

Progetto di Ingegneria del Software II

Philippe Scorsolini,
Lorenzo Semeria,
Gabriele Vanoni

Politecnico di Milano

6 Marzo 2017

Indice

- 1 Introduzione
- 2 Analisi dei requisiti e specifica
 - Requisiti Funzionali
 - Requisiti non funzionali
 - Modellazione in Alloy
- 3 Design
- 4 Test di integrazione

PowerEnJoy: caratteristiche principali

PowerEnJoy: caratteristiche principali

- Servizio di Car Sharing di auto elettriche

PowerEnJoy: caratteristiche principali

- Servizio di Car Sharing di auto elettriche
- Prenotazione tramite smartphone o sito web

PowerEnJoy: caratteristiche principali

- Servizio di Car Sharing di auto elettriche
- Prenotazione tramite smartphone o sito web
- Possibilità di iniziare e terminare il noleggio solo all'interno di zone predefinite

PowerEnJoy: caratteristiche principali

- Servizio di Car Sharing di auto elettriche
- Prenotazione tramite smartphone o sito web
- Possibilità di iniziare e terminare il noleggio solo all'interno di zone predefinite
- Possibilità di ottenere sconti in base ad alcuni parametri (numero di passeggeri, termine noleggio in una stazione di ricarica)

Analisi dei requisiti

I Goal

Identificati **11 Goal** e un totale di **60 Requisiti**

I Goal possono essere suddivisi in tre macrogruppi:

- **Gestione utenti**
- **Gestione auto**
- **Utilizzo del Servizio**

Gestione utenti

Le caratteristiche e i requisiti della gestione utenti vengono definiti dai primi 3 goal:

Gestione utenti

Le caratteristiche e i requisiti della gestione utenti vengono definiti dai primi 3 goal:

- G1: **Registrazione** Utenti
 - Sono necessarie email e patente valide. Il metodo di pagamento viene aggiunto successivamente

Gestione utenti

Le caratteristiche e i requisiti della gestione utenti vengono definiti dai primi 3 goal:

- G1: **Registrazione** Utenti
 - Sono necessarie email e patente valide. Il metodo di pagamento viene aggiunto successivamente
- G2: **Login** utente
 - I requisiti definiscono le caratteristiche principali di questo goal, ad esempio la necessità di inserire la combinazione corretta di username e password

Gestione utenti

Le caratteristiche e i requisiti della gestione utenti vengono definiti dai primi 3 goal:

- G1: **Registrazione** Utenti
 - Sono necessarie email e patente valide. Il metodo di pagamento viene aggiunto successivamente
- G2: **Login** utente
 - I requisiti definiscono le caratteristiche principali di questo goal, ad esempio la necessità di inserire la combinazione corretta di username e password
- G3: **Modifica** del profilo
 - È possibile effettuare ogni modifica a condizione che i nuovi dati inseriti siano validi

Gestione auto

La gestione delle auto è fortemente collegata con l'utilizzo del servizio, tra i goal principali possiamo individuare i seguenti.

Gestione auto

La gestione delle auto è fortemente collegata con l'utilizzo del servizio, tra i goal principali possiamo individuare i seguenti.

- G4: Mostrare le **informazioni** aggiornate sulle auto
 - Le auto mantengono il sistema aggiornato sulla propria posizione e carica.

Gestione auto

La gestione delle auto è fortemente collegata con l'utilizzo del servizio, tra i goal principali possiamo individuare i seguenti.

- G4: Mostrare le **informazioni** aggiornate sulle auto
 - Le auto mantengono il sistema aggiornato sulla propria posizione e carica.
- G6: Permettere lo **sblocco** dell'auto
 - L'auto viene sbloccata dopo che il sistema conferma che l'utente è nei pressi dell'auto

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata
 - Azione riservata agli utenti abilitati.

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata
 - Azione riservata agli utenti abilitati.
 - Comportamento nel caso in cui l'auto non venga sbloccata o messa in moto in tempo.

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata
 - Azione riservata agli utenti abilitati.
 - Comportamento nel caso in cui l'auto non venga sbloccata o messa in moto in tempo.
- G7: Calcolo della **tariffa**

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata
 - Azione riservata agli utenti abilitati.
 - Comportamento nel caso in cui l'auto non venga sbloccata o messa in moto in tempo.
- G7: Calcolo della **tariffa**
 - La tariffa viene calcolata tenendo conto di tutti i modificatori (sconti o sovrapprezzi) e del tempo di utilizzo.

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata
 - Azione riservata agli utenti abilitati.
 - Comportamento nel caso in cui l'auto non venga sbloccata o messa in moto in tempo.
- G7: Calcolo della **tariffa**
 - La tariffa viene calcolata tenendo conto di tutti i modificatori (sconti o sovrapprezzi) e del tempo di utilizzo.
 - Per alcuni casi particolari è stato previsto un comportamento ad hoc

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata
 - Azione riservata agli utenti abilitati.
 - Comportamento nel caso in cui l'auto non venga sbloccata o messa in moto in tempo.
- G7: Calcolo della **tariffa**
 - La tariffa viene calcolata tenendo conto di tutti i modificatori (sconti o sovrapprezzi) e del tempo di utilizzo.
 - Per alcuni casi particolari è stato previsto un comportamento ad hoc
- G9: **Pagamento** della tariffa

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata
 - Azione riservata agli utenti abilitati.
 - Comportamento nel caso in cui l'auto non venga sbloccata o messa in moto in tempo.
- G7: Calcolo della **tariffa**
 - La tariffa viene calcolata tenendo conto di tutti i modificatori (sconti o sovrapprezzi) e del tempo di utilizzo.
 - Per alcuni casi particolari è stato previsto un comportamento ad hoc
- G9: **Pagamento** della tariffa
 - Il pagamento viene effettuato immediatamente al termine del noleggio

