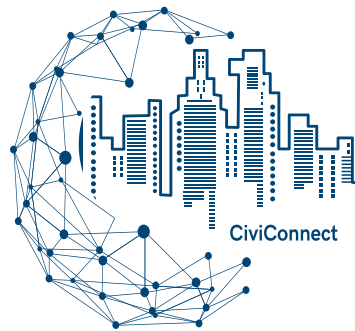




Laurea Triennale in informatica-Università di Salerno  
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci e Prof. F. Palomba



# Design Pattern Document CiviConnect

<b>Riferimento</b>	C14_ODD_ver.0.1
<b>Versione</b>	1.0
<b>Data</b>	12/10/2024
<b>Destinatario</b>	ANCI-DTD, Prof. Filomena Ferrucci, Prof. Fabio Palomba
<b>Presentato da</b>	ConnecTeam



**Approvato da**

Laurea Magistrale in informatica-Università di Salerno  
Corso di *Gestione dei Progetti Software*- Prof.ssa F.Ferrucci e Prof. F.Palomba

Bacco Luigi, Scala Benedetto



Laurea Triennale in informatica-Università di Salerno  
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci e Prof. F. Palomba

## Revision History

Data	Versione	Descrizione	Autori
14/11/2024	0.1	Prima stesura Obiettivo del sistema	MB
16/11/2024	0.2	Stesura Capitolo 1	MB
18/11/2024	0.3	Definizione Design Pattern	Tutti
15/12/2024	1.0	Revisione Generale del documento	Tutto il Team



## Team Members

Nome	Acronimo	Contatto
Luigi Bacco	LB	<a href="mailto:l.bacco2@studenti.unisa.it">l.bacco2@studenti.unisa.it</a>
Benedetto Scala	BS	<a href="mailto:b.scala1@studenti.unisa.it">b.scala1@studenti.unisa.it</a>
Generoso Sorridi	GSO	<a href="mailto:g.sorridi@studenti.unisa.it">g.sorridi@studenti.unisa.it</a>
Marco Brescia	MB	<a href="mailto:m.brescia16@studenti.unisa.it">m.brescia16@studenti.unisa.it</a>
Francesco Faiella	FF	<a href="mailto:f.faiella16@studenti.unisa.it">f.faiella16@studenti.unisa.it</a>
Gianluigi Citro	GC	<a href="mailto:g.citro37@studenti.unisa.it">g.citro37@studenti.unisa.it</a>
Giuseppe Speranza	GSP	<a href="mailto:g.speranza15@studenti.unisa.it">g.speranza15@studenti.unisa.it</a>
Manuel Cieri	MC	<a href="mailto:m.cieri1@studenti.unisa.it">m.cieri1@studenti.unisa.it</a>
Domenico Auriemma	DA	<a href="mailto:d.auriemma4@studenti.unisa.it">d.auriemma4@studenti.unisa.it</a>
Giuseppe Gambardella	GG	<a href="mailto:g.gambardella27@studenti.unisa.it">g.gambardella27@studenti.unisa.it</a>

## Sommario

Revision History .....	3
Team Members .....	4
Sommario .....	4
1. Introduzione .....	5
1.1 Linee Guida per la stesura del codice .....	5
1.2 Definizioni, Acronimi e Abbreviazioni .....	6
1.3 Riferimenti .....	6
2. Elementi di Riuso .....	7
2.1 Design Pattern .....	7
3. Glossario .....	10



## 1. Introduzione

---

*CiviConnect* ha come obiettivo principale quello di fornire una piattaforma che permetta ai cittadini di segnalare pericoli, dissesti e malfunzionamenti riscontrate sul territorio urbano. Il sistema permette ai cittadini di comunicare in modo efficace le problematiche all'amministrazione locale e di partecipare attivamente al processo di cura e miglioramento della città, contribuendo a rendere l'ambiente urbano più sicuro e vivibile per tutti.

In questa prima sezione del documento verranno elencate: le linee guida per la fase di implementazione, la nomenclatura e la documentazione.

