Smart Greenhouse

Progetto di Laboratorio di sistemi software

Mengozzi Maria, Vitali Anna, Yan Elena

Introduzione

- Per il progetto si è pensato di realizzare un'applicazione che consenta la **gestione** e il **monitoraggio** di una **serra intelligente**
- All'interno della serra sarà coltivata una sola tipologia di pianta monitorata dal **4 sensori** che rilevano:
  - luminosità
  - temperatura ambientale
  - umidità dell'aria
  - umidità del terreno
- Il sistema sarà caratterizzato da 4 componenti:
  - un modulo Arduino
  - un server
  - un'applicazione desktop
  - un'applicazione mobile

Strumenti utilizzati

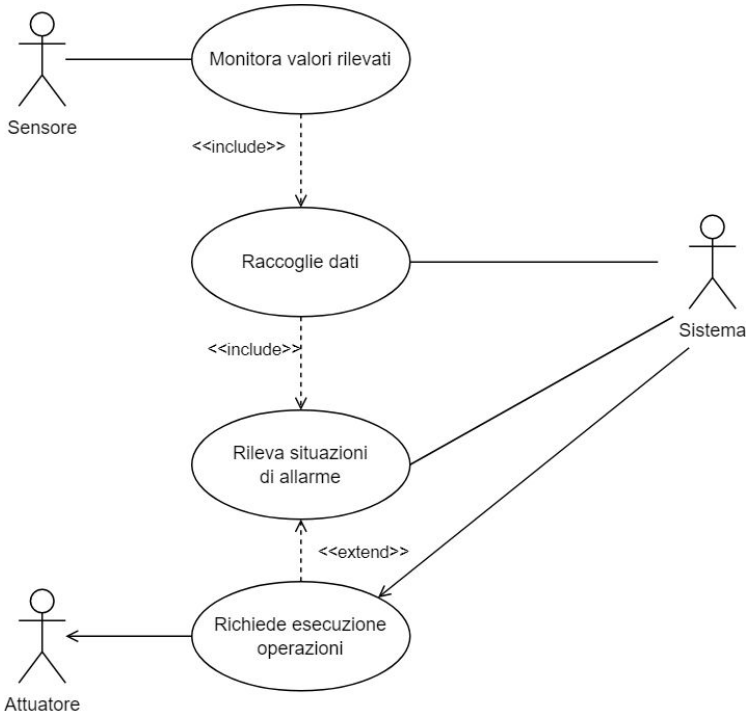
- Gli strumenti impiegati per automatizzare diversi aspetti del progetto sono stati i seguenti:
  - **Notion**, come tool collaborativo per la gestione del lavoro
  - **Egon**, come supporto per il Domain storytelling
  - **GitHub**, come servizio di hosting per il codice
  - **Gradle**, per la build automation
  - **GitHubAction**, per promuovere la Continuous Integration
  - **UnitTest ed Android espresso**, per la verifica delle funzionalità del server e dei client
  - **MongoDB**, come sistema di storage
  - **Vert.x**, per promuovere la realizzazione di applicazioni reattive
  - **Docker**, come piattaforma per lo sviluppo

Knowledge crunching

- A seguito della richiesta del committente il team ha effettuato diverse interviste con gli esperti del dominio, instaurando il processo di **knowledge crunching**
- Durante il processo:
  - sono stati raffinati i **requisiti** del sistema
  - prodotti i **prototipi** delle due applicazioni da realizzare
  - individuati i sottodomini e la loro **classificazione strategica**