Utilizzo del Servizio

Le modalità e i requisiti relativi all'utilizzo del servizio costituiscono la maggior parte dei goal che abbiamo identificato. Tra i principali:

- G5/G6: **Prenotazione** e **sblocco** dell'auto prenotata
 - Azione riservata agli utenti abilitati.
 - Comportamento nel caso in cui l'auto non venga sbloccata o messa in moto in tempo.
- G7: Calcolo della **tariffa**
 - La tariffa viene calcolata tenendo conto di tutti i modificatori (sconti o sovrapprezzi) e del tempo di utilizzo.
 - Per alcuni casi particolari è stato previsto un comportamento ad hoc
- G9: **Pagamento** della tariffa
 - Il pagamento viene effettuato immediatamente al termine del noleggio
 - In caso di fallimento (metodo non più valido) l'utente viene notificato e il suo account temporaneamente disabilitato.

Requisiti non funzionali

- Le applicazioni per gli utenti e il sito devono essere **user friendly**

Requisiti non funzionali

- Le applicazioni per gli utenti e il sito devono essere **user friendly**
- Il sistema deve essere **disponibile sempre** (giorno e notte)

Requisiti non funzionali

- Le applicazioni per gli utenti e il sito devono essere **user friendly**
- Il sistema deve essere **disponibile sempre** (giorno e notte)
- Il sistema deve essere facilmente **scalabile** in caso di crescita

Requisiti non funzionali

- Le applicazioni per gli utenti e il sito devono essere **user friendly**
- Il sistema deve essere **disponibile sempre** (giorno e notte)
- Il sistema deve essere facilmente **scalabile** in caso di crescita
- I dati personali devono essere salvati in modo **sicuro** e nel rispetto delle norme

Requisiti non funzionali

- Le applicazioni per gli utenti e il sito devono essere **user friendly**
- Il sistema deve essere **disponibile sempre** (giorno e notte)
- Il sistema deve essere facilmente **scalabile** in caso di crescita
- I dati personali devono essere salvati in modo **sicuro** e nel rispetto delle norme
- È fondamentale che il sistema sia ben protetto da attacchi per prevenire, ad esempio, il furto di auto

Soluzioni di alcuni casi ambigui

In alcuni casi quanto scritto nella consegna non era del tutto esaustivo dei possibili scenari, per cui abbiamo qualora necessario deciso come gestire le singole eccezioni, ad esempio:

Soluzioni di alcuni casi ambigui

In alcuni casi quanto scritto nella consegna non era del tutto esaustivo dei possibili scenari, per cui abbiamo qualora necessario deciso come gestire le singole eccezioni, ad esempio:

- Il “Problema IKEA”:

Soluzioni di alcuni casi ambigui

In alcuni casi quanto scritto nella consegna non era del tutto esaustivo dei possibili scenari, per cui abbiamo qualora necessario deciso come gestire le singole eccezioni, ad esempio:

- Il “Problema IKEA”: abbiamo deciso di permettere agli utenti di “parcareggiare” nel senso comune del termine, ovvero continuando a pagare per l'auto, se si trovano fuori dalle zone in cui è possibile terminare il noleggio. (A12)

Soluzioni di alcuni casi ambigui

In alcuni casi quanto scritto nella consegna non era del tutto esaustivo dei possibili scenari, per cui abbiamo qualora necessario deciso come gestire le singole eccezioni, ad esempio:

- Il “Problema IKEA”: abbiamo deciso di permettere agli utenti di “parcareggiare” nel senso comune del termine, ovvero continuando a pagare per l'auto, se si trovano fuori dalle zone in cui è possibile terminare il noleggio. (A12)
- Apertura delle portiere senza accensione:

Soluzioni di alcuni casi ambigui

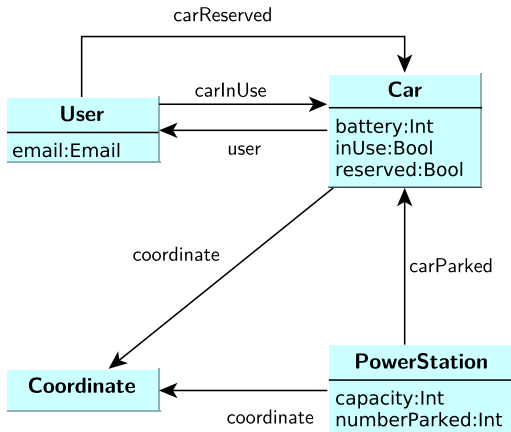
In alcuni casi quanto scritto nella consegna non era del tutto esaustivo dei possibili scenari, per cui abbiamo qualora necessario deciso come gestire le singole eccezioni, ad esempio:

- Il “Problema IKEA”: abbiamo deciso di permettere agli utenti di “parcheggiare” nel senso comune del termine, ovvero continuando a pagare per l'auto, se si trovano fuori dalle zone in cui è possibile terminare il noleggio. (A12)
- Apertura delle portiere senza accensione: dopo un breve periodo di tempo il sistema chiede all'utente di confermare di star bene, in caso contrario un Operatore si reca a verificare che non ci siano problemi (ad esempio un malessere dell'utente). Altrimenti, trascorso il tempo massimo di prenotazione, inizia a pagare la tariffa di utilizzo. (R30 e R31)