### 1.1 Linee Guida per la stesura del codice

Le linee guida includono una lista di standard che gli sviluppatori devono seguire per la progettazione delle interfacce. Di seguito viene presentata una lista di link alle documentazioni ufficiali:

- <https://dart.dev/effective-dart/style>



## 1.2 Definizioni, Acronimi e Abbreviazioni

Si elencano tutte le definizioni, acronimi e abbreviazioni presenti all'interno del documento:

- **Categoria Segnalazione:** Rifiuti, Dissesto stradale, Manutenzione, Illuminazione
- **Stato Segnalazione:** In Verifica, Accettata, In lavorazione, Completata, Scartata
- **Priorità Segnalazione:** Bassa, Media, Alta, Non impostata
- **GU:** Gestione Utente Cittadino
- **GA:** Gestione Admin
- **GSC:** Gestione Segnalazione Cittadino
- **GSA:** Gestione Segnalazione Comune
- **AD:** Analisi Dati - Report
- **RF:** Requisito Funzionale
- **RNF:** Requisito Non Funzionale
- **AC:** Amministratore Comunale
- **DG:** Design Goal
- **DAO:** Data Access Object
- **RAD:** Requirement Analysis Document

## 1.3 Riferimenti

Di seguito una lista di riferimenti ad altri documenti utili durante la lettura:

- [Statement of Work](#)
- [Requirements Analysis Document](#)
- [System Design Document](#)
- [Matrice di tracciabilità](#)



## 2. Elementi di Riuso

---

### 2.1 Design Pattern

#### **Design Pattern: Proxy**

Il *Proxy* è un design pattern strutturale che introduce un intermediario o sostituto per controllare l'accesso a un oggetto reale. Questo approccio consente di nascondere al sistema la complessità o il costo associato alla creazione e gestione dell'oggetto, ad esempio nel caso di risorse che richiedono un notevole impiego di tempo o memoria (come risorse remote o oggetti di grandi dimensioni).

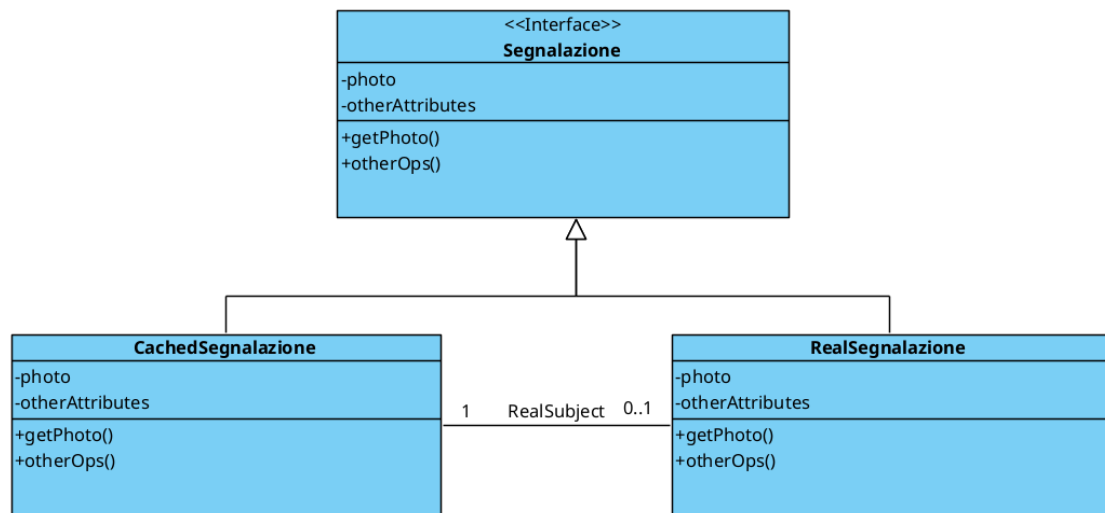
Attraverso il *Proxy*, è possibile:

- Garantire un alto livello di disaccoppiamento, migliorando la manutenibilità e la sicurezza del sistema.
- Ottimizzare le prestazioni, utilizzando tecniche come il caricamento ritardato (*lazy loading*), che permette di istanziare l'oggetto reale solo quando è effettivamente necessario.

In questo modo, il pattern contribuisce a rendere il sistema più efficiente e robusto.

In *CiviConnect*, si usa il design pattern *Proxy* per evitare di caricare immediatamente le immagini associate agli oggetti segnalazione, migliorando così le prestazioni dell'applicazione. Il *Proxy* agisce come intermediario, caricando le immagini solo quando necessario e rendendo l'app più efficiente.

