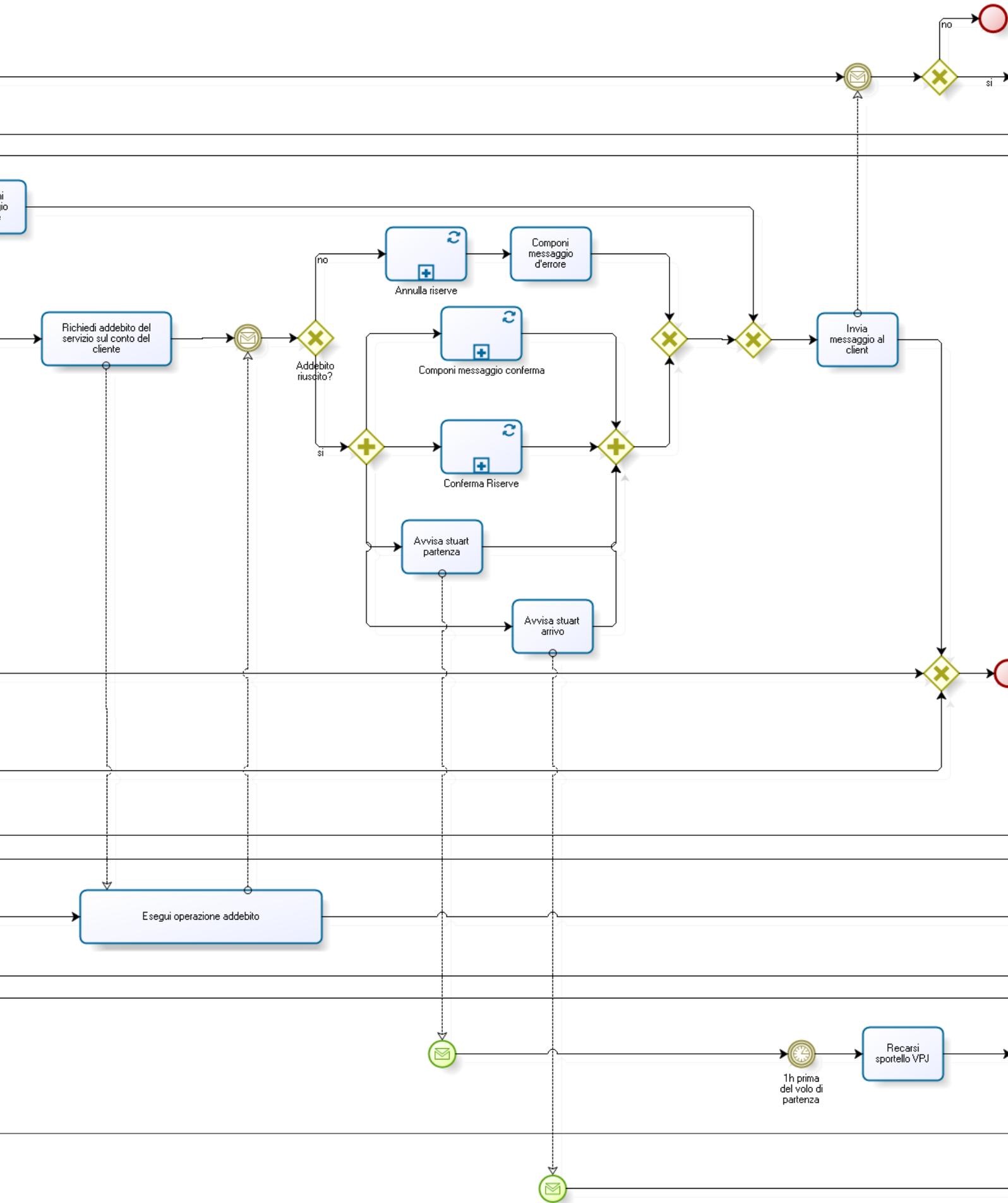
Nelle prossime pagine saranno riportati i BPMN dell'intero processo e di tutti i sub-process che utilizza.