Modellazione in Alloy

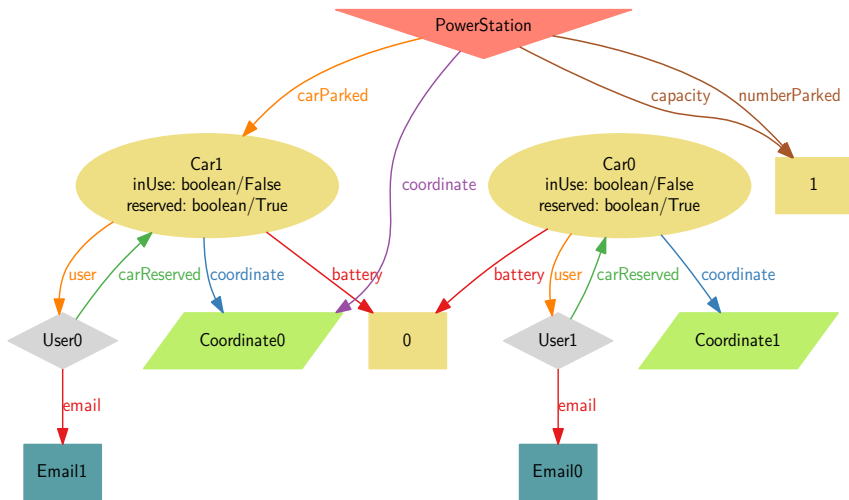
Il modello statico

Abbiamo scelto di modellare in **Alloy** questa parte del nostro sistema.



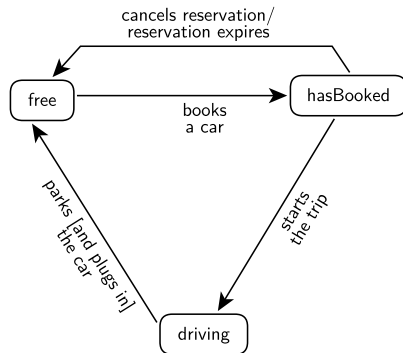
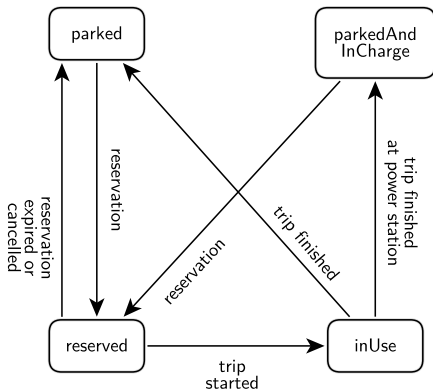
Uno dei mondi possibili

Questo è uno dei possibili output generati dal solver di **Alloy**.

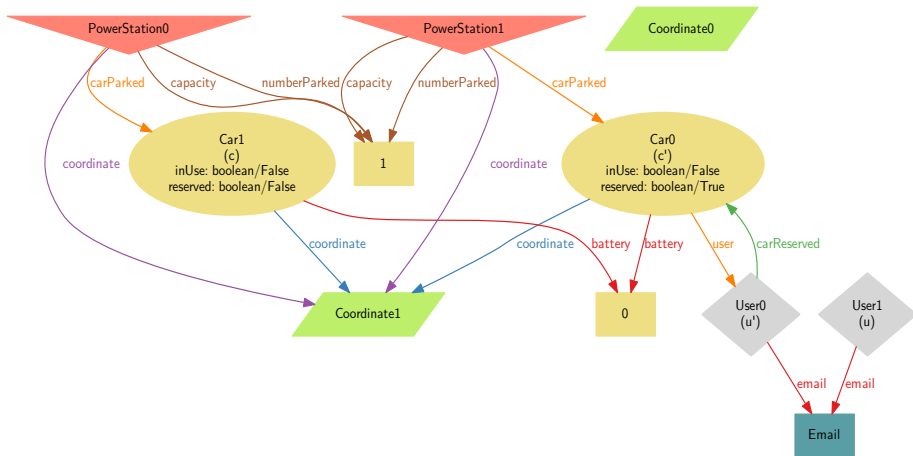


Il modello dinamico

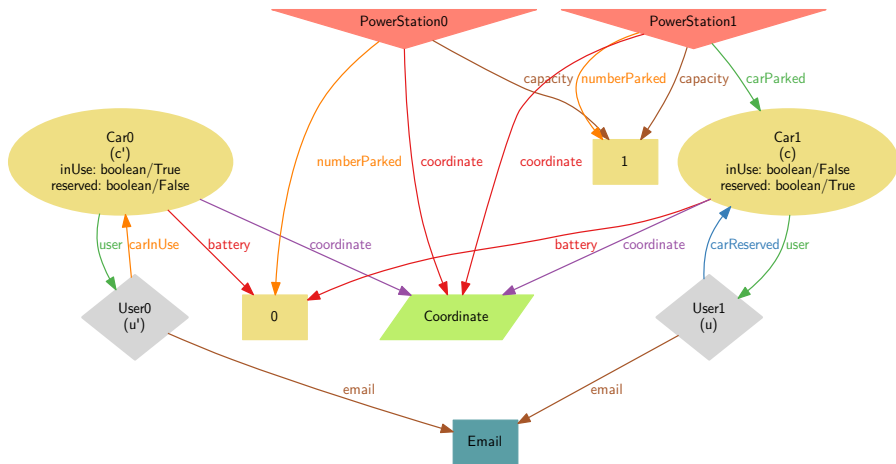
Queste **azioni** sono state modellate in Alloy come *predicati*.