Esempio Diagramma dei casi d'uso

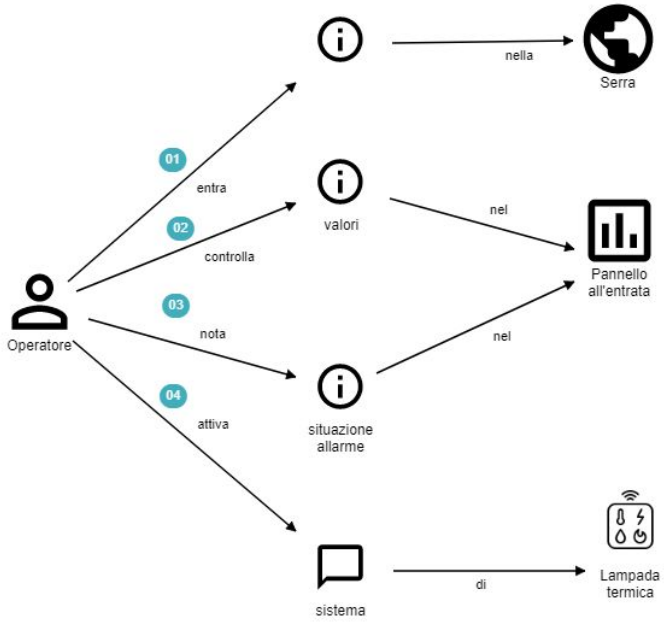
- caso d'uso relativo al **monitoraggio** dei **valori** della pianta e della **rilevazione** di situazioni di **allarme**
- attori
  - **sensore** → rileva parametro della pianta
  - **sistema** → raccoglie dati monitorati e rileva situazioni di allarme
  - **attuatore** → esegue le operazioni correttive richieste dal sistema



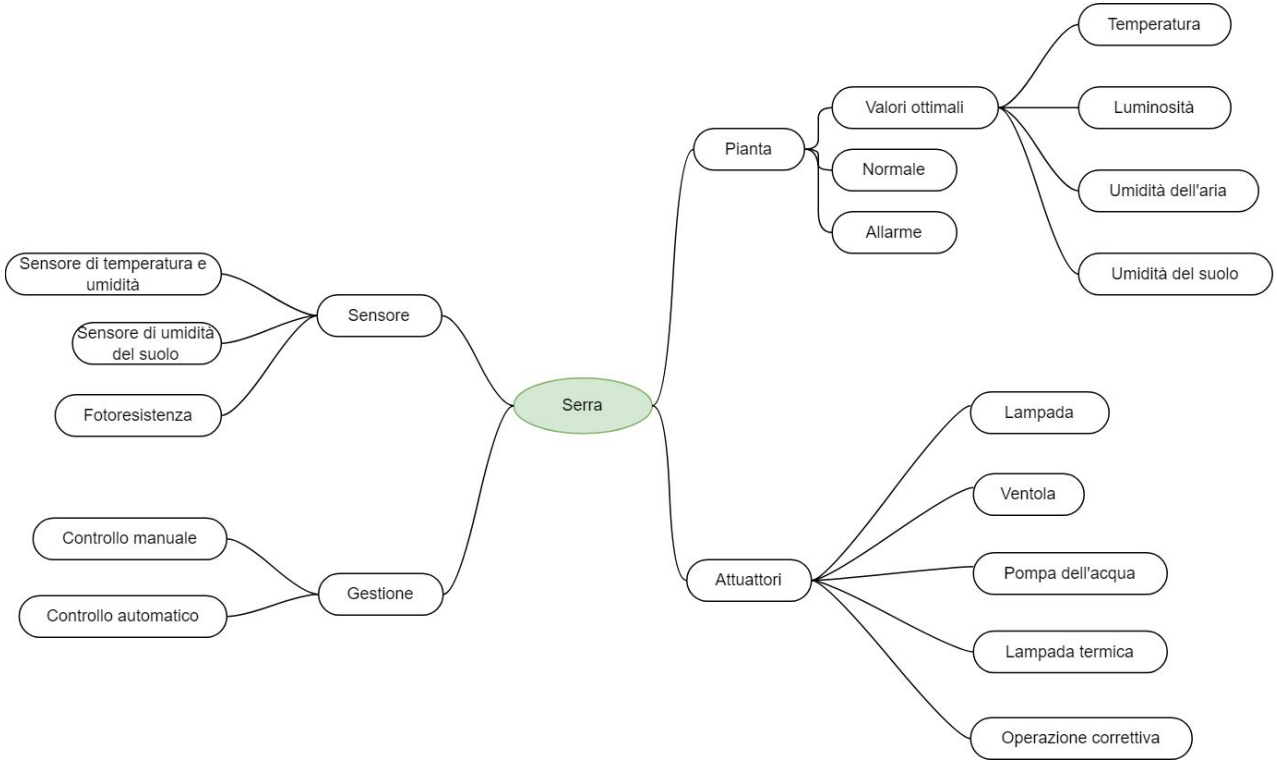
Esempio di Domain Story Telling

Analista: Come avviene la gestione della serra da parte tua, quali sono le operazioni che svolgi?

