

SISTEMA OCA MVM ASSICURAZIONI ARCHITETTURA DEL SOFTWARE



701637 SGURA Marco
829992 GRAVAGNA Vincenzo Sergio
830287 GIUDETTI Mario Giuseppe

Indice

1. Traccia
2. Assunzioni
3. Architettura del problema
4. Architettura logica
5. Architettura concreta
6. Architettura di deployment

Obiettivo del progetto

Si chiede di realizzare un sistema di Osservazione del Comportamento degli Automobilisti (sistema OCA) che consenta a una Compagnia di assicurazione di monitorare il comportamento degli automobilisti assicurati per ridefinire le polizze in base al livello di rischio di ciascun assicurato e per fornire assistenza in caso di sinistro.

Sui veicoli degli assicurati sono presenti:

- a. Un sensore di accelerazione ACC (accelerometro);
- b. Un sensore di velocità TAC (tachimetro);
- c. Un sensore GPS.

Obiettivo del progetto

Il sistema OCA deve essere in grado di:

1. Supportare l'assistenza immediata in caso di sinistri:
 - a. Riconoscimento di eventuali sinistri (collisioni);
 - b. Notifica all'operatore competente delle informazioni necessarie per organizzare l'assistenza;
 - c. Attivazione di una connessione telefonica tra operatore e assicurato coinvolto.
2. Determinare per ciascuna polizza su base annuale:
 - a. Il chilometraggio totale K ;
 - b. La velocità media V per ciascuna tipologia T di tronco viario;
 - c. Il livello di prudenza P , definito come la percentuale di chilometri percorsi rispettando i limiti di velocità con una tolleranza di 10 km/h.

Obiettivo del progetto

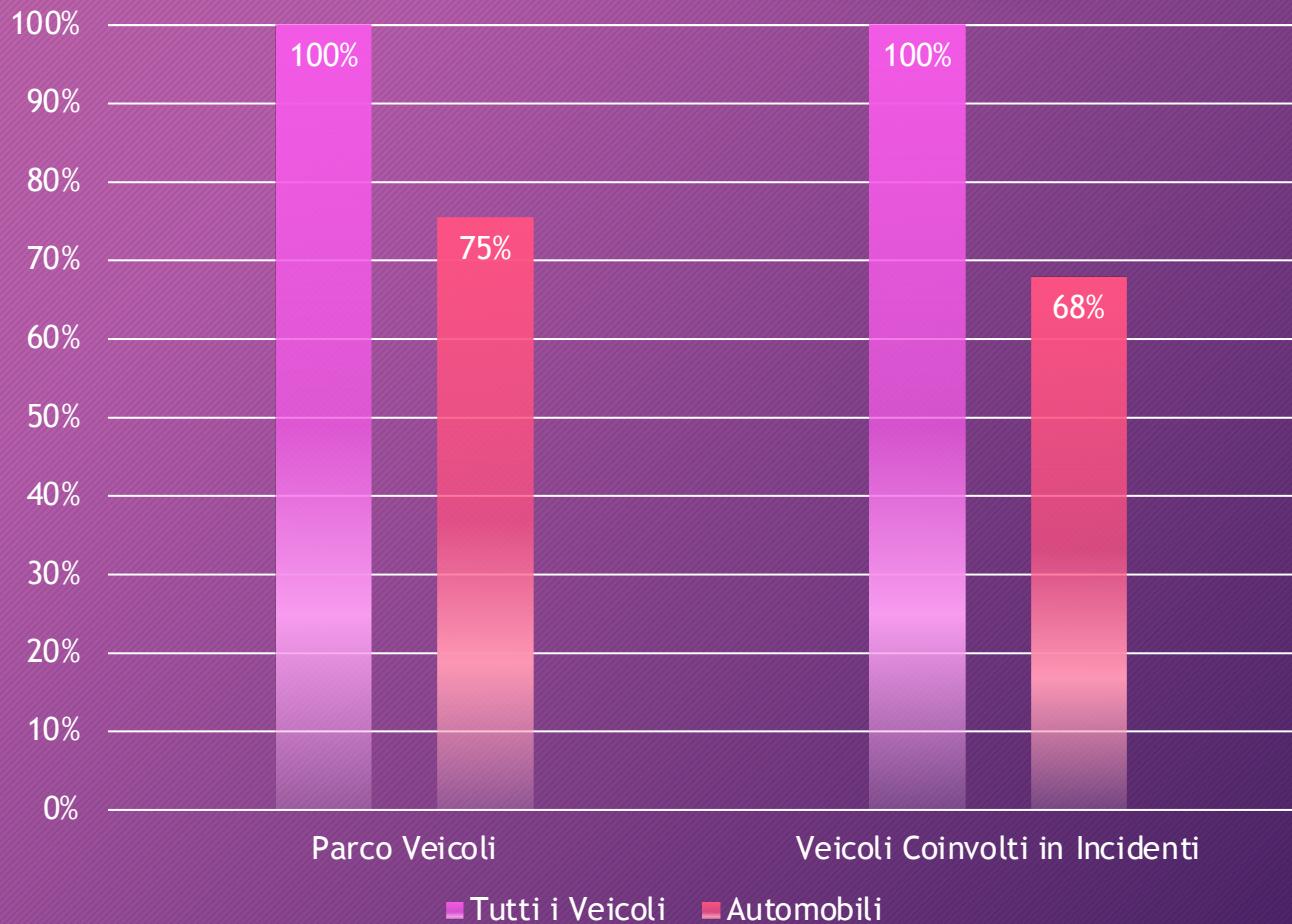
3. Calcolare il livello di rischio R di ciascun assicurato come $R = f(K, V, P, E, D, C)$, dove:
 - a. K, V, P sono cumulativi per tutte le polizze dell'assicurato;
 - b. E è l'età dell'assicurato;
 - c. D il numero di denunce sinistro nell'ultimo anno;
 - d. C il Comune di residenza;
 - e. R può assumere valori discreti da 1 a 10.
4. Consentire all'operatore competente di visualizzare, per ciascun assicurato e per ciascuna polizza, le attuali condizioni di polizza e il livello di rischio dell'assicurato.

Assunzioni

Dati Istat al 2019 in Italia

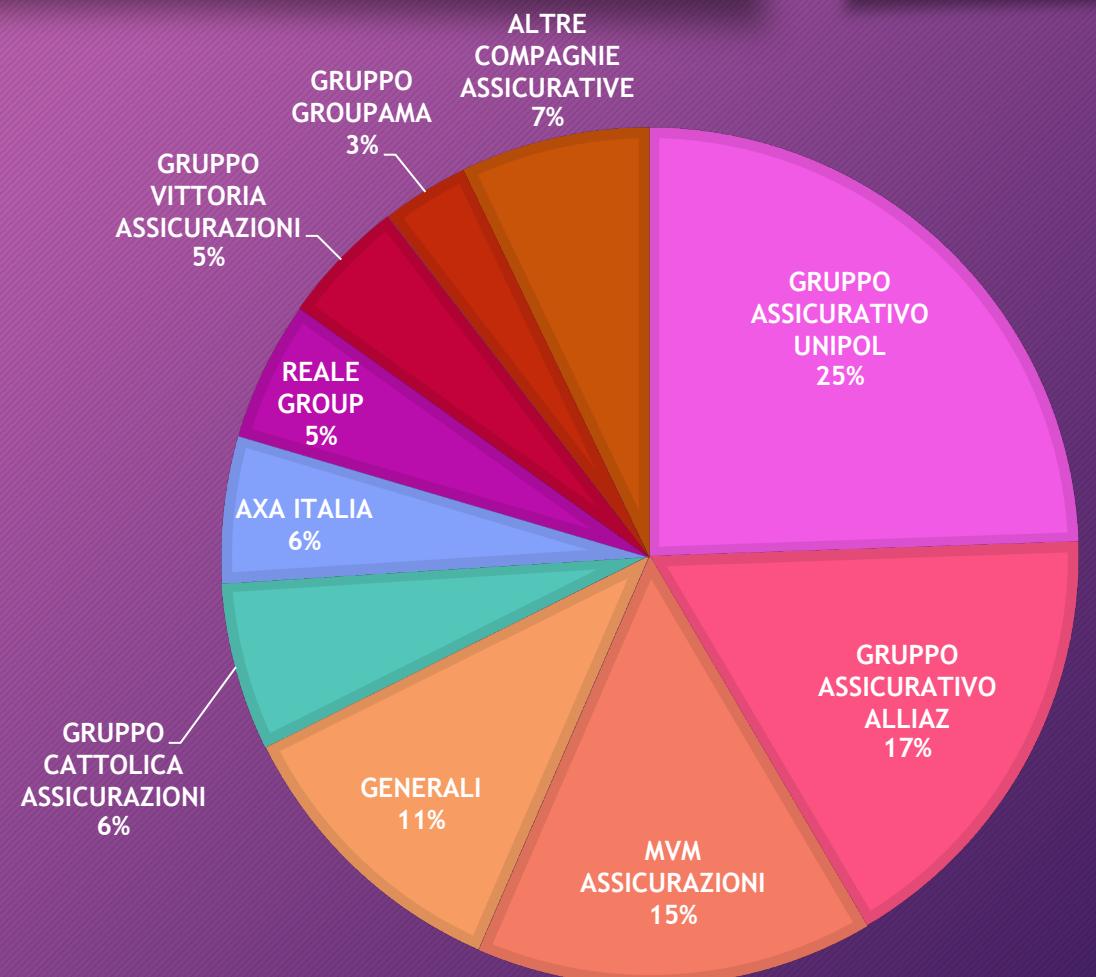
Nel 2019 il parco circolante in Italia contava **52.401.299** veicoli, di cui **39.545.232** automobili.

Inoltre, i veicoli coinvolti in incidenti contavano **316.206** unità, di cui **214.772** automobili.



Dati ANIA al 2019 in Italia

Compagnia Assicurativa	Automobili	Incidenti
TOTALE	39.545.232	214.772
GRUPPO ASSICURATIVO UNIPOL	9.668.809	52.512
GRUPPO ASSICURATIVO ALLIAZ	6.754.326	36.683
MVM ASSICURAZIONI	5.931.785	32.216
GENERALI	4.405.833	23.928
GRUPPO CATTOLICA ASSICURAZIONI	2.511.617	13.641
AXA ITALIA	2.187.346	11.880
REALE GROUP	2.088.483	11.343
GRUPPO VITTORIA ASSICURAZIONI	1.870.984	10.161
GRUPPO GROUPAMA	1.313.396	7.133
ALTRE COMPAGNIE ASSICURATIVE	2.812.655	15.276



Elenco delle assunzioni

MVM ASSICURAZIONI:

- è la terza compagnia assicurativa più importante in Italia;
- ha contratti con circa il 15% delle automobili italiane (pari a 5.931.785);
- conta in media 32.216 automobili coinvolte in incidenti stradali all'anno (circa 88 al giorno);
- il numero dei clienti assicurati conta 3.954.523 persone;
- ha 932 sedi sparse in tutta Italia, una per ciascuno dei comuni italiani più popolosi;
- dispone di uno spazio cloud in cui sono presenti gli attori BDA e BDG.

Elenco delle assunzioni

Il sistema proposto:

- ha come unico target l'**RC Auto**;
- non gestisce la funzionalità di sottoscrizione di nuove polizze;
- effettua il rinnovo del premio di assicurazione automaticamente (senza un intervento dell'operatore) ad un mese dalla scadenza della polizza.

Il sensore di accelerazione invia 10 valori al secondo;

Il sensore GPS e di velocità inviano 1 valore ogni 5 secondi;

Vi è la possibilità di installare una scatola nera su ogni automobile nella quale sono registrati i dati del veicolo.