Azione - prenotazione dell'auto



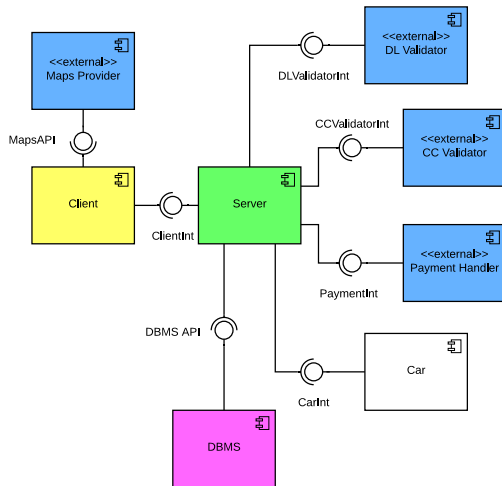
Azione - inizio del viaggio



Design

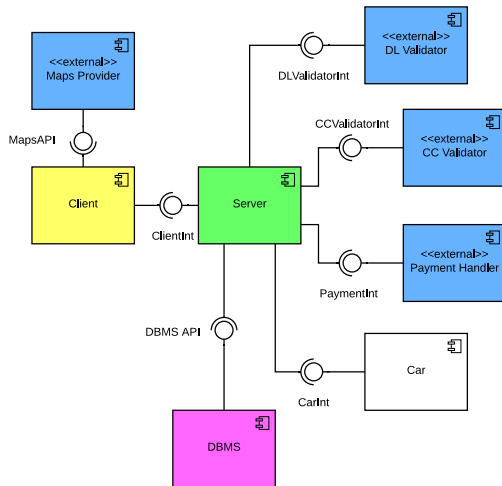
Design - Prime scelte

- Approccio misto
Top-down/Bottom-up



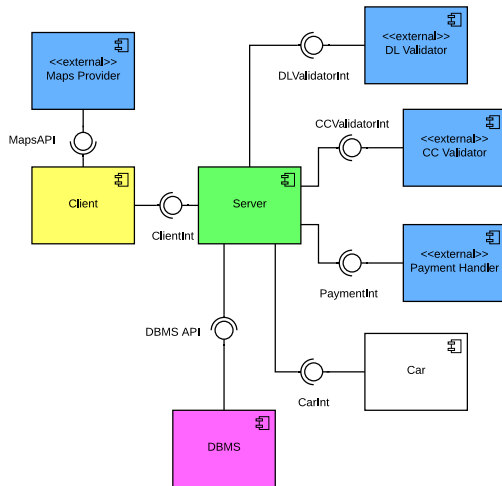
Design - Prime scelte

- Approccio misto
Top-down/Bottom-up
- Architettura 3 Tier
Client-Server con Thin Client:



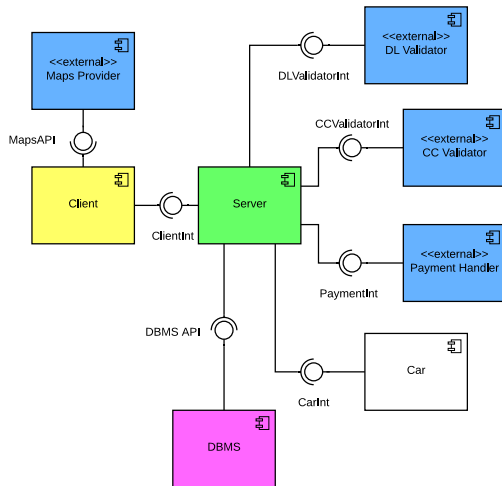
Design - Prime scelte

- Approccio misto
Top-down/Bottom-up
- Architettura 3 Tier
Client-Server con Thin Client:
 - possibilità di gestire diversi client



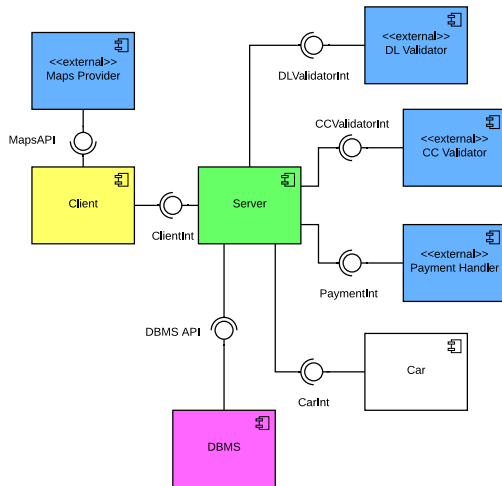
Design - Prime scelte

- Approccio misto
Top-down/Bottom-up
- Architettura 3 Tier
Client-Server con Thin Client:
 - possibilità di gestire diversi client
 - nessuna limitazione sull'hardware dei client



Design - Prime scelte

- Approccio misto
Top-down/Bottom-up
- Architettura 3 Tier
Client-Server con Thin Client:
 - possibilità di gestire diversi client
 - nessuna limitazione sull'hardware dei client
- Interazioni con componenti esterne



-
- ```

classDiagram
 class ClientHandler
 class RegistrationManager
 class AssistanceTicketManager
 class PowerStationManager
 class ReservationManager
 class CarManager
 class AdministrationModule
 class AuthenticationManager
 class ModelManager

 ClientHandler --> RegistrationManager
 ClientHandler --> AssistanceTicketManager
 ClientHandler --> PowerStationManager
 ClientHandler --> ReservationManager
 ClientHandler --> AuthenticationManager
 ClientHandler --> ModelManager

 RegistrationManager --> AssistanceTicketManager
 RegistrationManager --> PowerStationManager
 RegistrationManager --> ReservationManager
 RegistrationManager --> CarManager
 RegistrationManager --> ModelManager

 AssistanceTicketManager --> PowerStationManager
 AssistanceTicketManager --> ReservationManager
 AssistanceTicketManager --> CarManager
 AssistanceTicketManager --> ModelManager

 PowerStationManager --> ReservationManager
 PowerStationManager --> CarManager
 PowerStationManager --> ModelManager

 ReservationManager --> CarManager
 ReservationManager --> ModelManager

 CarManager --> ModelManager

 AdministrationModule --> AuthenticationManager
 AdministrationModule --> ModelManager

 AuthenticationManager --> ModelManager

 ModelManager --> ClientHandler
 ModelManager --> RegistrationManager
 ModelManager --> AssistanceTicketManager
 ModelManager --> PowerStationManager
 ModelManager --> ReservationManager
 ModelManager --> CarManager
 ModelManager --> AdministrationModule
 ModelManager --> AuthenticationManager