Esperto (operatore): Quando entro nella serra, per prima cosa guardo il pannello all' entrata e controllo i valori dell'umidità dell'aria e della temperatura, se mi rendo conto che non vanno bene ad esempio se la temperatura è troppo bassa, attivo le lampade termiche per un po' fino a quando non noto un'innalzamento.

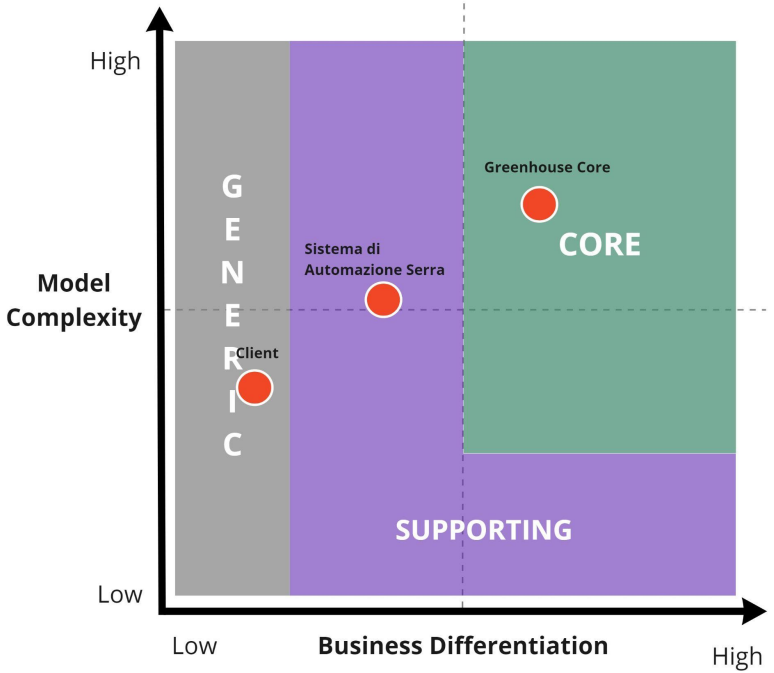


Ubiquitous language



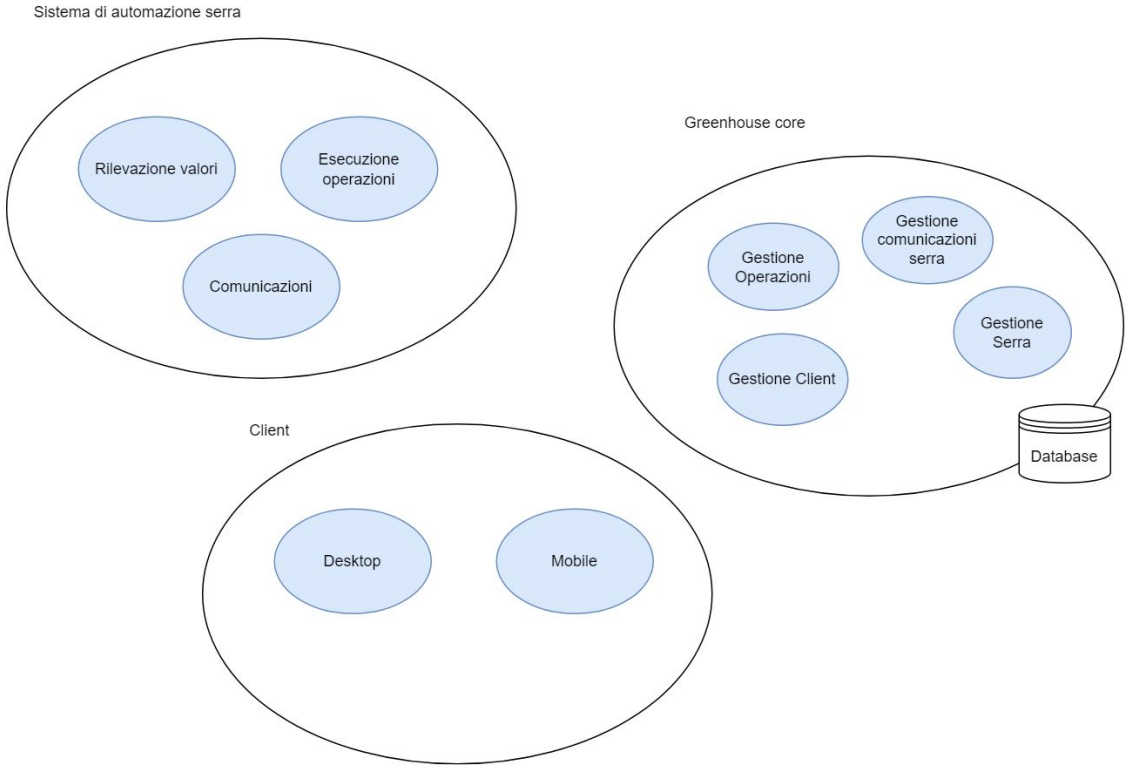
Subdomain