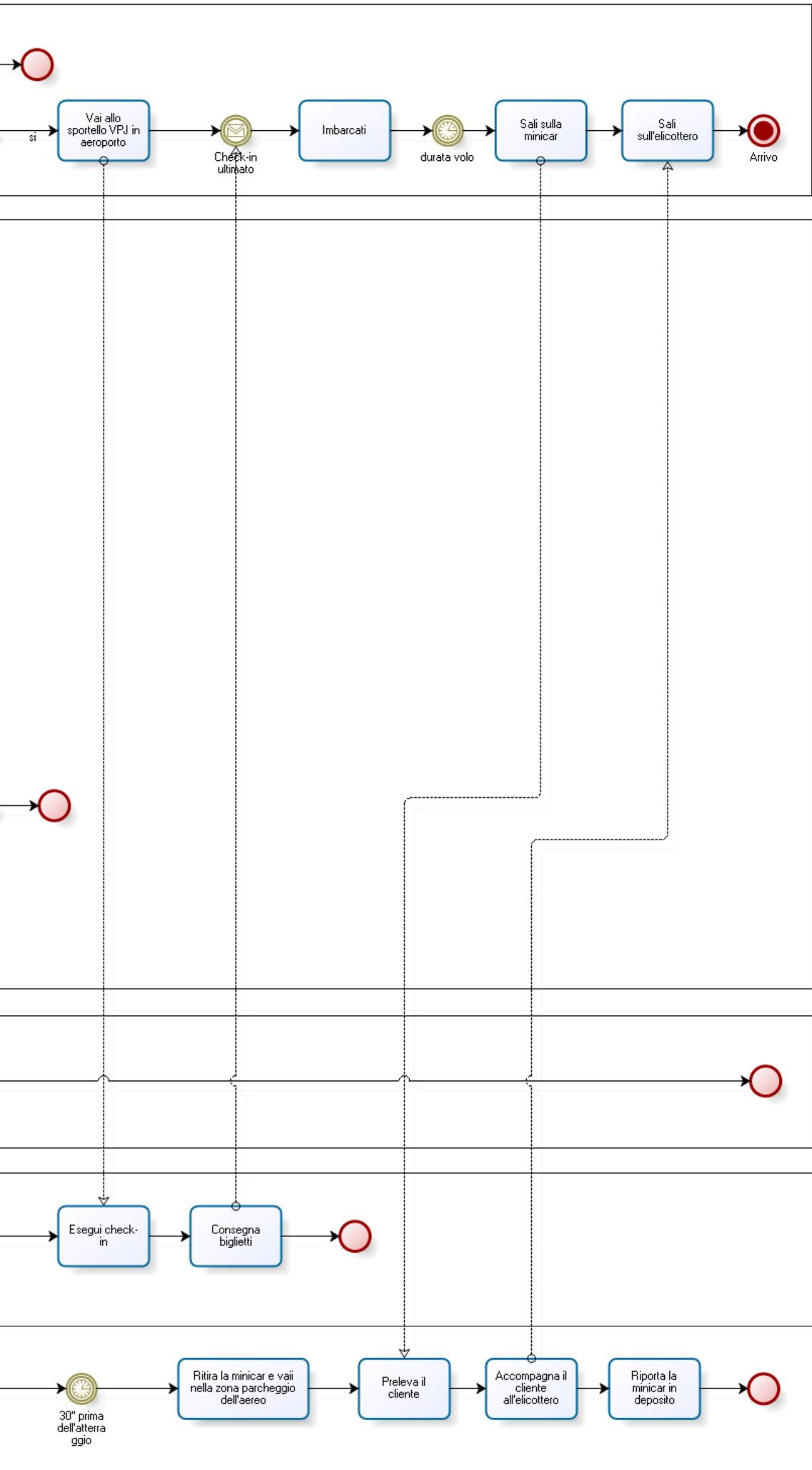
# Sub-process: Processo prenotazioni



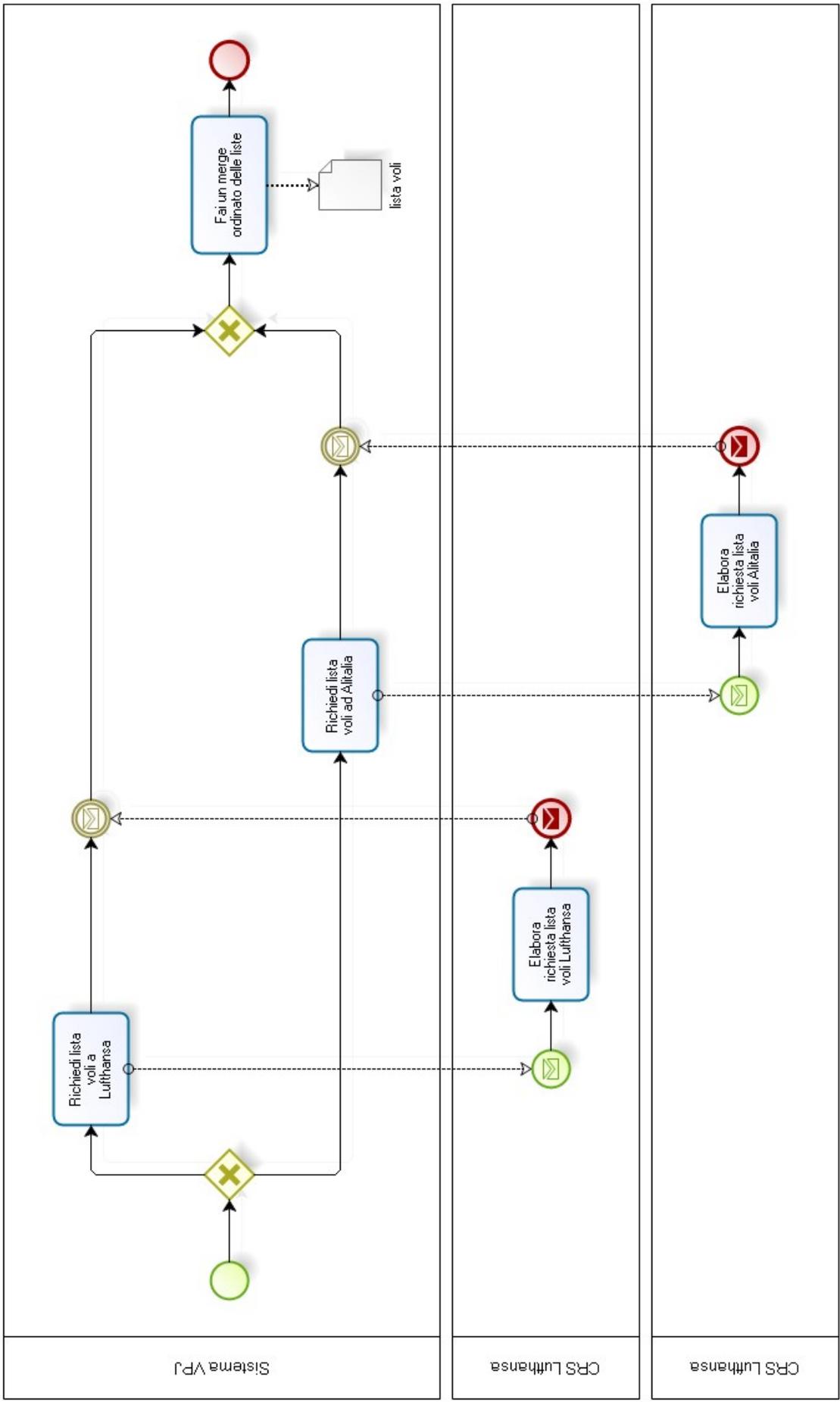




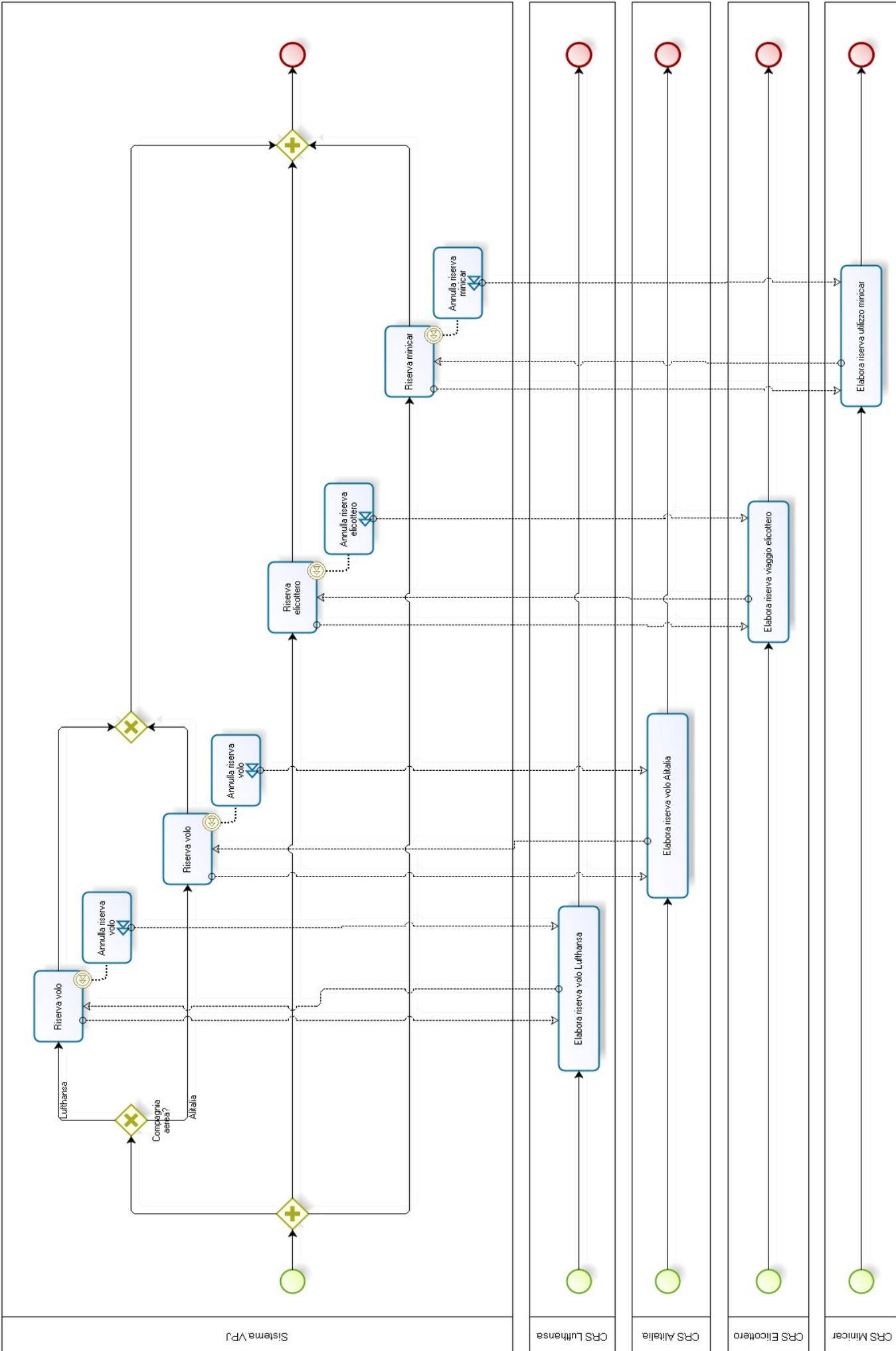




# Sub-process: Genera lista voli



# Sub-process: Verifica disponibilità veicoli



Sistema VPJ

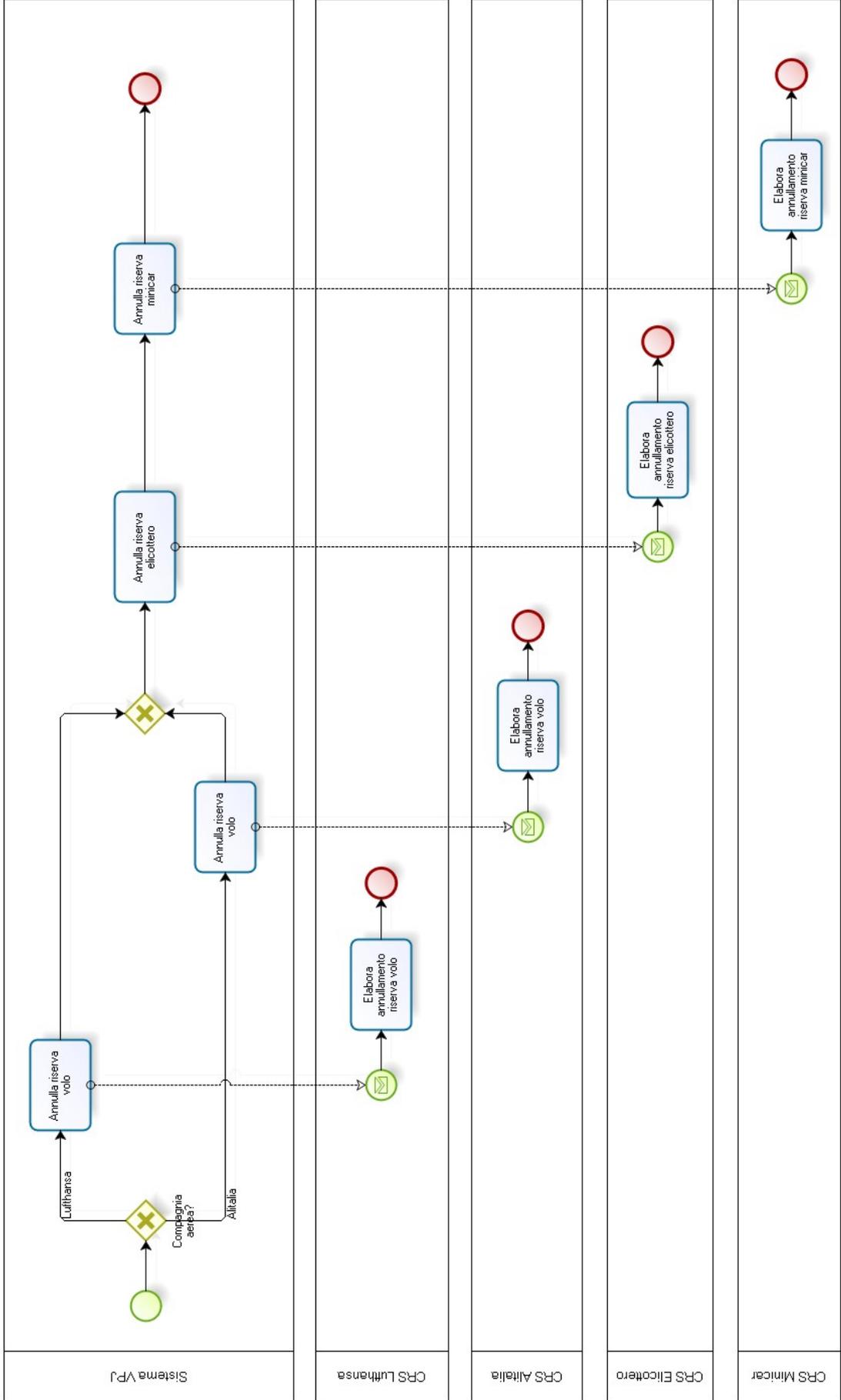
CRS Lufthansa

CRS Alitalia

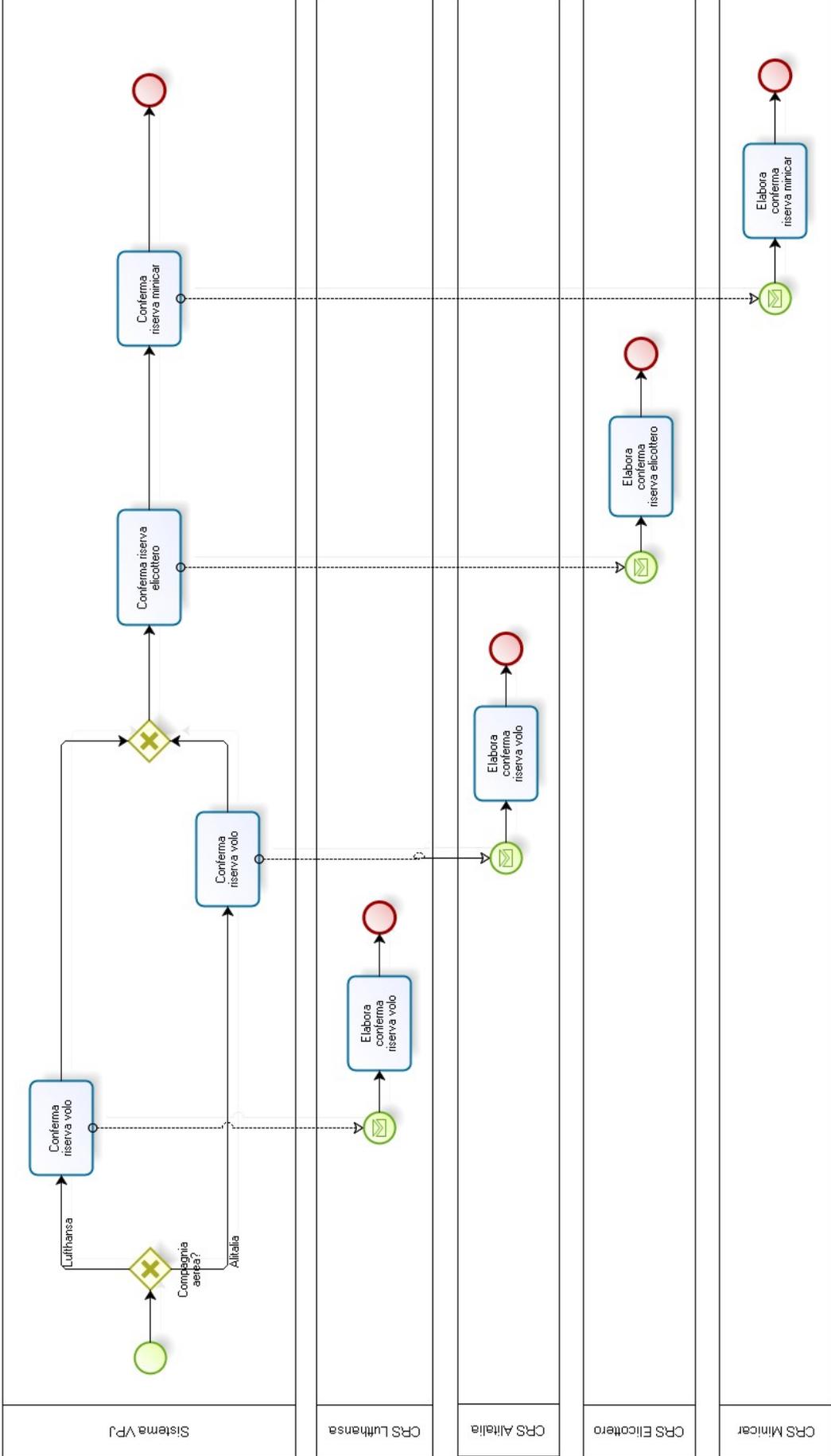
CRS Elicottero

CRS Minicar

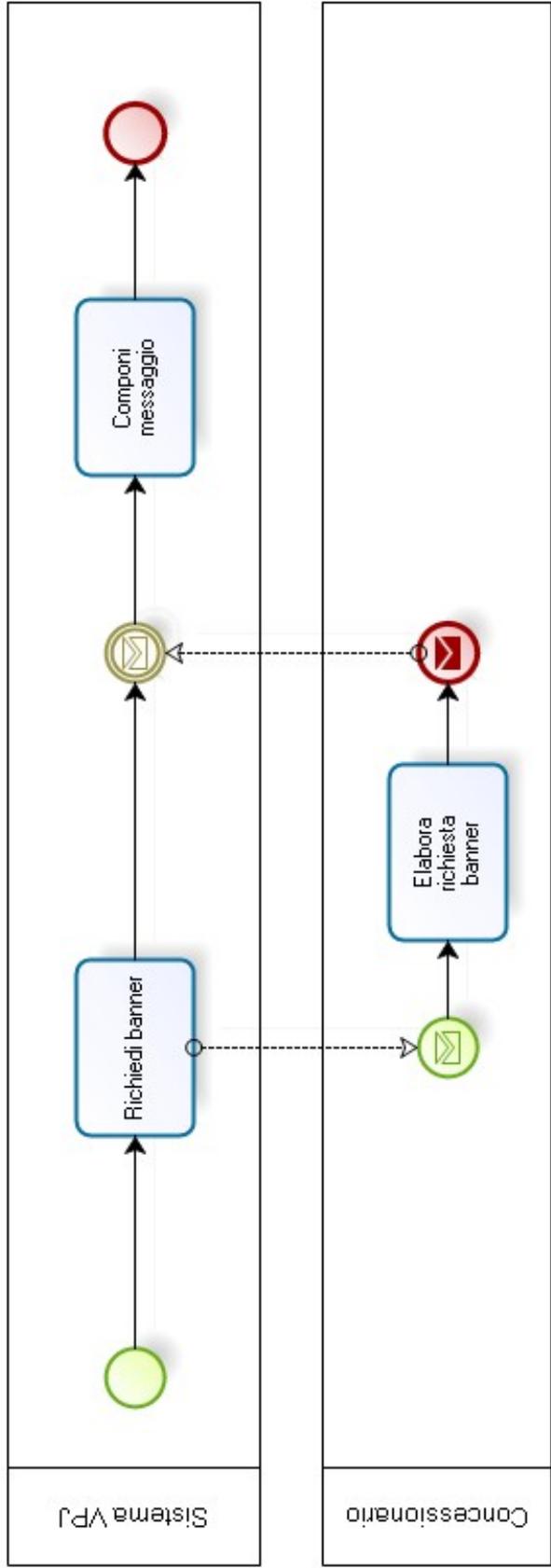
# Sub-process: Annulla riserve



# Sub-process: Conferma riserve



# Sub-process: Componi messaggio conferma



## Analisi qualità GQM

---

Per la misurazione della qualità del servizio VPJ si seguirà l'approccio **Goal Question Metric (GQM)**, uno dei paradigmi più utilizzati per la derivazione di misure e metriche connesse agli obiettivi aziendali. In questo progetto ci concentreremo sulla fase di definizione dell'approccio, ovvero:

1. individuare un insieme di *goal* misurabili;
2. generare per ogni *goal* una serie di *domande* che aiutino a definirlo e interpretarlo in modo quantitativo;
3. specificare quali *metriche* è necessario raccogliere per rispondere alle domande, quantificando così la distanza dal *goal*.

I goal individuati sono:

- **business goal:** ottenere una buona penetrazione nel mercato e un buon apprezzamento del servizio;
- **goal disponibilità:** impiegare una quantità ottimale di risorse, individuando quelle scarse;
- **goal attesa:** ridurre al minimo il tempo di attesa del cliente dalla prenotazione del servizio alla partenza del volo.

Nei prossimi paragrafi saranno mostrate domande e metriche per ogni obiettivo. I dati saranno prelevati dai log dell'applicazione e dalle risposte ad un questionario sulla qualità sottoposto ad ogni cliente a fine viaggio.

## 1. Legenda metriche

- (C.R.) = clienti registrati;
- (C.1volo) = clienti che hanno effettuato almeno un volo;
- (C.A.S.) = clienti che hanno acceduto al servizio;
- (C.T.P.) = clienti che terminano il processo di prenotazione e viaggio;
- (V.N.C.) = voli che non rispettano il contratto sui tempi di viaggio;
- (M.E.C.S.) = mancate entrate per gli sconti sull'offerta di viaggio.

## 2. Business goal: Question & Metriche

Q.1 *“Il servizio ha registrato consensi?”*

M.1 **#(C.R.)**

M.2 **%(C.1volo)** =  $\#(\text{C.1volo}) / \#(\text{C.R.})$

Q.2 *“Il servizio è utile per i clienti?”*

M.1 **#(C.A.S.)** =  $\#(\text{“Calcola tratte”})$

M.2 **%(C.T.P.)** =  $\#(\text{“Conferma riserve”}) / \#(\text{“Calcola tratte”})$

Q.3 *“I clienti sono soddisfatti del servizio?”*

M.1 **#(voli per cliente)**

M.2 **(valutazione questionario)**

### 3. Goal disponibilità: Question & Metriche

Q.1 *“È necessario ampliare le risorse?”*

M.1 **#(fail)** = #(rollback “Verifica disponibilità veicoli”)

M.2 **%(fail elicottero)** = #(rollback “Riserva elicottero”) / #(fail)

M.3 **%(fail minicar)** = #(rollback “Riserva minicar”) / #(fail)

M.4 **%(V.N.C.)** = # (“Aggiungi sconto”) / # (“Invia offerta al cliente”)

M.5 **(M.E.C.S.)** =  $\sum_{\text{prenotazioni scontate}} (\text{costo} - \text{costoScontato})$

### 4. Goal attesa: Question & Metriche

Q.1 *“Quanto rispettiamo gli accordi col cliente?”*

M.1 **%(contratti rispettati)** = 
$$\frac{\# \text{prenotazioni} - \#(\text{prenotazioni scontate})}{\# \text{prenotazioni}}$$