Architettura Del Problema

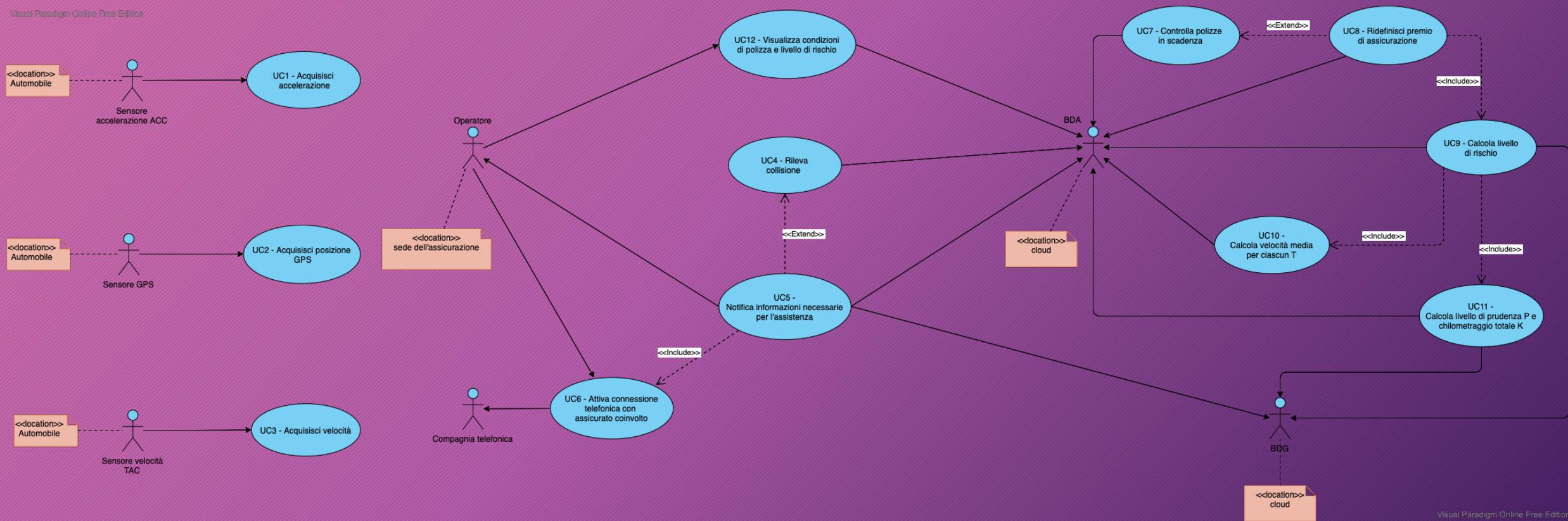
Diagramma dei casi d'uso

Modello dei dati

Diagrammi delle attività

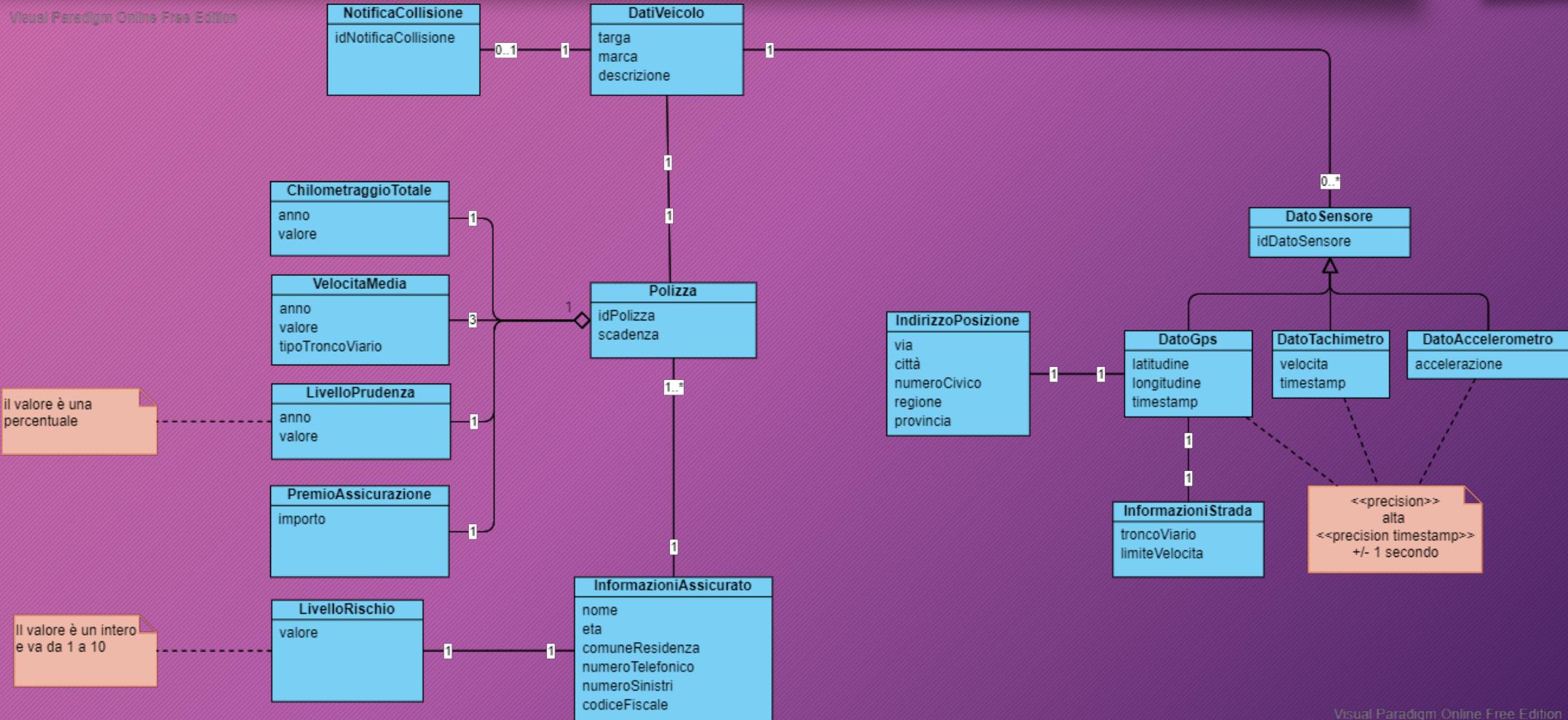
Diagramma dei casi d'uso

Visual Paradigm Online Free Edition

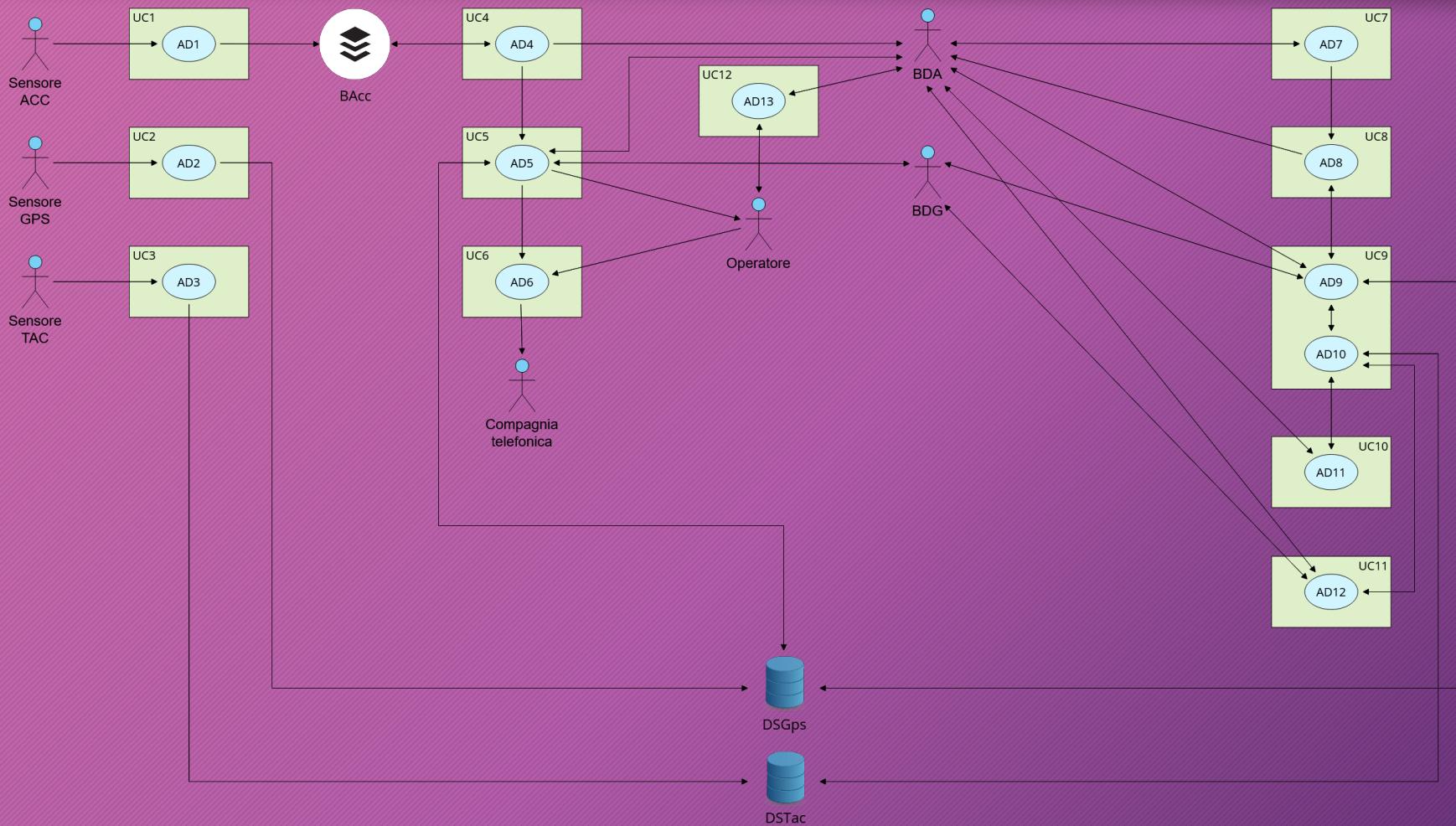


Visual Paradigm Online Free Edition

Modello dei dati



Mappa dei diagrammi delle attività



Vengono esplicitati i collegamenti tra i vari diagrammi delle attività e gli attori/datastore/buffer senza però fare distinzione tra flusso dati e flusso di controllo.

Legenda:



Attività



Caso d'uso



Attore

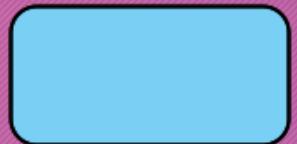


Buffer



Datastore

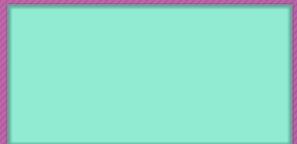
Legenda colori diagramma attività



Azione



Note: Complexity, Frequency, Delay, Nota I/O, Nota Funzionamento



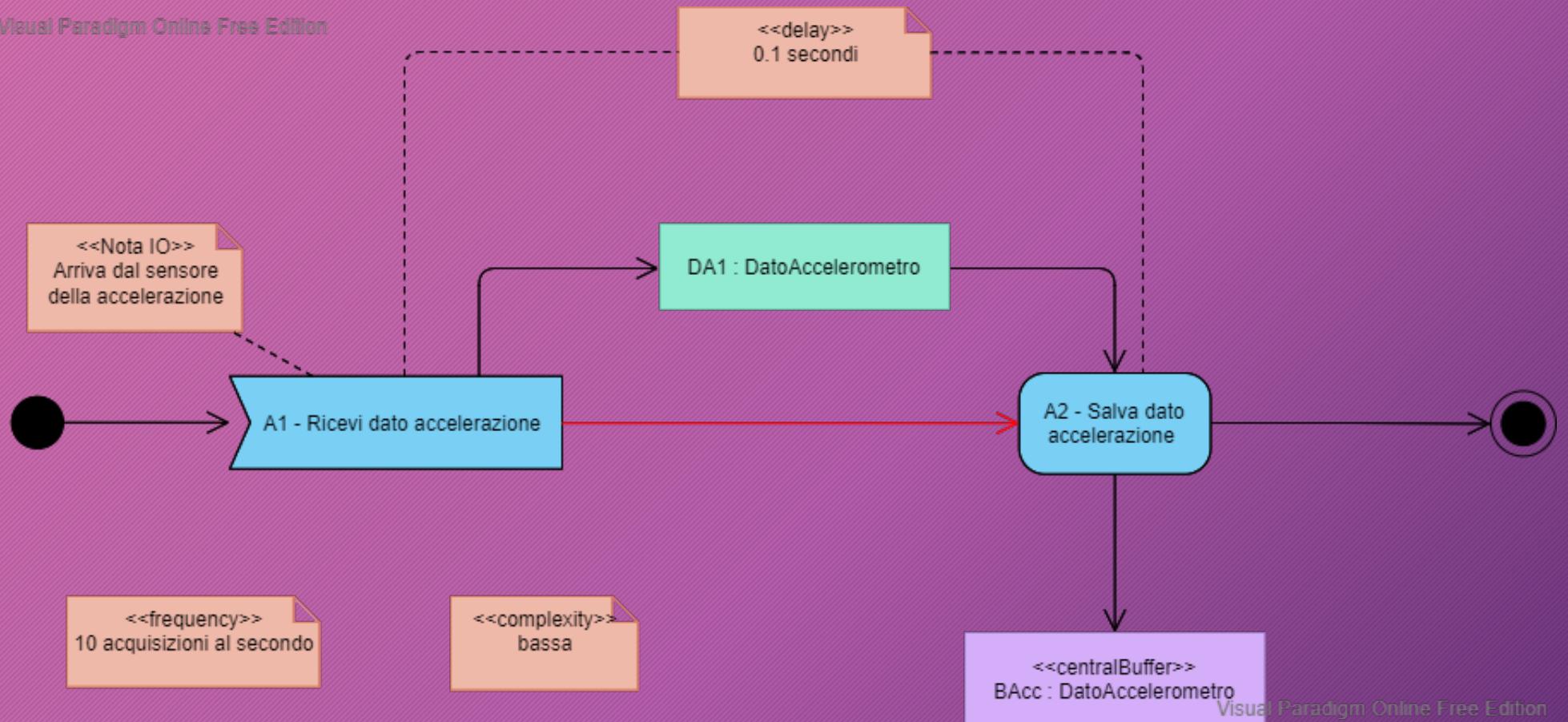
Oggetto o array di oggetti



Buffer

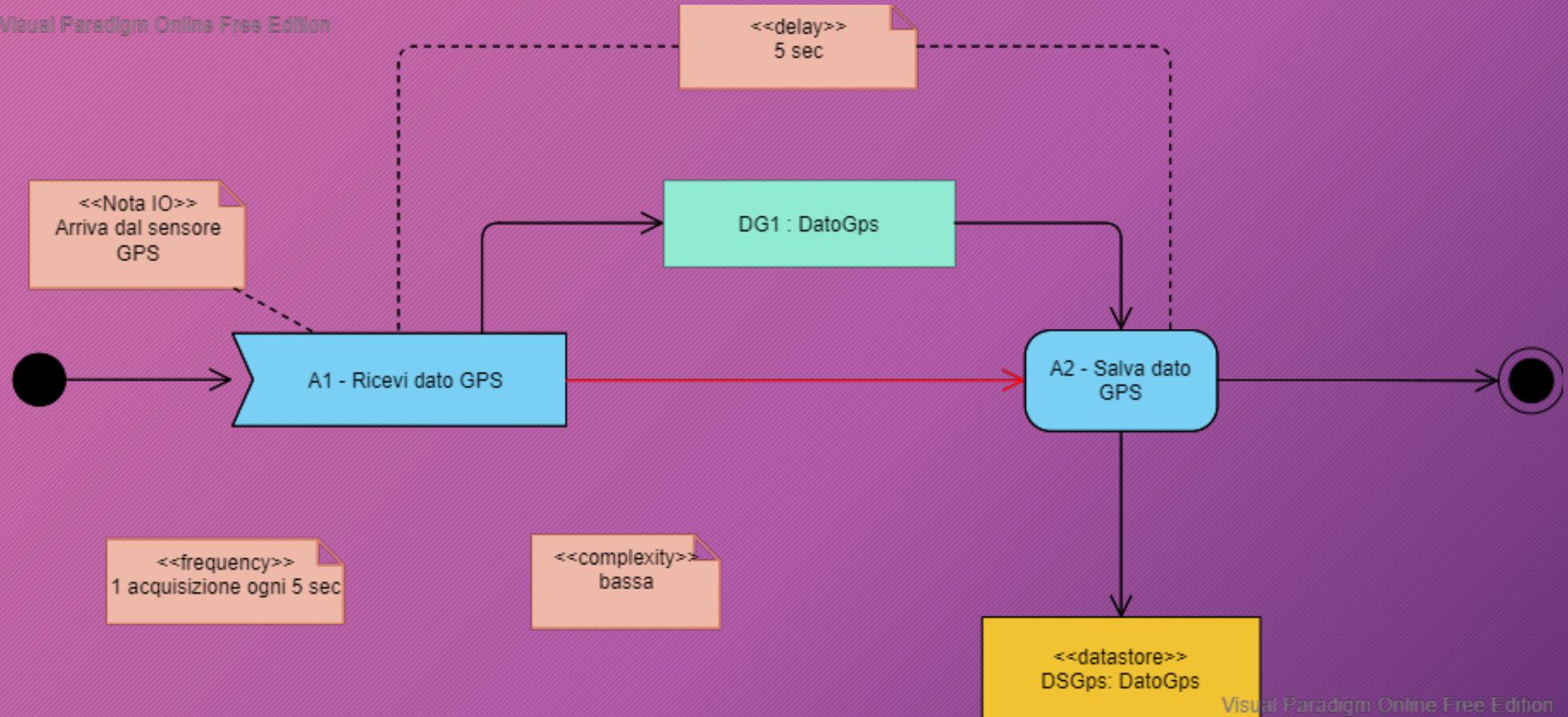


Datastore



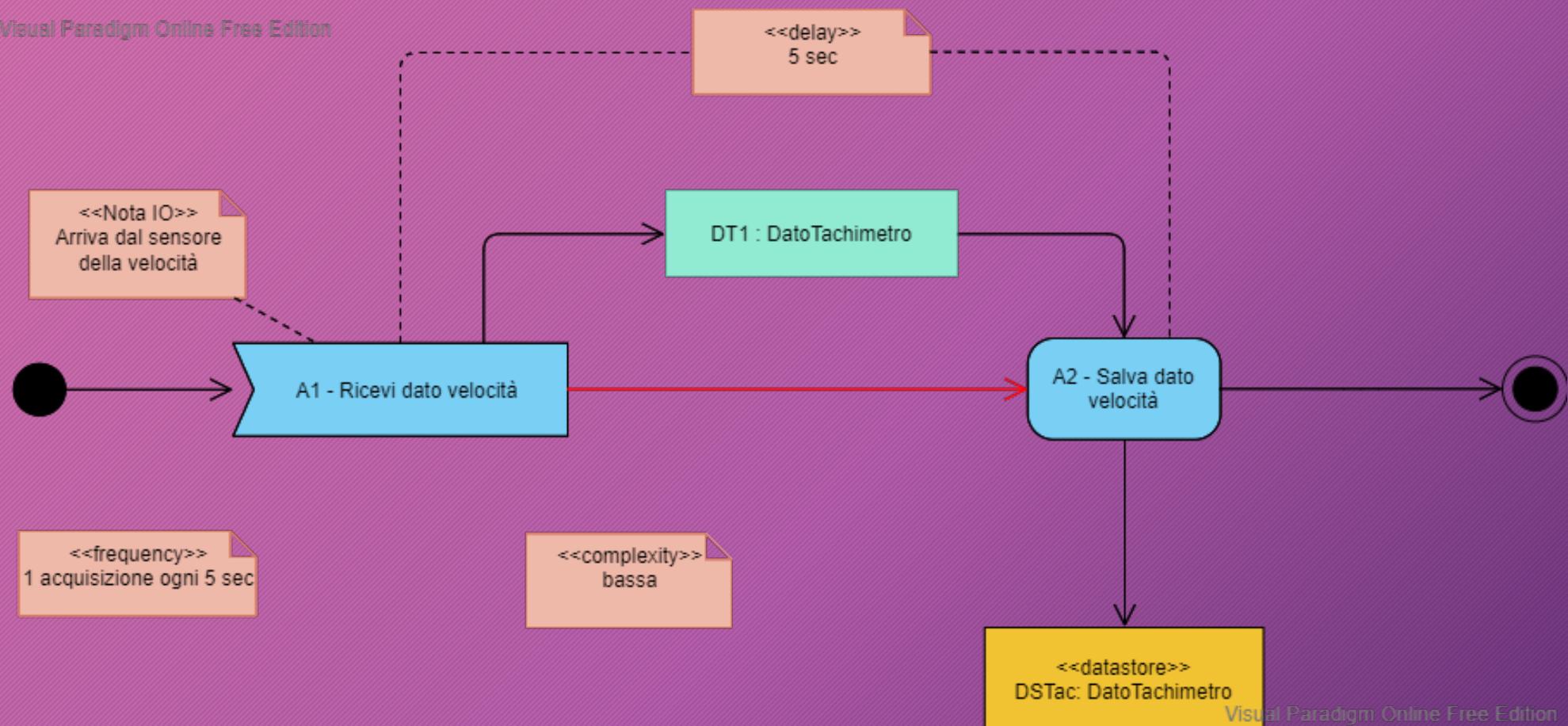
AD1 - Acquisisci accelerazione

UC1- Acquisisci accelerazione



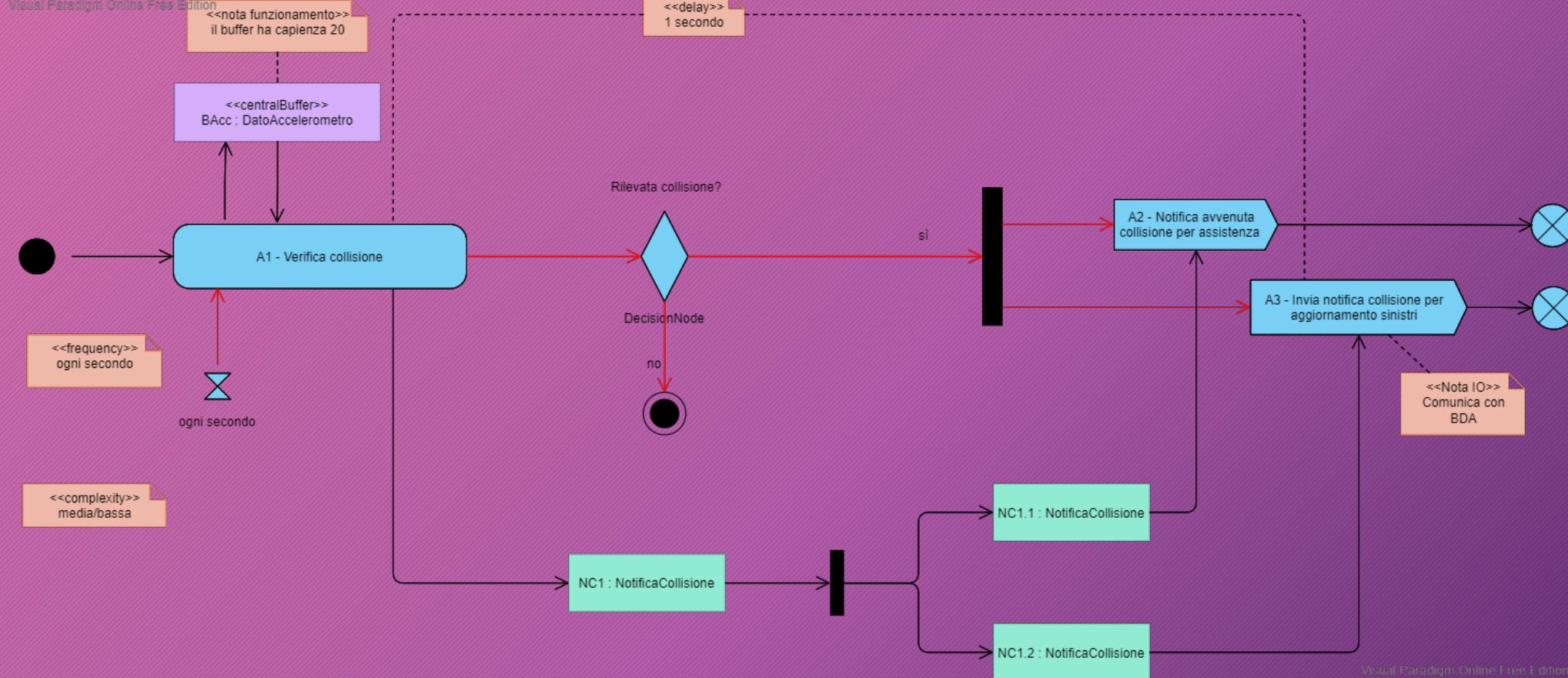