Di seguito un esempio di *Proxy* nel nostro sistema:



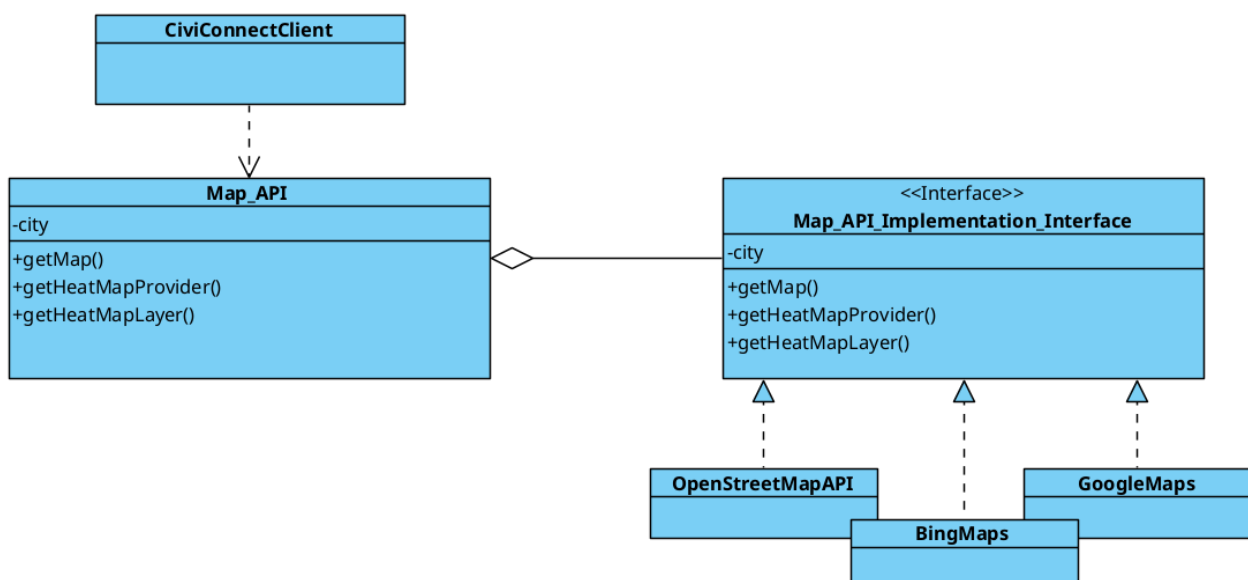


## Design Pattern: Bridge

Il design pattern *Bridge* è un pattern strutturale che permette di separare un'astrazione dalla sua implementazione, facilitando l'estensione e la modifica di entrambi senza interferire l'uno con l'altro. Questa suddivisione migliora la manutenibilità del codice, permettendo di aggiornare o sostituire singolarmente l'astrazione o l'implementazione senza influenzare l'altra. Facilita, inoltre, l'estensibilità, consentendo l'aggiunta di nuove funzionalità o implementazioni in modo modulare e senza rischi di duplicazione o complessità eccessiva.

L'astrazione definisce le operazioni di alto livello, mentre l'implementazione si occupa dei dettagli concreti. Questo approccio promuove la composizione, migliorando la flessibilità del codice.

In CiviConnect, il design pattern *Bridge* è strutturabile come interfaccia per disaccoppiare la logica implementativa del sistema all'implementazione API effettivamente usata per caricare la struttura e i dati delle mappe e il layer aggiuntivo per le heatmap, applicabile al requisito RF\_AD\_1.





### 3. Glossario

Terminologia	Definizione
Dart	Dart è un linguaggio di programmazione sviluppato da Google. Lo scopo di Dart è quello di sostituire JavaScript come lingua franca per lo sviluppo web sulla piattaforma "web aperto".
Design Pattern	Si tratta di una descrizione o modello logico da applicare per la risoluzione di un problema che può presentarsi in diverse situazioni durante le fasi di progettazione e sviluppo del software, ancor prima della definizione dell'algoritmo risolutivo della parte computazionale.
Proxy	Il Proxy è un design pattern strutturale che introduce un intermediario o sostituto per controllare l'accesso a un oggetto reale. Questo approccio consente di nascondere al sistema la complessità o il costo associato alla creazione e gestione dell'oggetto.
Bridge	Il Bridge è un design pattern strutturale che separa un'astrazione dalla sua implementazione, permettendo di modificarle indipendentemente. Questo approccio facilita la flessibilità e la scalabilità del sistema, consentendo di gestire separatamente l'interfaccia dall'implementazione sottostante.