M.2 **#(prenotazioni scontate)**

Q.2. *“I tempi d'attesa sono competitivi?”*

M.1 **(tempo attesa medio)**

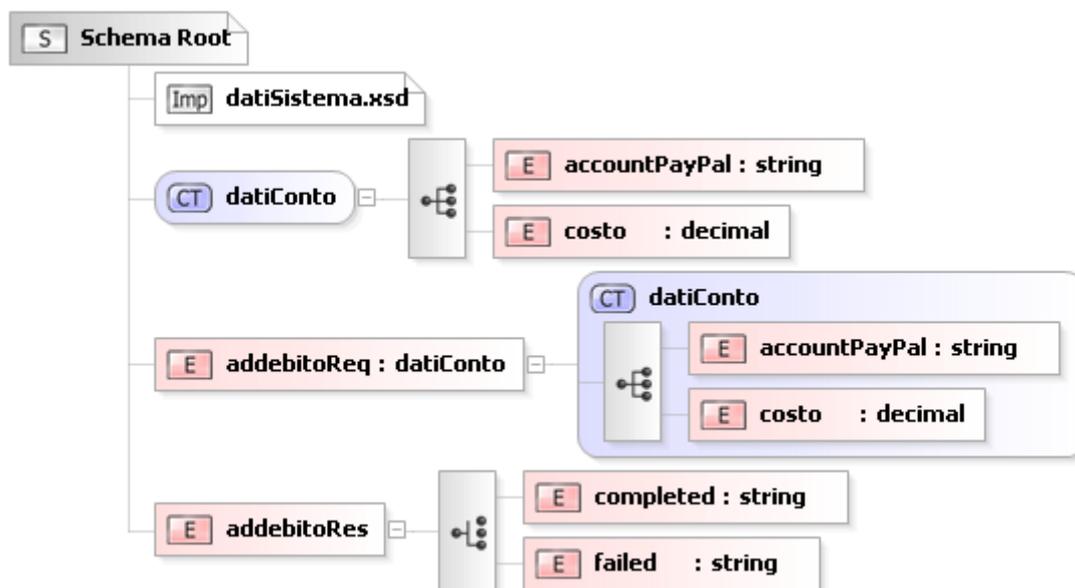
M.2 **(deviazione standard tempo attesa)**

## Descrizione servizi BPEL

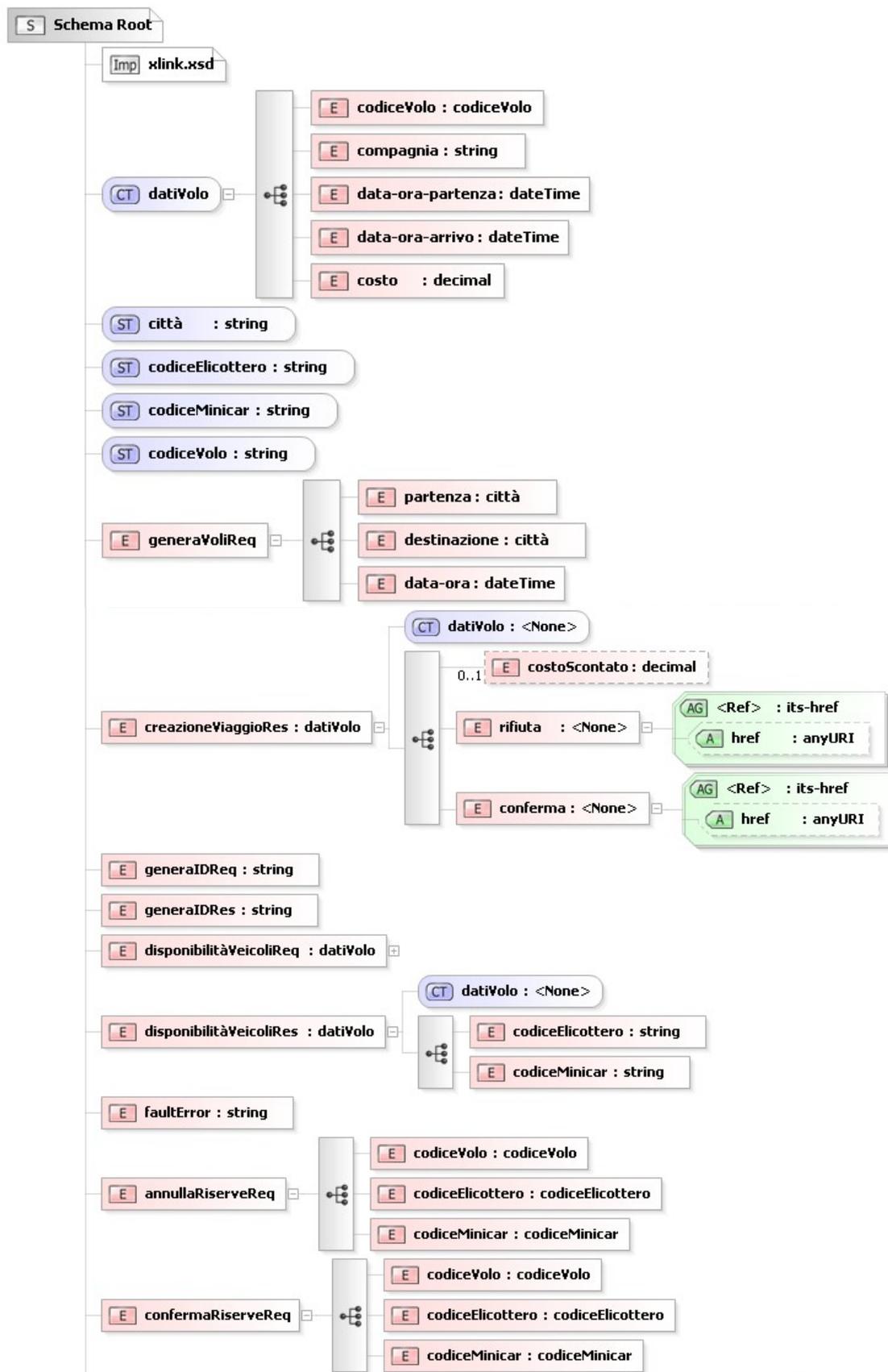
Parte del processo è stato descritto in BPEL per coordinare i servizi necessari alla sua esecuzione. Di seguito sono riportati:

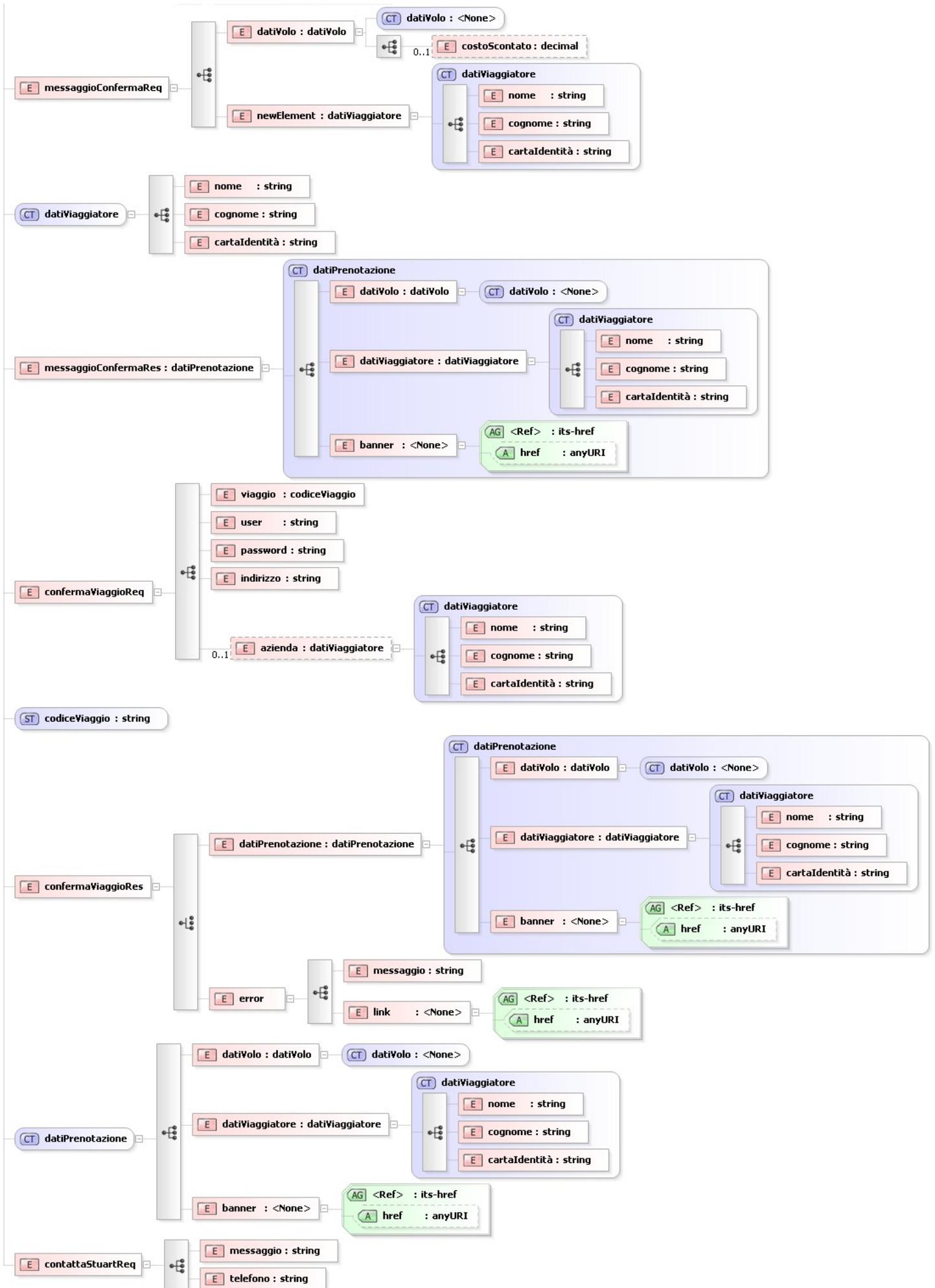
- lo schema dei messaggi scambiati tra i servizi;
- i partner link dei servizi invocati, descritti nel wsdl;
- i diagrammi BPEL del sottoprocesso di creazione del viaggio (che ha inizio con la richiesta di una risorsa `"/tratte/[nome-aeroporto]-[nome-aeroporto]"`) e del sottoprocesso di conferma del viaggio (che ha inizio con l'aggiornamento delle informazioni del viaggio tramite form).

### Schema dei messaggi scambiati con i servizi esterni



## Schema dei messaggi scambiati tra i servizi del sistema





## Partner link dei servizi esterni

### Partner Link: **payment**

**Input:** paymentReq

```
request
  ■ accountPayPal
  ■ costo
```

**Output:** paymentRes

```
response
  ■ completed
  ■ failed
```

## Partner link dei servizi del sistema

### Partner Link: **annullaRiserve**

**Input:** annullaRiserveReq

```
request
  ■ codiceVolo
  ■ codiceElicottero
  ■ codiceMinicar
```

### Partner Link: **confermaRiserve**

**Input:** confermaRiserveReq

```
request
  ■ codiceVolo
  ■ codiceElicottero
  ■ codiceMinicar
```

**Partner Link: confermaViaggio****Input:** confermaViaggioReq

```
request
  ■ viaggio
  ■ user
  ■ password
  ■ indirizzo
  ■ azienda
    ■ nome
    ■ cognome
    ■ cartalidentità
```

**Output:** confermaViaggioRes

```
response
  ■ datiPrenotazione
    ■ datiVolo
      ■ codiceVolo
      ■ compagnia
      ■ data-ora-partenza
      ■ data-ora-arrivo
      ■ costo
    ■ datiViaggiatore
      ■ nome
      ■ cognome
      ■ cartalidentità
    ■ banner
  ■ error
    ■ messaggio
    ■ link
```