AD2 - Acquisisci posizione GPS

UC2- Acquisisci posizione GPS



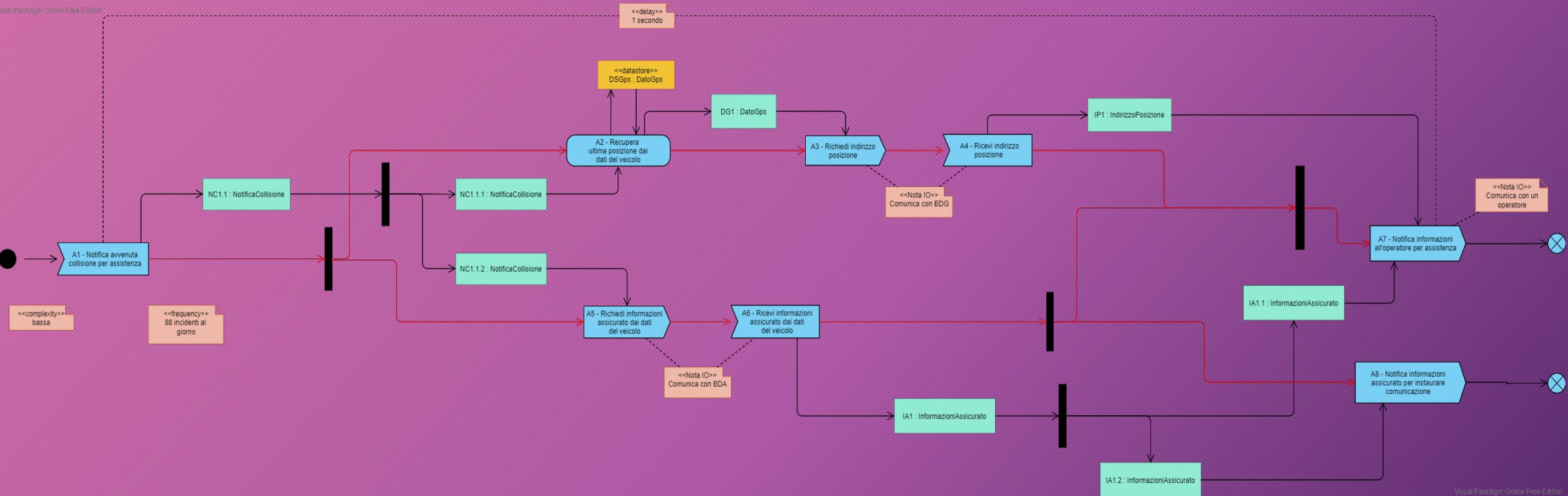
AD3 - Acquisisci velocità

UC3- Acquisisci velocità



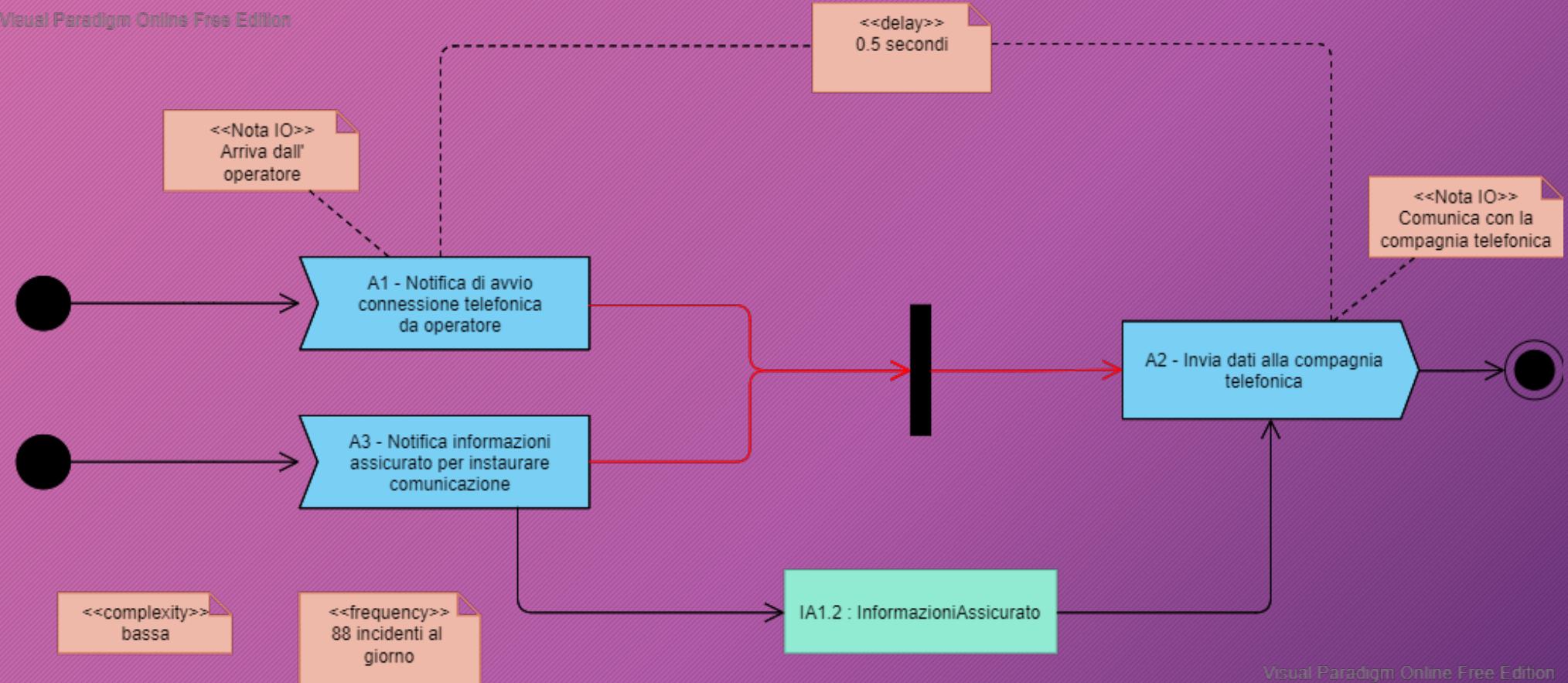
AD4 - Rileva collisione

UC4- Rileva collisione



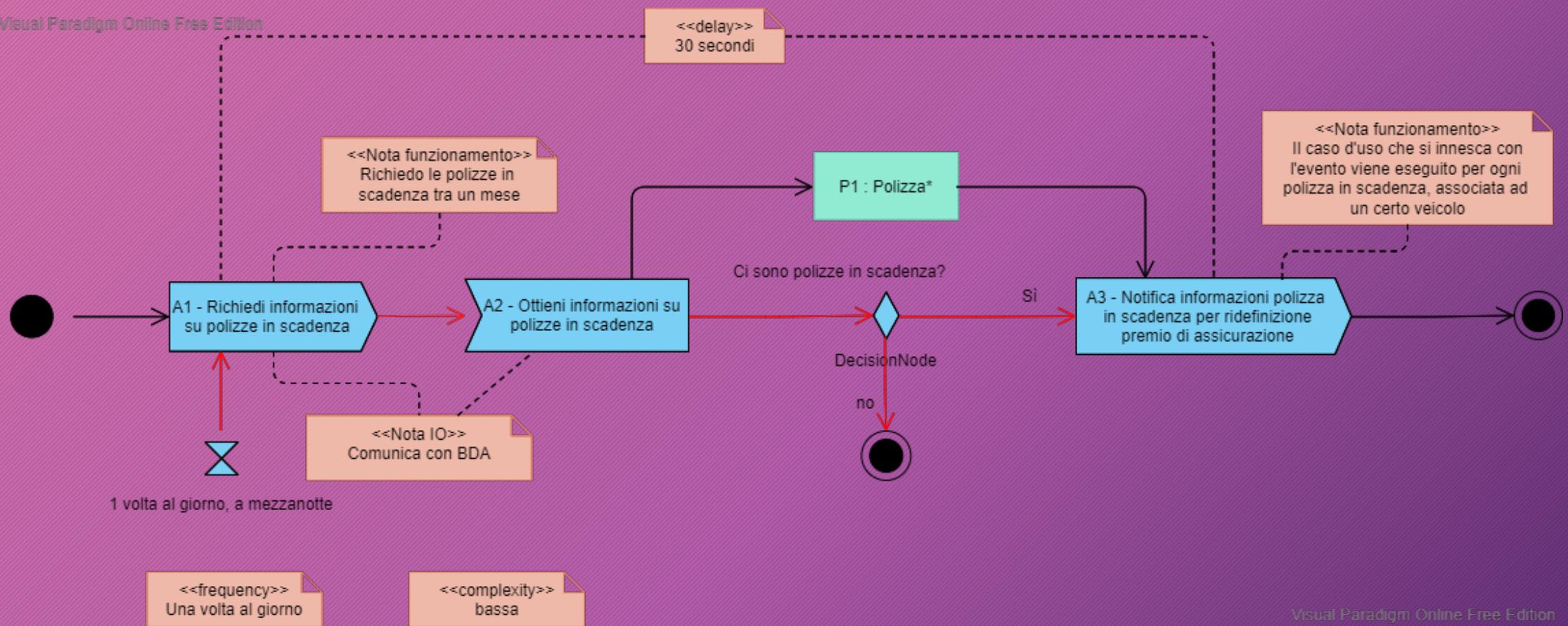
AD5 - Notifica informazioni necessarie per l'assistenza

UC5- Notifica informazioni necessarie per l'assistenza



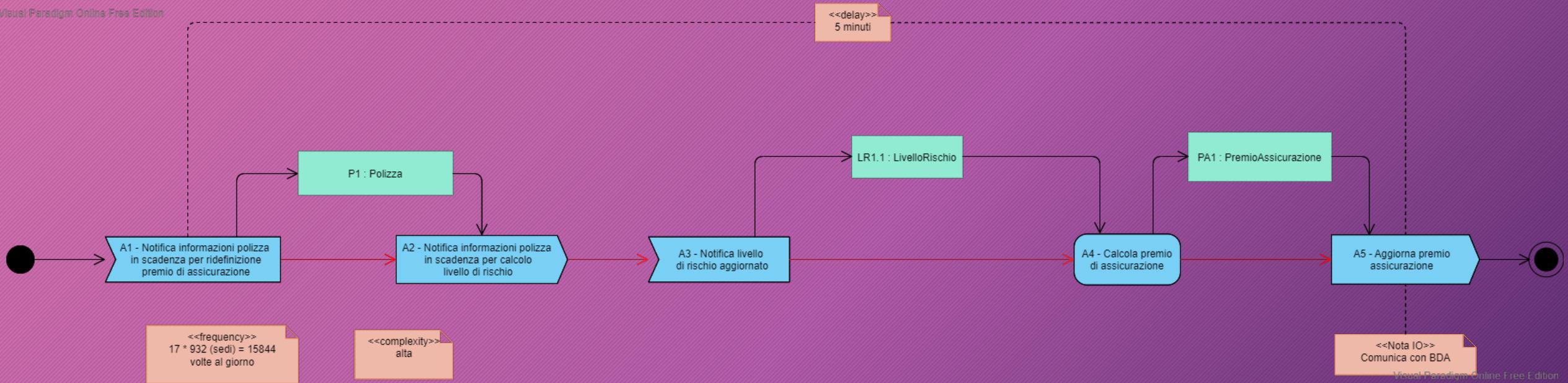
AD6 - Attiva connessione telefonica con assicurato coinvolto