```
- The diagram illustrates the following components and their provided/required interfaces (G):
- Client Handler**: Provides G1, G6, G7, G8. Requires G1, G6, G7, G8.
  - Registration Manager**: Provides G1, G3. Requires G1, G3.
  - Assistance Ticket Manager**: Provides G1. Requires G1.
  - Power Station Manager**: Provides G10, G7. Requires G10, G7.
  - Reservation Manager**: Provides G4, G5, G6, G7, G9, G10, G11. Requires G4, G5, G6, G7, G9, G10, G11.
  - Car Manager**: Provides G4, G5, G6, G7, G10, G11. Requires G4, G5, G6, G7, G10, G11.
  - Administration Module**: Provides G1, G6, G7, G8. Requires G1, G6, G7, G8.
  - Authentication Manager**: Provides G2, G3, G5, G8. Requires G2, G3, G5, G8.
  - Model Manager**: Provides G8, G4. Requires G8, G4.
- Key relationships and dependencies include:
- Client Handler** depends on **Registration Manager**, **Assistance Ticket Manager**, **Power Station Manager**, **Reservation Manager**, **Authentication Manager**, and **Model Manager**.
  - Registration Manager** depends on **Assistance Ticket Manager**, **Power Station Manager**, **Reservation Manager**, **Car Manager**, and **Model Manager**.
  - Assistance Ticket Manager** depends on **Power Station Manager**, **Reservation Manager**, **Car Manager**, and **Model Manager**.
  - Power Station Manager** depends on **Reservation Manager** and **Car Manager**.
  - Reservation Manager** depends on **Car Manager** and **Model Manager**.
  - Car Manager** depends on **Model Manager**.
  - Administration Module** depends on **Authentication Manager** and **Model Manager**.
  - Authentication Manager** depends on **Model Manager**.
  - Model Manager** depends on **Client Handler**, **Registration Manager**, **Assistance Ticket Manager**, **Power Station Manager**, **Reservation Manager**, **Car Manager**, **Administration Module**, and **Authentication Manager**.

# Come e dove mettere i componenti?

# Come e dove mettere i componenti?

- Crescente utilizzo di provider PaaS

# Come e dove mettere i componenti?

- Crescente utilizzo di provider PaaS
- Architettura a Micro Servizi su Cloud

# Come e dove mettere i componenti?

- Crescente utilizzo di provider PaaS
- Architettura a Micro Servizi su Cloud
  - Scalabilità in caso di necessità

# Come e dove mettere i componenti?

- Crescente utilizzo di provider PaaS
- Architettura a Micro Servizi su Cloud
  - Scalabilità in caso di necessità
  - Indipendenza nello sviluppo

# Come e dove mettere i componenti?

- Crescente utilizzo di provider PaaS
- Architettura a Micro Servizi su Cloud
  - Scalabilità in caso di necessità
  - Indipendenza nello sviluppo
  - Indipendenza nel deployment

# Come e dove mettere i componenti?

- Crescente utilizzo di provider PaaS
- Architettura a Micro Servizi su Cloud
  - Scalabilità in caso di necessità
  - Indipendenza nello sviluppo
  - Indipendenza nel deployment
  - Indipendenza dall'hardware

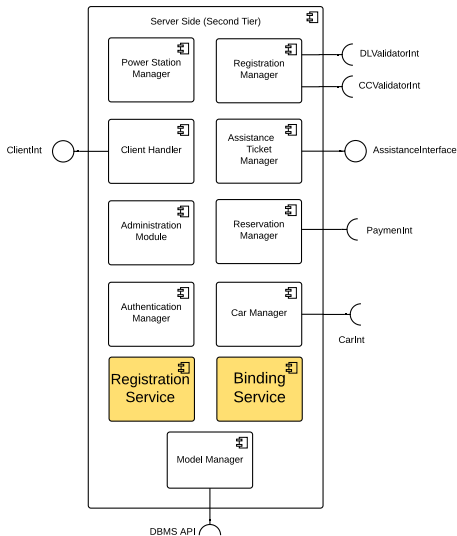


# Come e dove mettere i componenti?

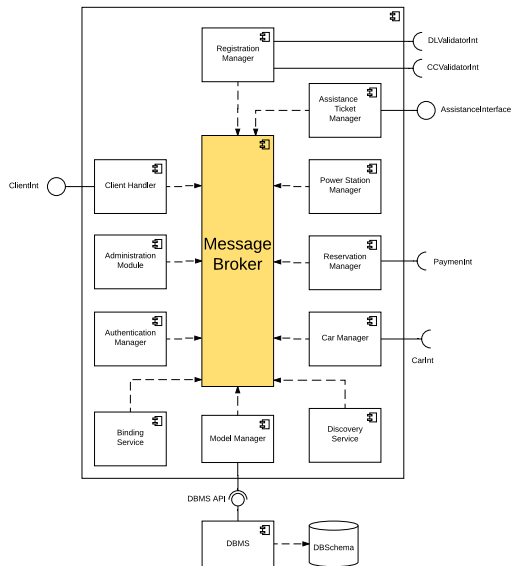
- Crescente utilizzo di provider PaaS
- Architettura a Micro Servizi su Cloud
  - Scalabilità in caso di necessità
  - Indipendenza nello sviluppo
  - Indipendenza nel deployment
  - Indipendenza dall'hardware
- Database centralizzato

# Come e dove mettere i componenti?

- Crescente utilizzo di provider PaaS
- Architettura a Micro Servizi su Cloud
  - Scalabilità in caso di necessità
  - Indipendenza nello sviluppo
  - Indipendenza nel deployment
  - Indipendenza dall'hardware
- Database centralizzato
- Necessari alcuni componenti per sfruttare appieno questa scelta architetturale