- i **sottodomini** individuati sono tre:
  - **Sistema di automazione serra**
  - **Greenhouse core**
  - **Client**
- ciascuno costituito da un certo numero di **bounded-context**

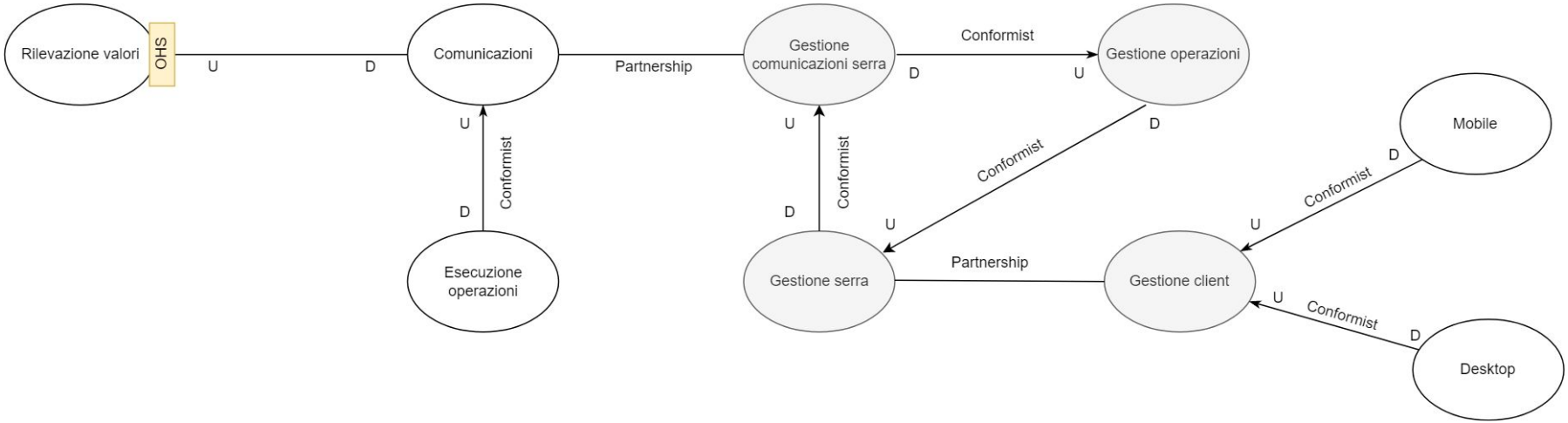


Bounded context

- i **bounded context** sono:
  - rilevazione valori
  - esecuzione operazioni
  - comunicazioni
  - gestione serra
  - gestione operazioni
  - gestione comunicazioni serra
  - gestione client
  - desktop e mobile