UC6- Attiva connessione telefonica con assicurato coinvolto



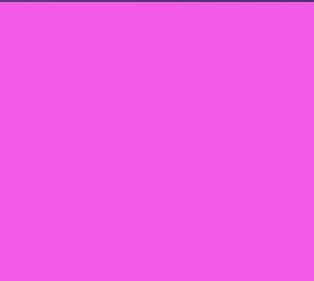
AD7 - Controlla polizze in scadenza

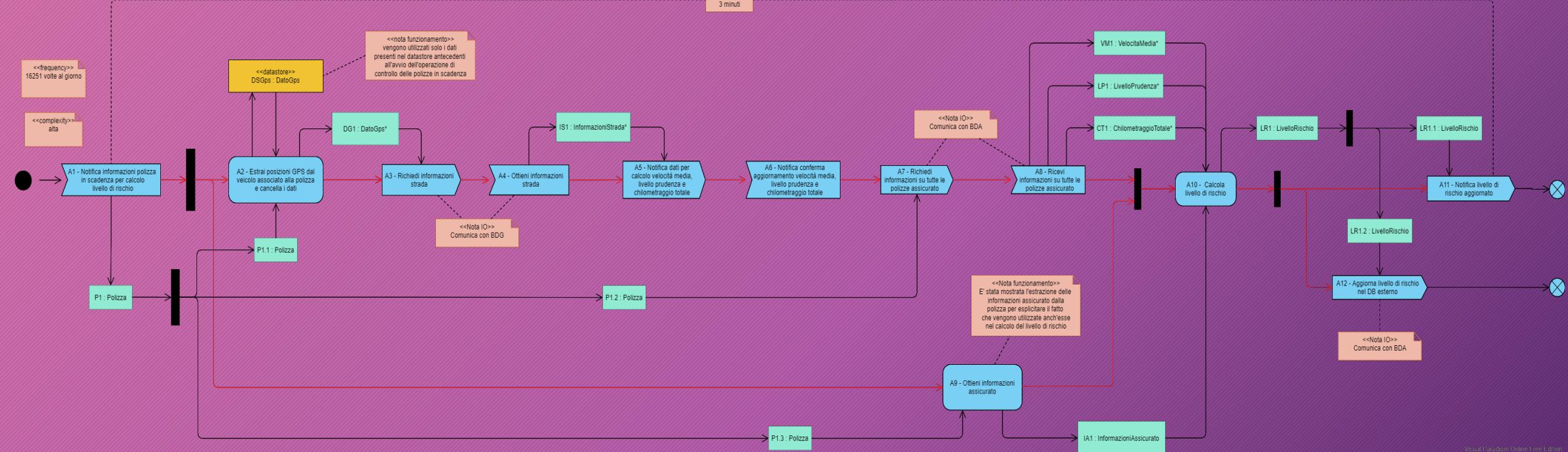
UC7- Controlla polizze in scadenza



AD8 - Ridefinisci premio di assicurazione

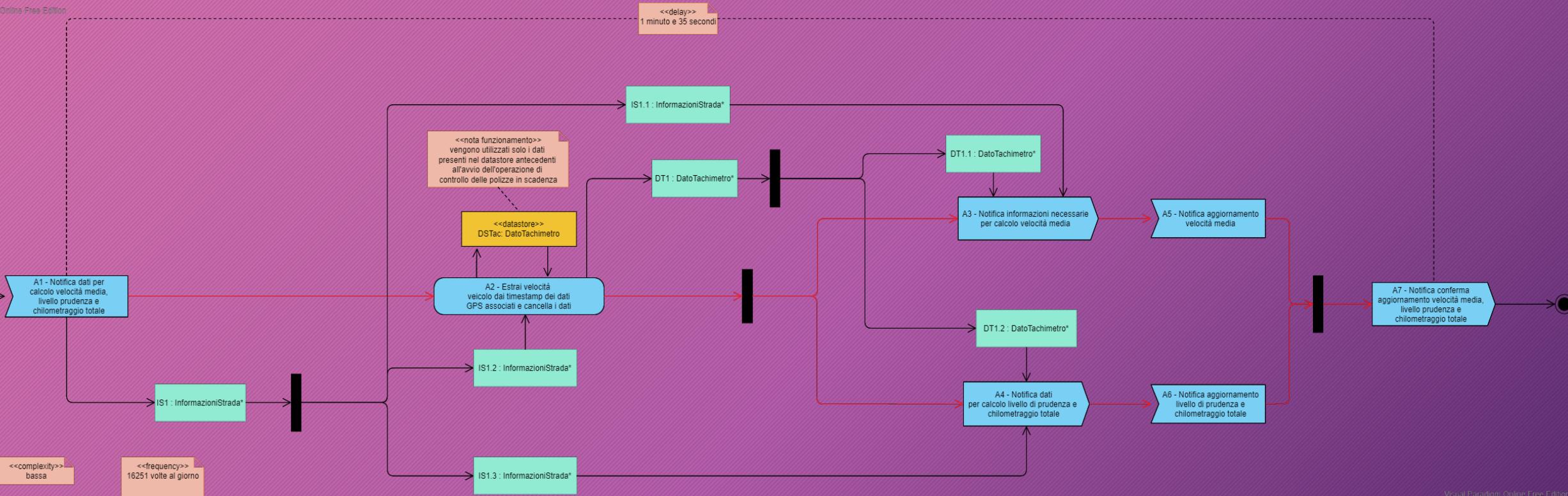
UC8- Ridefinisci premio di assicurazione





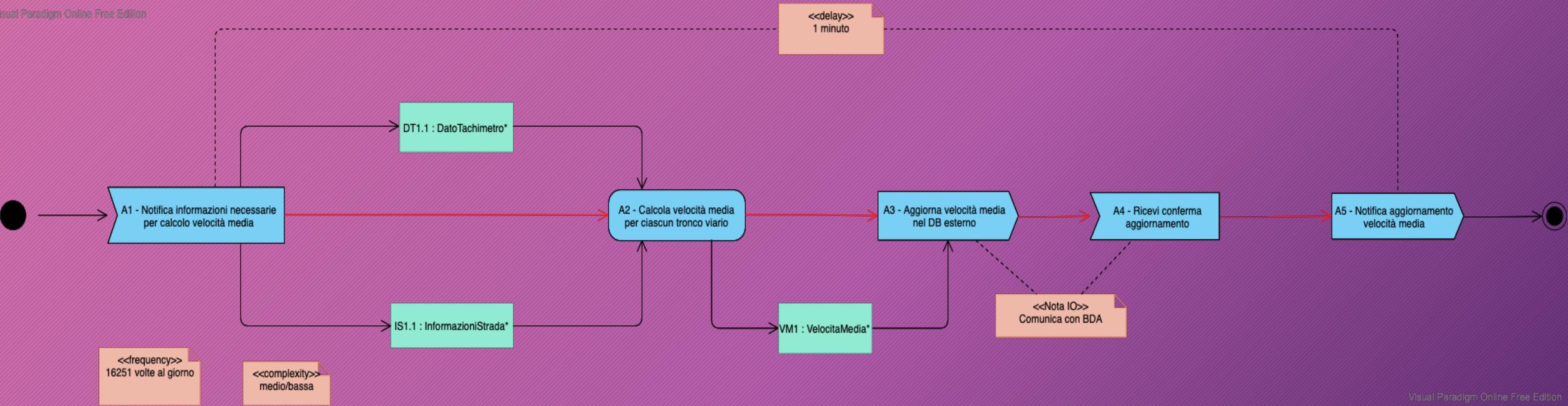
AD9 - Calcola livello di rischio

UC9- Calcola livello di rischio



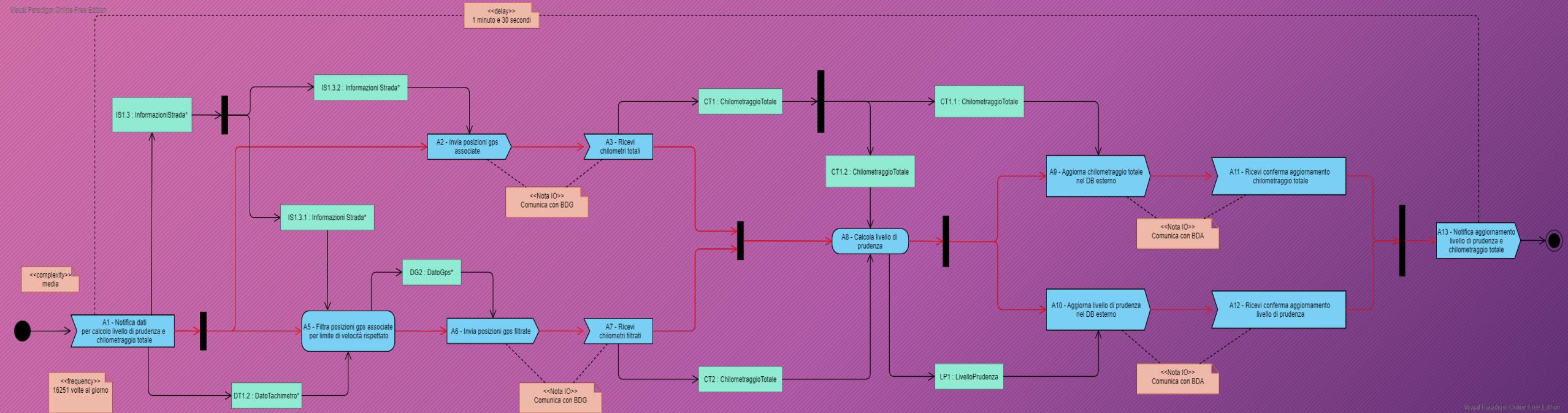
AD10 - Estrai dati tachimetro per calcolo parametri livello di rischio

UC9- Calcola livello di rischio



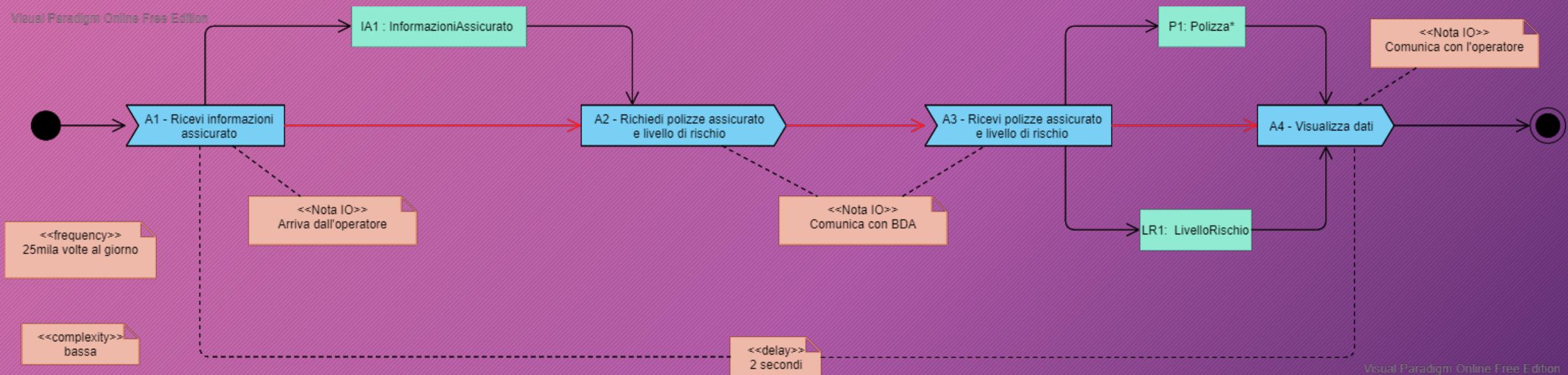
AD11 - Calcola velocità media per ciascun tronco viario T