**Partner Link: contattaStuart****Input:** contattaStuartReq

```
request
  ■ messaggio
  ■ telefono
```

**Partner Link: creazioneViaggio****Input:** creazioneViaggioReq

```
request
  ■ partenza
  ■ destinazione
```

**Output:** creazioneViaggioRes

```
response
  ■ codiceVolo
  ■ compagnia
  ■ data-ora-partenza
  ■ data-ora-arrivo
  ■ costo
  ■ costoScontato
  ■ rifiuta
  ■ conferma
```

**Partner Link: disponibilitàVeicoli****Input:** disponibilitàVeicoliReq

```
request
  ■ codiceVolo
  ■ compagnia
  ■ data-ora-partenza
  ■ data-ora-arrivo
  ■ costo
```

**Output:** disponibilitàVeicoliRes

```
response
  ■ codiceVolo
  ■ compagnia
  ■ data-ora-partenza
  ■ data-ora-arrivo
  ■ costo
  ■ codiceElicottero
  ■ codiceMinicar
```

**Fault:** disponibilitàVeicoliFault

```
fault
```

**Partner Link: generalD****Input:** generalDReq

```
generalDReq
```

**Output:** generalDRes

```
generalDRes
```

**Partner Link: generaVoli****Input:** generaVoliReq

```
generaVoliReq
  ■ partenza
  ■ destinazione
  ■ data-ora
```

**Output:** generaVoliRes

```
generaListaRes
  ■ volo
    ■ codiceVolo
    ■ compagnia
    ■ data-ora-partenza
    ■ data-ora-arrivo
    ■ costo
```

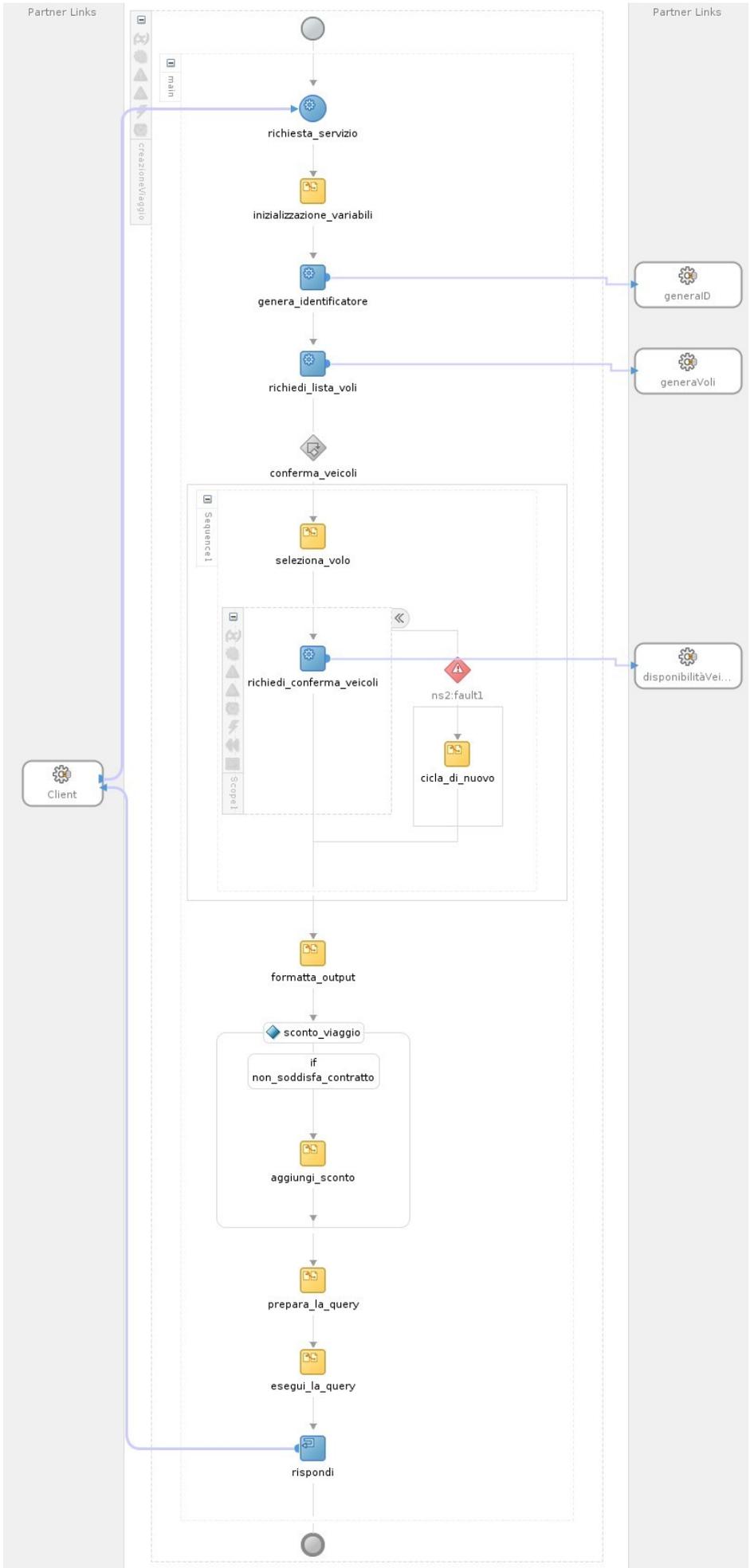
**Partner Link: messaggioConferma****Input:** messaggioConfermaReq

```
request
  ■ datiVolo
    ■ codiceVolo
    ■ compagnia
    ■ data-ora-partenza
    ■ data-ora-arrivo
    ■ costo
    ■ costoScontato
  ■ newElement
    ■ nome
    ■ cognome
    ■ cartaIdentità
```

**Output:** messaggioConfermaRes

```
response
  ■ datiVolo
    ■ codiceVolo
    ■ compagnia
    ■ data-ora-partenza
    ■ data-ora-arrivo
    ■ costo
  ■ datiViaggiatore
    ■ nome
    ■ cognome
    ■ cartaIdentità
  ■ banner
```

# BPEL: Creazione Viaggio



# BPEL: Conferma Viaggio