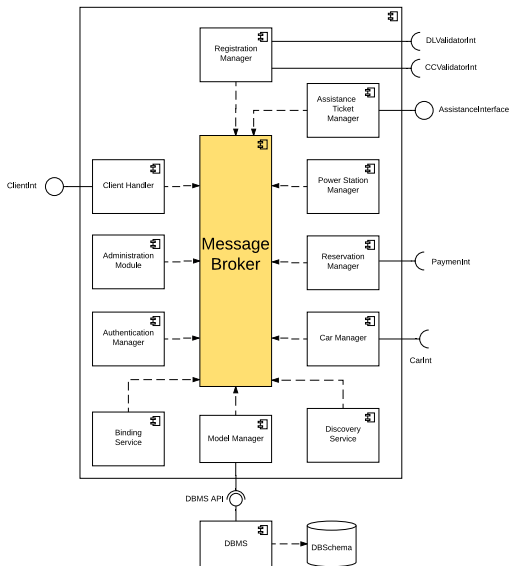


# La comunicazione



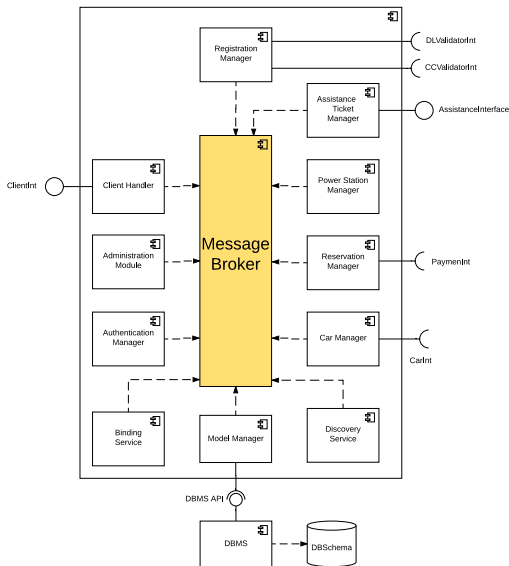
# La comunicazione

- Broker di messaggi centralizzato a code Pub/Sub:



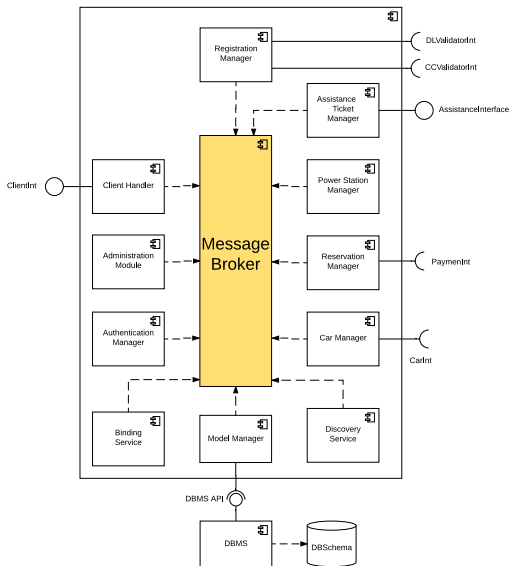
# La comunicazione

- Broker di messaggi centralizzato a code Pub/Sub:
  - Possibile singolo punto di rottura, ma casi reali dimostrano la solidità dell'approccio



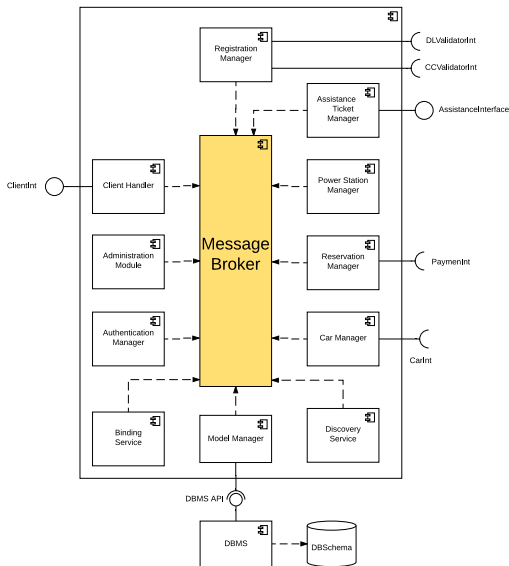
# La comunicazione

- Broker di messaggi centralizzato a code Pub/Sub:
  - Possibile singolo punto di rottura, ma casi reali dimostrano la solidità dell'approccio
  - Comunicazione asincrona



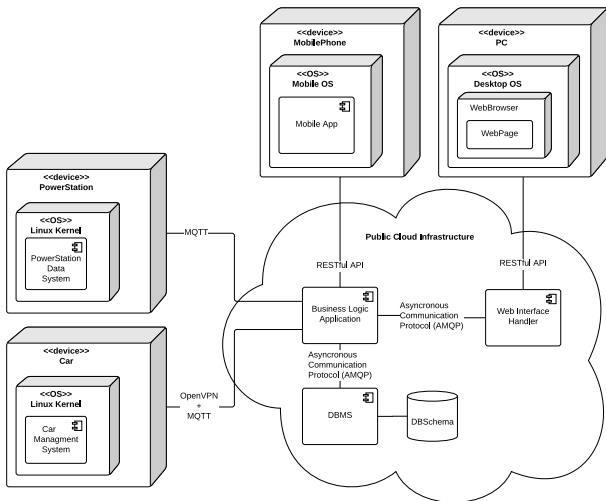
# La comunicazione

- Broker di messaggi centralizzato a code Pub/Sub:
  - Possibile singolo punto di rottura, ma casi reali dimostrano la solidità dell'approccio
  - Comunicazione asincrona
  - Possibilità di sfruttarlo per fare load balancing tra le diverse istanze dei componenti, anche "per versione"