UC10- Calcola velocità media per ciascun tronco viario



AD12 - Calcola livello di prudenza P e chilometraggio totale K

UC11- Calcola livello di prudenza P e chilometraggio totale K



AD13 - Visualizza condizioni di polizza e livello di rischio

UC12- Visualizza condizioni di polizza e livello di rischio

Architettura Logica

Partizionamento per complexity

Partizionamento per abstraction dei task

Confronto tra i due partizionamenti

Partizionamento per complexity

Diagramma Attività	Delay	Frequency	Complexity	Componente	Componente	Molteplicità
DA1 - Acquisisci accelerazione	0.1 secondi	10 acquisizioni al secondo 864000 volte al giorno	bassa	Complessità bassa	Complessità bassa	932
DA2 - Acquisisci posizione GPS	5 secondi	1 acquisizione ogni 5 secondi 17280 volte al giorno	bassa	Complessità bassa	Complessità media	1
DA3 - Acquisisci velocità	5 secondi	1 acquisizione ogni 5 secondi 17280 volte al giorno	bassa	Complessità bassa	Complessità alta	1
DA6 - Attiva connessione telefonica con assicurato coinvolto	0.5 secondi	88 incidenti al giorno	bassa	Complessità bassa		
DA7 - Controlla polizze in scadenza	30 secondi	1 volta al giorno	bassa	Complessità bassa		
DA5 - Notifica informazioni necessarie per l'assistenza	1 secondo	88 incidenti al giorno	bassa	Complessità bassa		
DA4 - Rileva collisione	1 secondo	1 volta al secondo 86400 volte al giorno	media/bassa	Complessità bassa		
DA13 - Visualizza condizioni di polizza e livello di rischio	2 secondi	25000 volte al giorno	bassa	Complessità bassa		
DA10 - Estra dati tachimetro per calcolo parametri livello di rischio	1 minuto e 35 secondi	16251 volte al giorno	bassa	Complessità bassa		
DA12 - Calcola livello di prudenza P e chilometraggio totale K	1 minuto e 30 secondi	16251 volte al giorno	media	Complessità media		
DA11 - Calcola velocità media per ciascun tronco viario T	1 minuto	16251 volte al giorno	media/bassa	Complessità media		
DA9 - Calcola livello di rischio	3 minuti	16251 volte al giorno	alta	Complessità alta		
DA8 - Ridefinisci premio di assicurazione	5 minuti	16251 volte al giorno	alta	Complessità alta		

I buffer sono allocati all'interno della componente a complessità bassa, mentre i datastore di GPS e velocità sono esterni alle componenti.

Stima delle dimensioni statiche

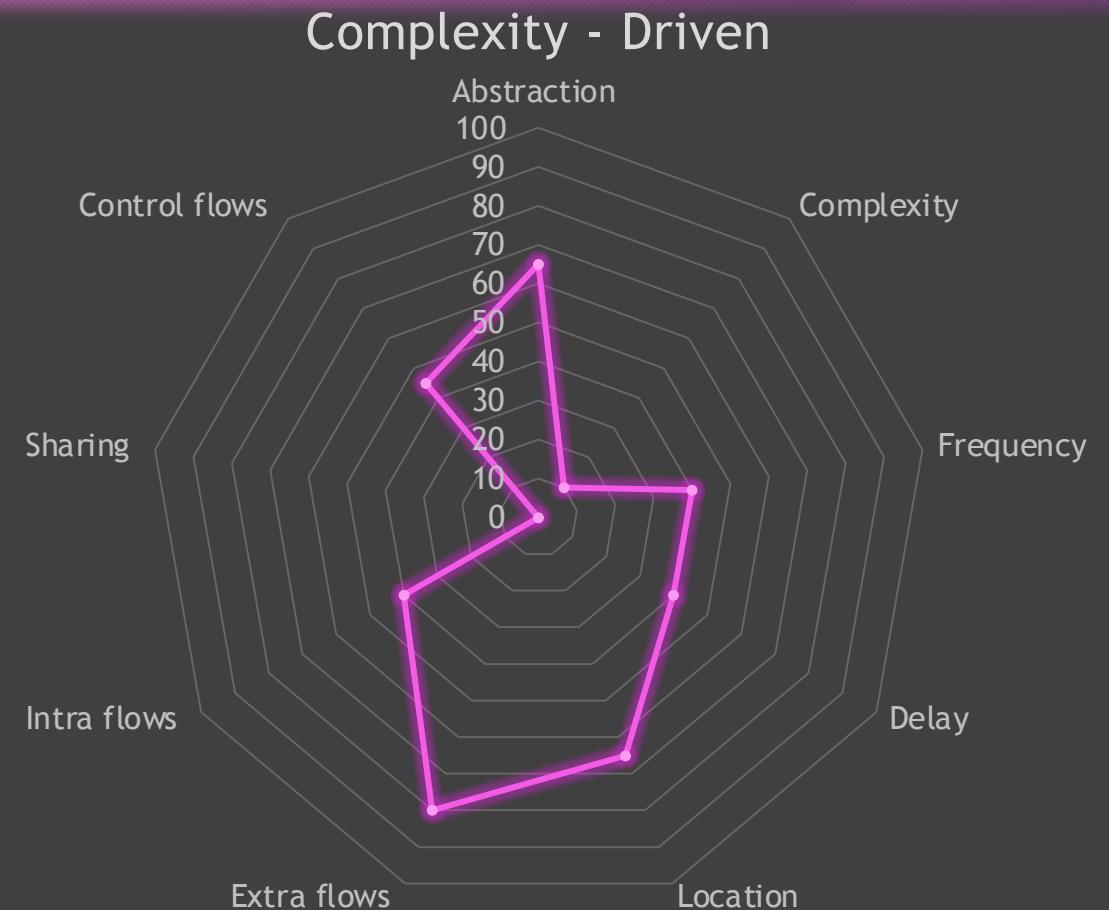
	Dimension	Values	Commenti
Statiche Spread	Abstraction	65	La componente a bassa complessità incide in modo significativo sull'innalzamento dello spread in quanto i task contenuti risultano fortemente eterogenei al punto che andrebbero separati in componenti diverse.
	Complexity	10	Le componenti logiche radunano quasi alla perfezione le attività con complessità omogenee.
	Frequency	40	La componente a bassa complessità racchiude molteplici attività con frequenze eterogenee, aggravando così sullo spread. Le componenti ad alta e media complessità, d'altro canto, hanno al loro interno attività con frequenze fortemente omogenee.
	Delay	40	La componente a bassa complessità incide in modo significativo sull'innalzamento dello spread a causa dei delay non propriamente omogenei.
	Location	65	La componente a bassa complessità incide in modo significativo sull'innalzamento dello spread in quanto coinvolge molteplici attori e di conseguenza non sarà coresidente con molti di essi.

Stima delle dimensioni dinamiche

	Dimension	Values	Commenti
Dinamiche Interferenza	Extra flows	80	Il valore dell'interferenza è alto perché ogni istanza della componente a bassa complessità comunica con un alto numero di attori (i sensori installati sulle automobili). Incidono, seppur in modo inferiore rispetto alla componente citata prima, anche le istanze delle altre due componenti a causa delle comunicazioni con BDA e BDG per gestire le attività di rinnovo polizza.
	Intra flows	40	Il valore dell'interferenza è medio in quanto vi è una comunicazione diretta tra le istanze delle varie componenti nel momento in cui viene avviata l'operazione di rinnovo di una polizza, ma allo stesso tempo viene mitigato dal numero di istanze della componente a bassa complessità.
	Sharing	0	Il valore dell'interferenza è zero, perché il sistema è progettato in modo da utilizzare i dati di GPS e velocità salvati nei datastore entro l'istante di inizio del controllo sulle polizze in scadenza, quindi sicuramente in fase di rinnovo delle polizze non vengono letti i dati che in quel momento le componenti a bassa complessità stanno caricando sui datastore.
	Control flows	45	Il valore dell'interferenza è medio perché ci sono diversi flussi di controllo tra le varie istanze delle componenti, questo a causa delle attività di rinnovo polizza.

Partizionamento per complexity

	Dimension	Values
Statiche	Abstraction	65
	Complexity	10
	Frequency	40
	Delay	40
	Location	65
Dinamiche	Extra flows	80
	Intra flows	40
	Sharing	0
	Control flows	45



Partizionamento per abstraction dei task

Diagramma Attività	Delay	Frequency	Complexity	Componente	Componente	Molteplicità
DA6 - Attiva connessione telefonica con assicurato coinvolto	0.5 secondi	88 incidenti al giorno	bassa	Gestore assistenza	Scatola nera	5,931,785
DA5 - Notifica informazioni necessarie per l'assistenza	1 secondo	88 incidenti al giorno	bassa	Gestore assistenza	Sistema di rinnovo polizze	1
DA13 - Visualizza condizioni di polizza e livello di rischio	2 secondi	25000 volte al giorno	bassa	Gestore polizze	Gestore assistenza	932
DA1 - Acquisisci accelerazione	0.1 secondi	10 acquisizioni al secondo 864000 volte al giorno	bassa	Scatola nera	Gestore polizze	932
DA2 - Acquisisci posizione GPS	5 secondi	1 acquisizione ogni 5 secondi 17280 volte al giorno	bassa	Scatola nera		
DA3 - Acquisisci velocità	5 secondi	1 acquisizione ogni 5 secondi 17280 volte al giorno	bassa	Scatola nera		
DA4 - Rileva collisione	1 secondo	1 volta al secondo 86400 volte al giorno	media/bassa	Scatola nera		
DA12 - Calcola livello di prudenza P e chilometraggio totale K	1 minuto e 30 secondi	16251 volte al giorno	media	Sistema di rinnovo polizze		
DA9 - Calcola livello di rischio	3 minuti	16251 volte al giorno	alta	Sistema di rinnovo polizze		
DA11 - Calcola velocità media per ciascun tronco viario T	1 minuto	16251 volte al giorno	media/bassa	Sistema di rinnovo polizze		
DA7 - Controlla polizze in scadenza	30 secondi	1 volta al giorno	bassa	Sistema di rinnovo polizze		
DA8 - Ridefinisci premio di assicurazione	5 minuti	16251 volte al giorno	alta	Sistema di rinnovo polizze		
DA10 - Estrai dati tachimetro per calcolo parametri livello di rischio	1 minuto e 35 secondi	16251 volte al giorno	bassa	Sistema di rinnovo polizze		

I buffer sono allocati all'interno della scatola nera, mentre i datastore di GPS e velocità sono esterni alle componenti.