Context map

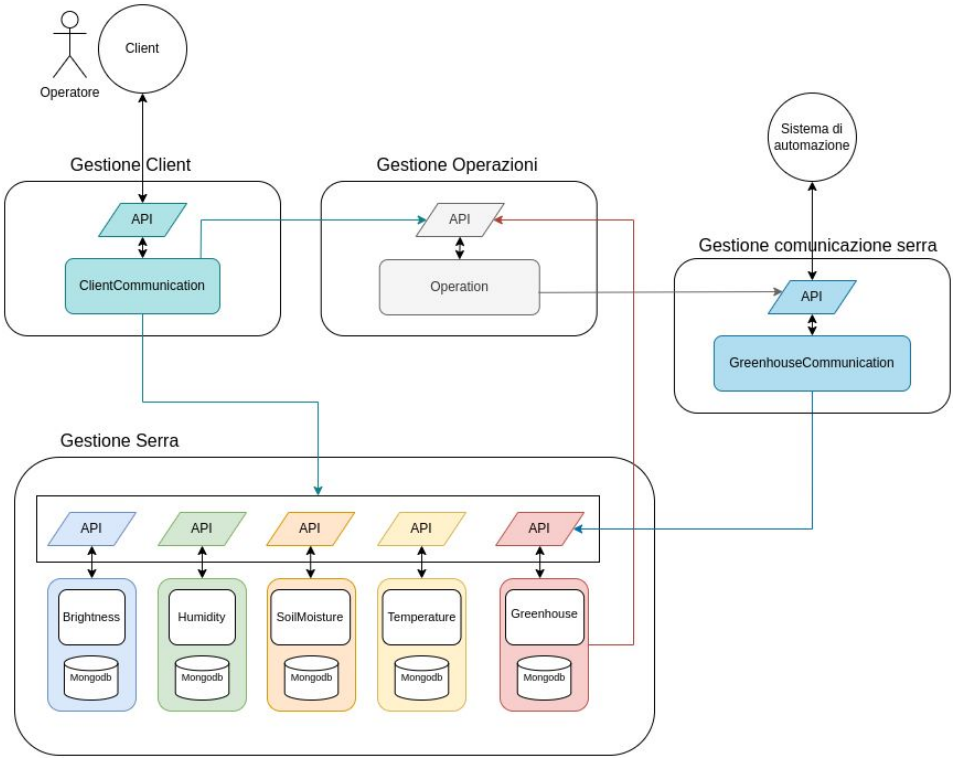


- le relazioni fra i bounded-context sono per lo più di tipo **customer-supplier**
- **rilevazione valor**i adotta un *Open Host Service* nei confronti del bounded context **comunicazioni**
- fra i bounded-context **comunicazioni** e **gestione comunicazioni** vi è una relazione di **partnership**

Bounded context e architettura a microservizi

Microservizi: componenti **indipendenti** che eseguono ciascun processo applicativo come un servizio

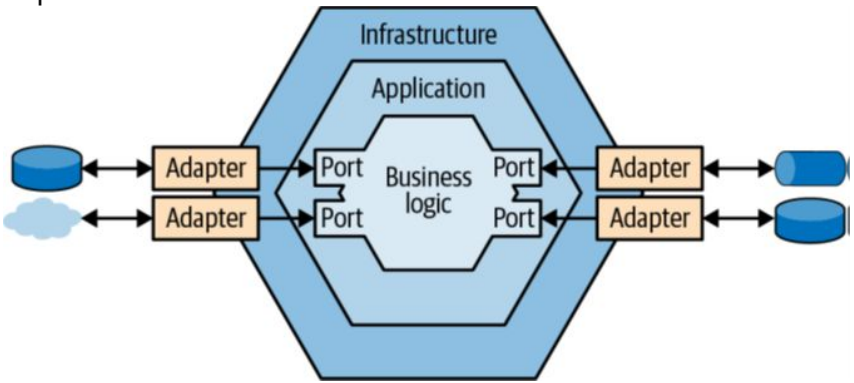
- comunicano attraverso la rete ed espongono un'**interfaccia** ben definita
- possono essere **aggiornati**, **distribuiti** e **ridimensionati** per rispondere alla richiesta di funzioni specifiche



Dettagli micro-servizio: Architettura Esagonale

Il microservizi sono stati realizzati tramite un'architettura **esagonale port and adapters**

- la logica di dominio è racchiusa nella parte più interna dell'architettura e definisce delle porte che devono essere implementate
- prevede l'applicazione del *principio di inversione delle dipendenze*
- gli adapter:
  - sono la concreta implementazione dell'interfaccia delle porte
  - filtrano la comunicazione



Pratiche DevOps

- Build automation
  - Gradle
    - ➜ utilizzata una struttura multi-progetto
    - ➜ definiti diversi task personalizzati
- Continuous Integration
  - GitHub actions
    - ➜ diversi workflows per i progetti
    - ➜ test multipiattaforma
    - ➜ automatic release
- Version control
  - Semantic Versioning
    - ➜ commit scritti seguendo l’approccio Conventional Commit

The end