# II Deployment

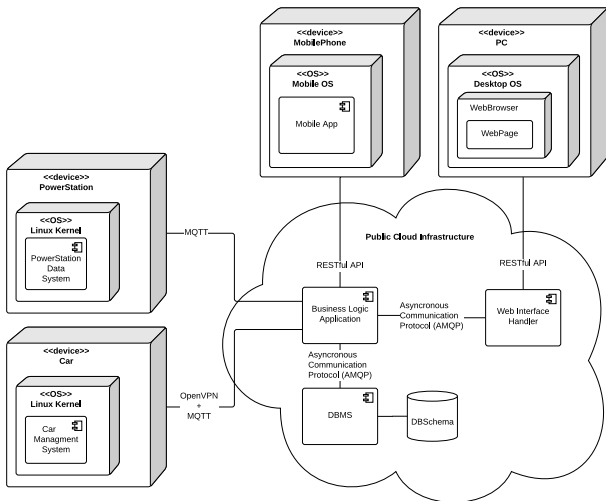
- PowerStation: sistema di acquisizione dati





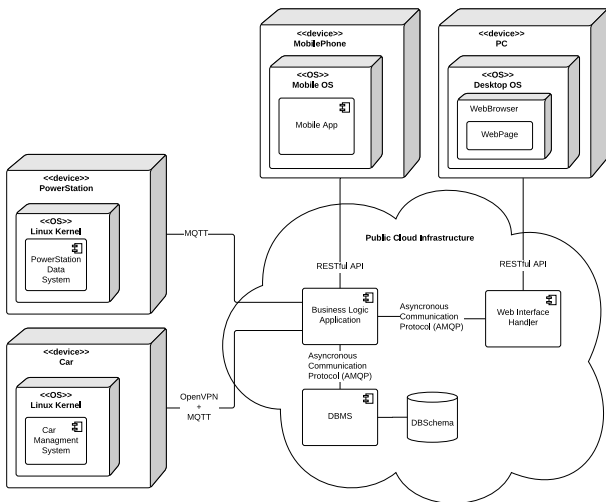
# II Deployment

- PowerStation: sistema di acquisizione dati
- Automobile: sistema di gestione dell'auto



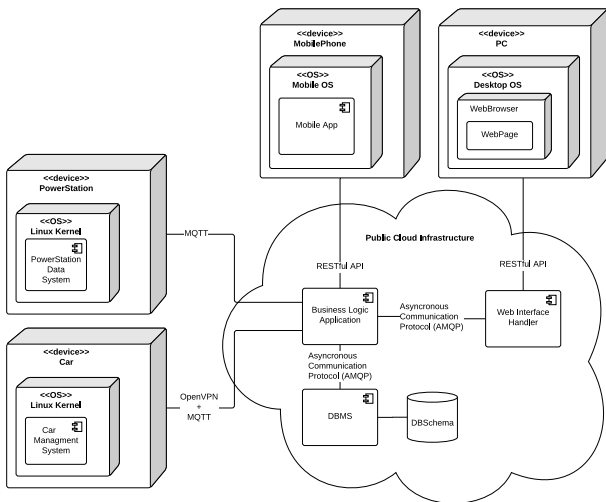
# II Deployment

- PowerStation: sistema di acquisizione dati
- Automobile: sistema di gestione dell'auto
- Clients:



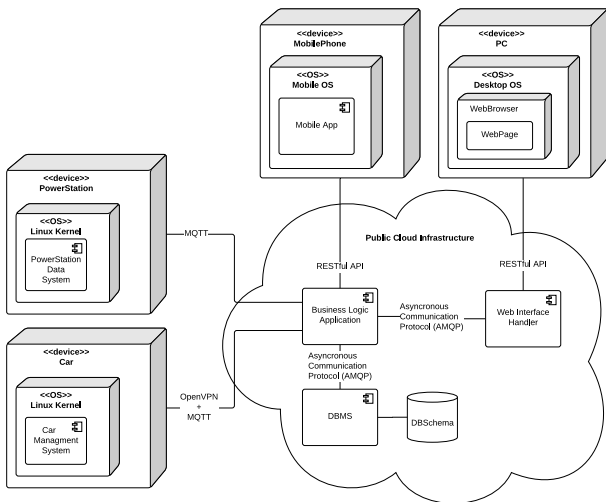
# II Deployment

- PowerStation: sistema di acquisizione dati
- Automobile: sistema di gestione dell'auto
- Clients:
  - Mobile App



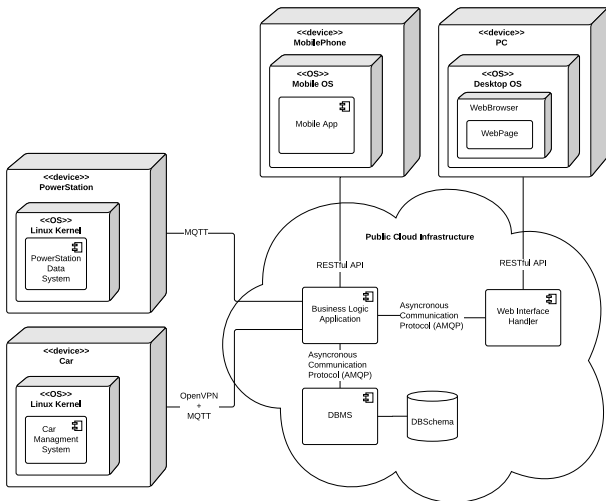
# II Deployment

- PowerStation: sistema di acquisizione dati
- Automobile: sistema di gestione dell'auto
- Clients:
  - Mobile App
  - Web App