Stima delle dimensioni statiche

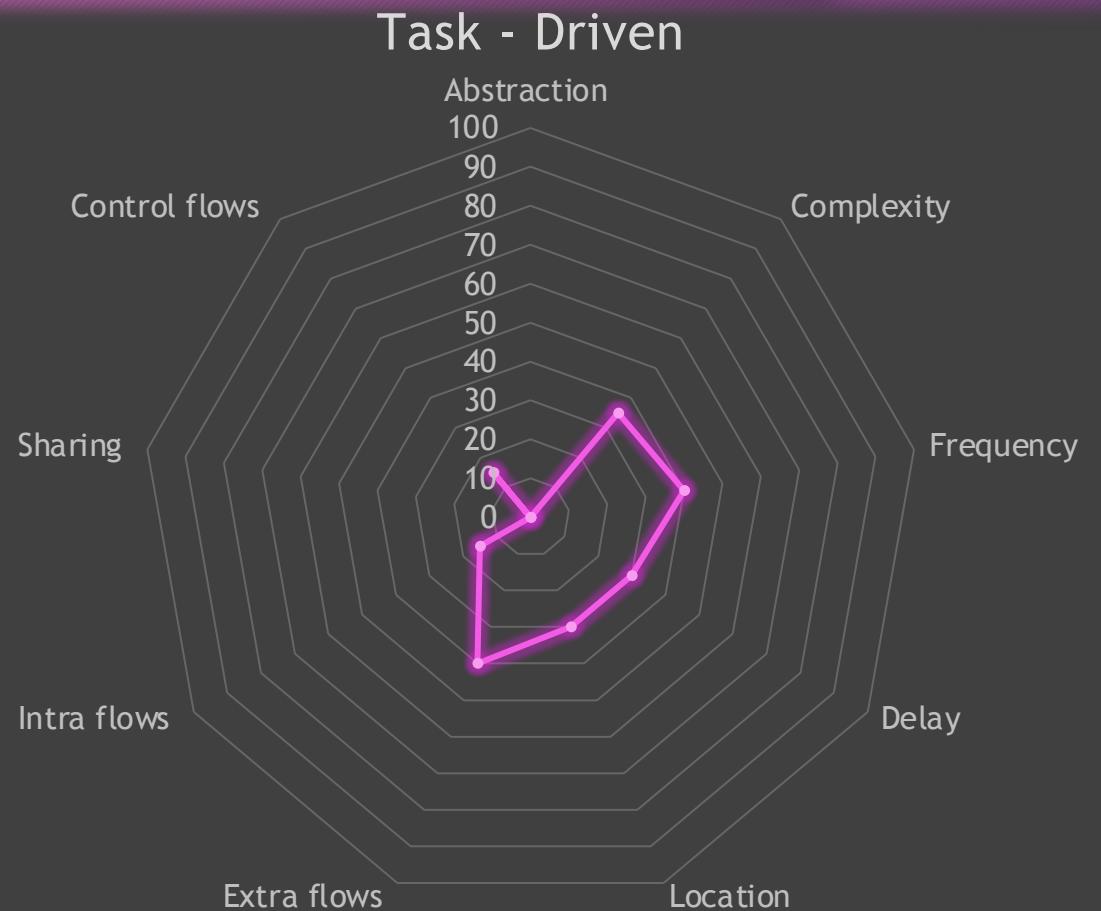
Dimension	Values	Commenti
Statiche Spread	Abstraction	Rispetto al partizionamento per complexity, lo spread è nullo perché questa è la dimensione su cui è stato partizionato il sistema in questa nuova soluzione.
	Complexity	Rispetto al partizionamento per complexity, lo spread è aumentato a causa del partizionamento secondo una diversa dimensione, ma è ancora ragionevolmente basso.
	Frequency	Rispetto al partizionamento per complexity, lo spread è rimasto invariato perché la componente logica che racchiudeva frequenze diverse è stata suddivisa in più componenti, alcune delle quali però sono ancora disomogenee.
	Delay	Rispetto al partizionamento per complexity, i delay per ogni componente sono leggermente più omogenei.
	Location	Rispetto al partizionamento per complexity, lo spread è migliorato perché, ripartizionando per task, è stato possibile anche migliorare la coresidenza delle componenti logiche con gli attori con cui esse comunicano.

Stima delle dimensioni dinamiche

	Dimension	Values	Commenti
Dinamiche Interferenza	Extra flows	40	Rispetto al partizionamento per complexity, l'interferenza è migliorata perché è aumentato il numero di istanze che comunica con i sensori, che sono tra gli attori che comunicano con maggior probabilità in un dato istante con il sistema.
	Intra flows	15	Rispetto al partizionamento per complexity, l'interferenza è migliorata perché avviene una comunicazione tra le istanze dei componenti esclusivamente quando si verifica una collisione.
	Sharing	0	Rispetto al partizionamento per complexity, l'interferenza rimane sempre nulla per il motivo già spiegato nel precedente partizionamento.
	Control flows	15	Rispetto al partizionamento per complexity, l'interferenza è migliorata perché vi è un flusso di controllo solo quando viene notificata una collisione al gestore assistenza, istanziato tante volte quanto il numero di sedi.

Partizionamento per abstraction dei task

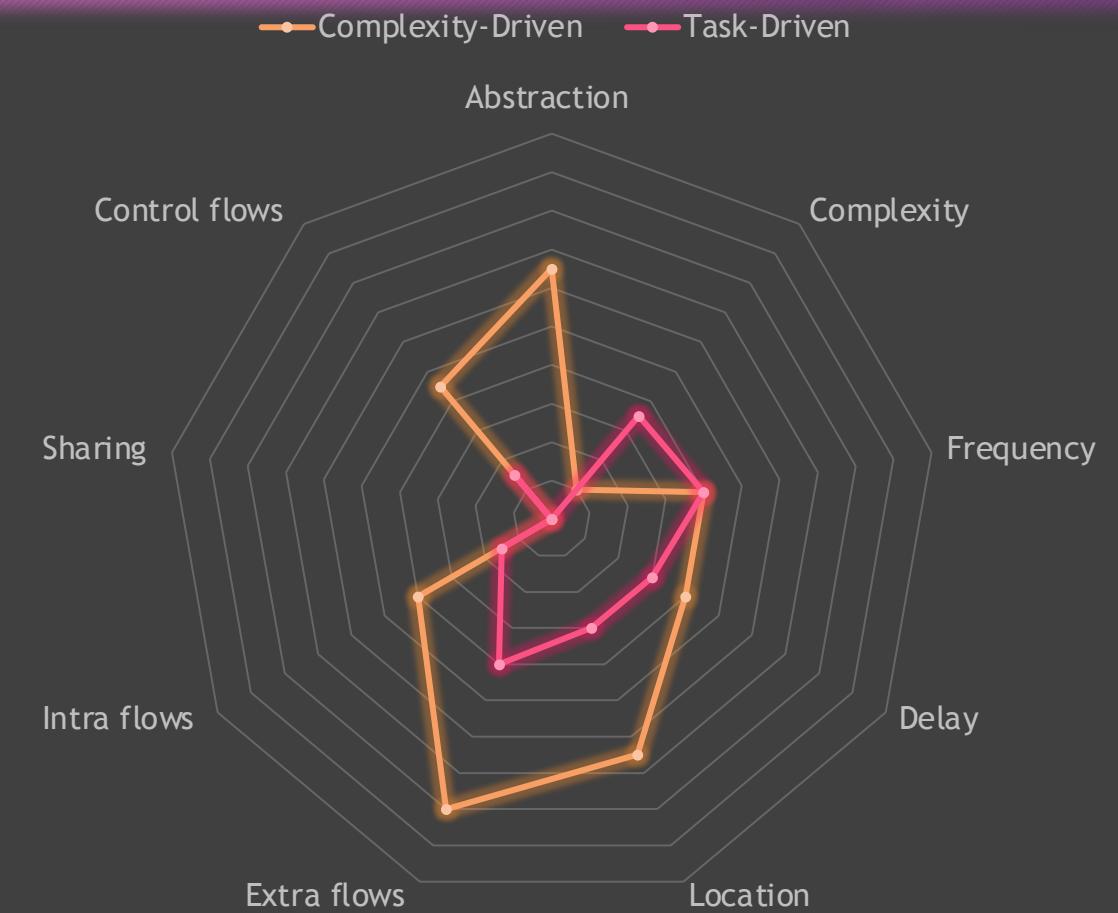
	Dimension	Values
Statiche	Abstraction	0
	Complexity	35
	Frequency	40
	Delay	30
	Location	30
Dinamiche	Extra flows	40
	Intra flows	15
	Sharing	0
	Control flows	15



Confronto tra i due partizionamenti

Come si evince dal grafico,
la soluzione Task-Driven risulta migliorare gran parte
delle dimensioni, ma come previsto peggiora la
Complexity.

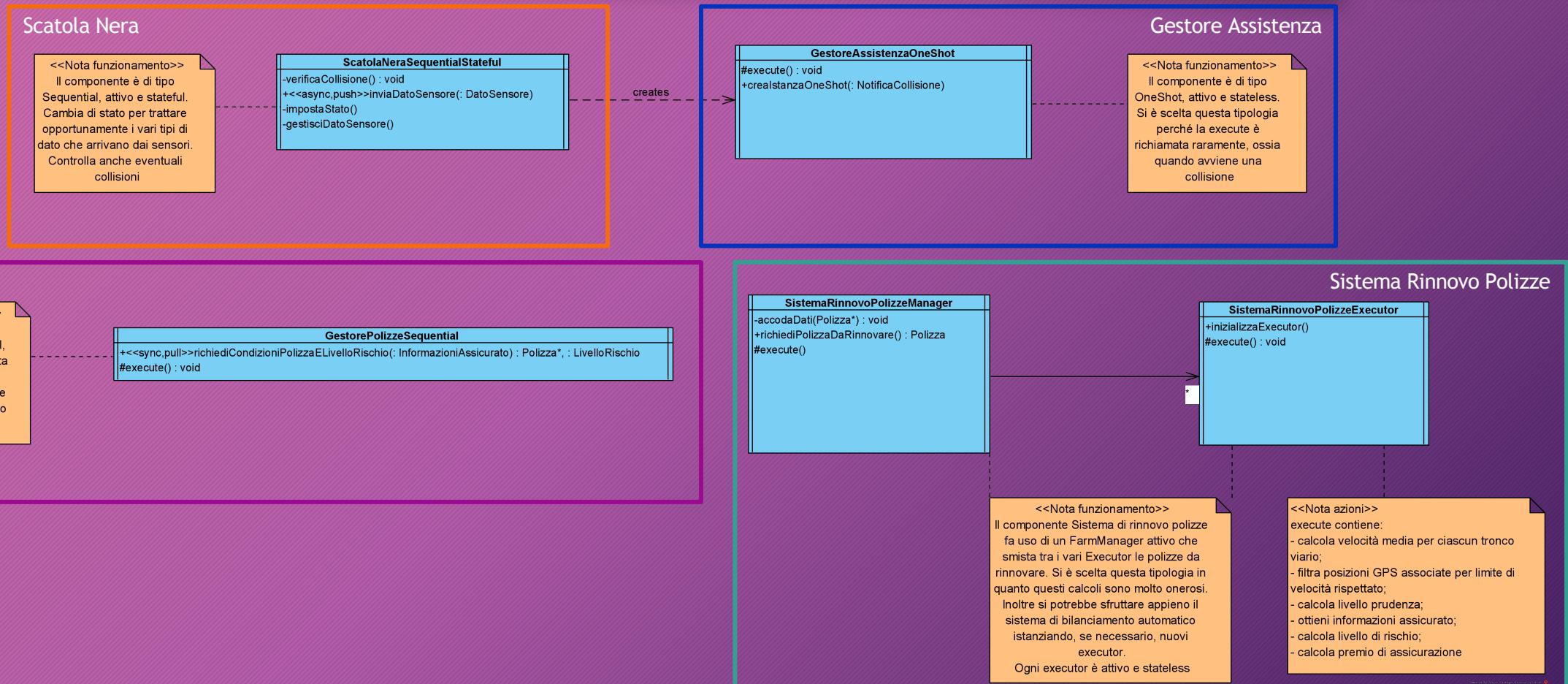
Per questo motivo si è deciso di adottare la seconda
architettura logica presentata.



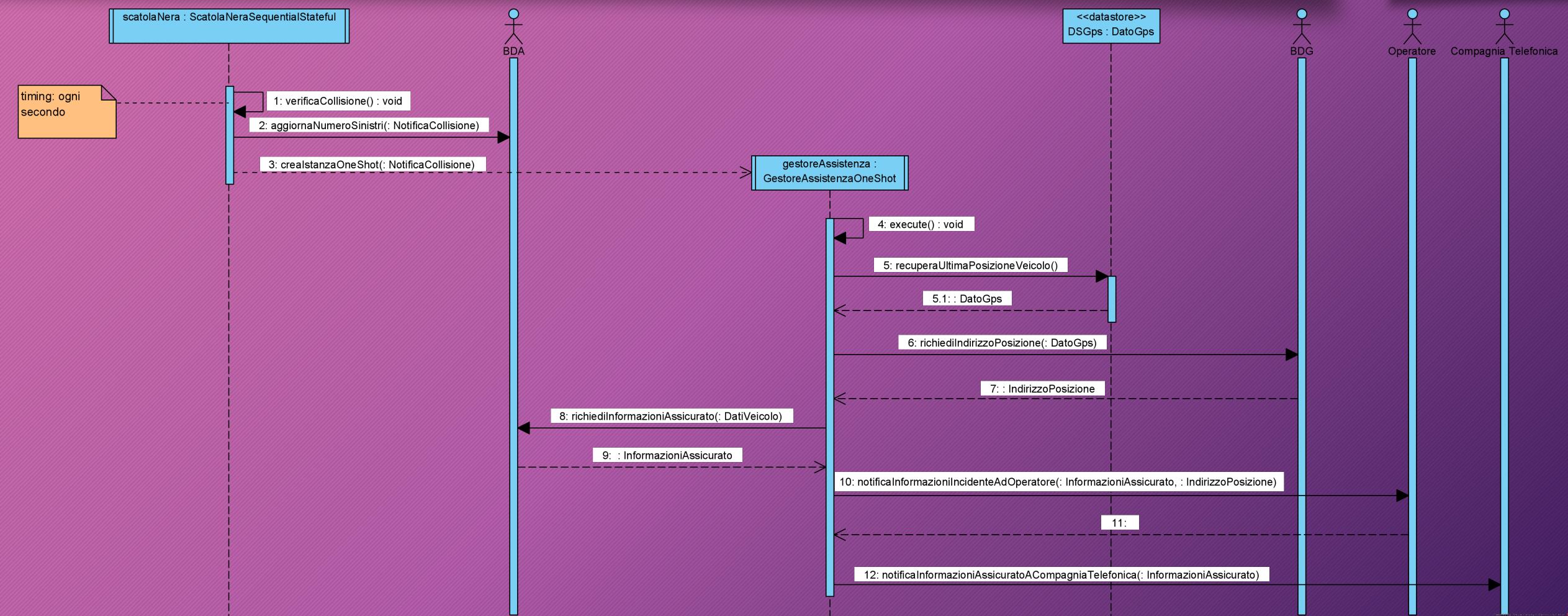
Architettura Concreta

Disegno interno dei componenti
Rilevamento e notifica collisione
Controllo polizze in scadenza e ridefinizione premio

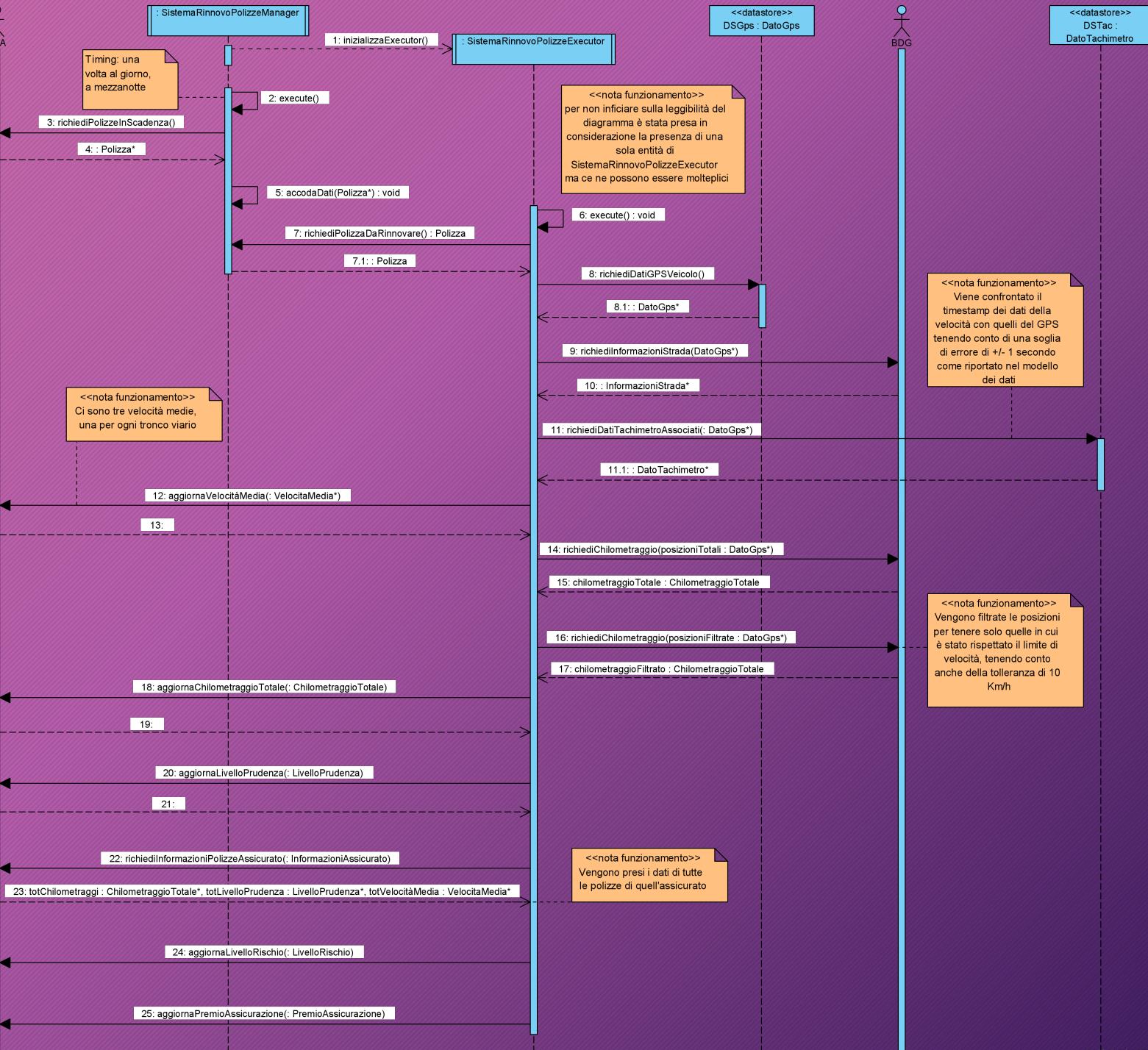
Disegno interno dei componenti



Rilevamento e notifica collisione



Controllo polizze in scadenza e ridefinizione premio di assicurazione

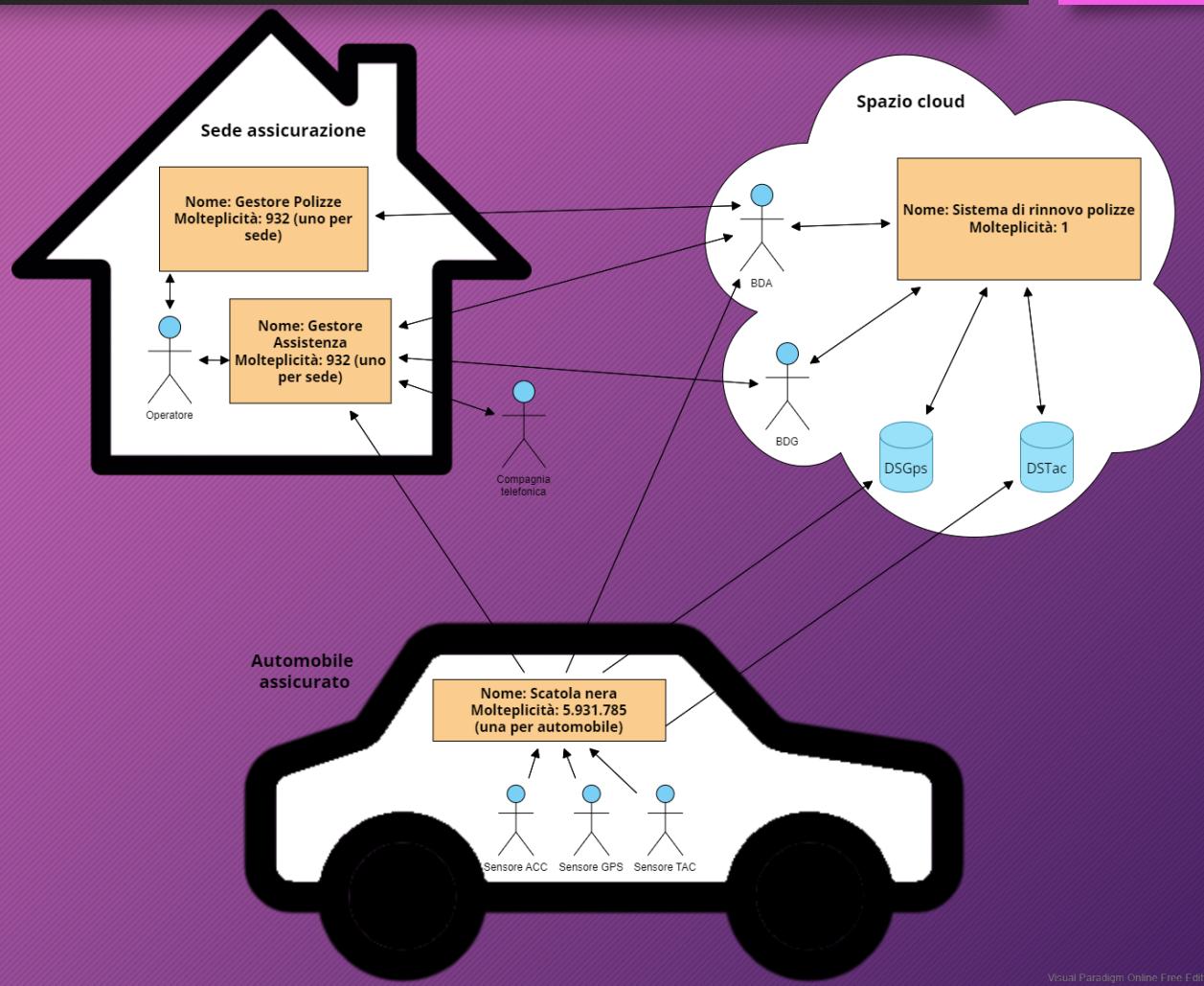


Architettura di Deployment

Scelte dell'architettura di deployment
Perché il cloud?

Scelte dell'architettura di deployment

1. **Scatola nera:** reificata in un sistema real-time;
2. **Gestore assistenza:** reificato in una infrastruttura interna alla sede;
3. **Gestore polizze:** reificato in una infrastruttura interna alla sede;
4. **Sistema di rinnovo polizze e DataStore:** reificati in un sistema cloud pubblico, fornito da AWS e delegando una parte della sicurezza dei dati, e dell'applicazione stessa, al provider.



Perché il cloud?

La scelta di gestire il DSGps e il DSTac in uno spazio cloud è motivata dalla proprietà di **ubiquitous network access** cioè l'avere delle risorse sempre disponibili.

Anche il sistema di rinnovo polizze si trova nello spazio cloud, per i seguenti motivi:

- Risulta vantaggioso economicamente per una questione di economia di scala;
- Risulta performante grazie alla proprietà di **rapid elasticity**: cioè l'allocazione di nuove risorse, se necessaria, avviene rapidamente;
- L'infrastruttura cloud può automaticamente riavviare applicazioni andate in crash.

Perché il cloud?

Tuttavia, nel momento in cui si effettua il design dell'architettura di deployment, bisogna tenere conto di alcune potenziali criticità:

- **Cloud privato:** L'utilizzo di un cloud privato provocherebbe un aumento non indifferente dei costi di mantenimento;
- **Sviluppo ad-hoc dell'applicazione:** al fine di rendere l'applicazione modificabile e adattabile nel tempo, risulta particolarmente importante progettare quest'ultima ad-hoc, rendendo però necessario un lavoro più accurato da parte dello sviluppatore;
- **Vulnerabilità informatica:** essendo uno spazio cloud, risulta poter essere preda di attacchi informatici, anche qual ora questi avvengano su altre applicazioni coresidenti alla nostra;
- **Sicurezza dei dati:** in uno spazio cloud condiviso, potrebbe avvenire una perdita di dati che provocherebbe gravi ripercussioni sui clienti e sulla società stessa.

Eventuali modifiche future dell'architettura

Nel caso in cui, in fase di sviluppo, si dovesse notare un eccessiva complessità algoritmica per determinare l'associazione tra posizione GPS e velocità istantanea utilizzando i loro timestamp, si potrebbe modificare l'architettura del sistema associando i due dati in fase di acquisizione e salvandoli in un unico DataStore.

Grazie per l'attenzione.