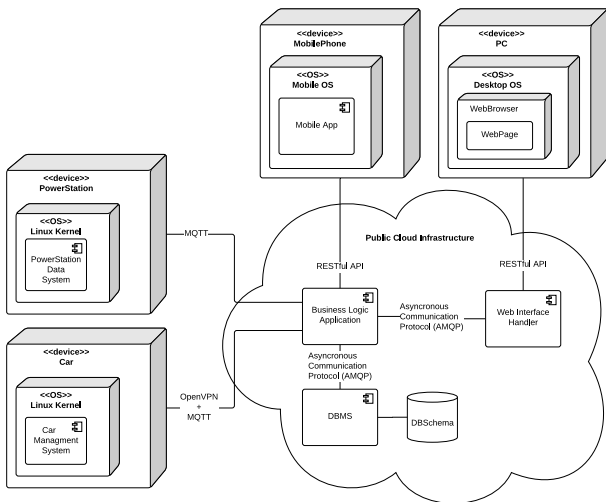
# II Deployment

- PowerStation: sistema di acquisizione dati
- Automobile: sistema di gestione dell'auto
- Clients:
  - Mobile App
  - Web App
- Schema DBMS



# II Deployment

- PowerStation: sistema di acquisizione dati
- Automobile: sistema di gestione dell'auto
- Clients:
  - Mobile App
  - Web App
- Schema DBMS
- Microservizi



# Altro

Nel design document abbiamo poi ulteriormente approfondito:

# Altro

Nel design document abbiamo poi ulteriormente approfondito:

- interazioni tra componenti → sequence diagram



# Altro

Nel design document abbiamo poi ulteriormente approfondito:

- interazioni tra componenti → sequence diagram
- interfaccia utente → UX diagram

# Altro

Nel design document abbiamo poi ulteriormente approfondito:

- interazioni tra componenti → sequence diagram
- interfaccia utente → UX diagram
- struttura del database → ER diagram

# Altro

Nel design document abbiamo poi ulteriormente approfondito:

- interazioni tra componenti → sequence diagram
- interfaccia utente → UX diagram
- struttura del database → ER diagram
- algoritmi: ottimizzazione posizionamento delle auto con MoneySavingOption

## **Test di integrazione**

# Criteri di ingresso

Prima di entrare nella fase di integrazione e relativo testing è necessario che le seguenti condizioni siano verificate.

# Criteri di ingresso

Prima di entrare nella fase di integrazione e relativo testing è necessario che le seguenti condizioni siano verificate.

- L'**ambiente** di test deve essere pronto.

# Criteri di ingresso

Prima di entrare nella fase di integrazione e relativo testing è necessario che le seguenti condizioni siano verificate.

- L'**ambiente** di test deve essere pronto.
- Tutti i **test di unità** devono essere completati.

# Criteri di ingresso

Prima di entrare nella fase di integrazione e relativo testing è necessario che le seguenti condizioni siano verificate.

- L'**ambiente** di test deve essere pronto.
- Tutti i **test di unità** devono essere completati.
- Le **dipendenze** tra i moduli devono essere definite.



# Criteri di ingresso

Prima di entrare nella fase di integrazione e relativo testing è necessario che le seguenti condizioni siano verificate.

- L'**ambiente** di test deve essere pronto.
- Tutti i **test di unità** devono essere completati.
- Le **dipendenze** tra i moduli devono essere definite.
- Per ogni caso di test devono essere definiti gli **input** e i relativi **output** attesi.

# Criteri di ingresso

Prima di entrare nella fase di integrazione e relativo testing è necessario che le seguenti condizioni siano verificate.

- L'**ambiente** di test deve essere pronto.
- Tutti i **test di unità** devono essere completati.
- Le **dipendenze** tra i moduli devono essere definite.
- Per ogni caso di test devono essere definiti gli **input** e i relativi **output** attesi.
- Devono essere pronti gli **stub** delle parti del sistema che non sono ancora state implementate.

# Strategia di integrazione

Sono da **integrare** tutti i moduli mostrati nella fase di design.

# Strategia di integrazione

Sono da **integrare** tutti i moduli mostrati nella fase di design.

- Abbiamo deciso di utilizzare un approccio **bottom-up**, cioè di integrare man mano i componenti che hanno meno dipendenze.

# Strategia di integrazione

Sono da **integrare** tutti i moduli mostrati nella fase di design.

- Abbiamo deciso di utilizzare un approccio **bottom-up**, cioè di integrare man mano i componenti che hanno meno dipendenze.
- In questo modo non è necessaria la creazione di stub, ma solo di **driver**.

# Strategia di integrazione

Sono da **integrare** tutti i moduli mostrati nella fase di design.

- Abbiamo deciso di utilizzare un approccio **bottom-up**, cioè di integrare man mano i componenti che hanno meno dipendenze.
- In questo modo non è necessaria la creazione di stub, ma solo di **driver**.
- Inoltre risulta più **semplice** verificare il comportamento dei componenti meno integrati testando i componenti di base il **prima possibile**.

# Strategia di integrazione

Sono da **integrare** tutti i moduli mostrati nella fase di design.

- Abbiamo deciso di utilizzare un approccio **bottom-up**, cioè di integrare man mano i componenti che hanno meno dipendenze.
- In questo modo non è necessaria la creazione di stub, ma solo di **driver**.
- Inoltre risulta più **semplice** verificare il comportamento dei componenti meno integrati testando i componenti di base il **prima possibile**.
- In questo modo tuttavia risulta molto difficile scovare errori nei componenti che vengono integrati alla fine.

# Strategia di integrazione

Sono da **integrare** tutti i moduli mostrati nella fase di design.

- Abbiamo deciso di utilizzare un approccio **bottom-up**, cioè di integrare man mano i componenti che hanno meno dipendenze.
- In questo modo non è necessaria la creazione di stub, ma solo di **driver**.
- Inoltre risulta più **semplice** verificare il comportamento dei componenti meno integrati testando i componenti di base il **prima possibile**.
- In questo modo tuttavia risulta molto difficile scovare errori nei componenti che vengono integrati alla fine.
- Inoltre risulta impossibile vedere l'intero sistema funzionante prima della fine del processo.