

Medchain

Innovation in Healthcare



Indice

- 01.** Contesto e Motivazione
- 02.** Hyperledger Fabric
- 03.** HL7 FHIR
- 04.** Funzionalità del sistema
- 05.** Medical Records e Patient
- 06.** Demo
- 07.** Conclusione



Contesto

Nel contesto medico, la gestione dei dati sanitari è una questione critica che richiede alta sicurezza, privacy e integrità. Con l'aumento della digitalizzazione nel settore sanitario, è essenziale proteggere le informazioni sensibili dei pazienti da accessi non autorizzati e frodi, garantendo al contempo la disponibilità e l'accuratezza dei dati per i professionisti del settore.





Motivazione

L'utilizzo della tecnologia blockchain, offre una soluzione per affrontare queste sfide. Con una blockchain permissioned, solo le entità autorizzate, come ospedali, cliniche, medici possono accedere e aggiornare i dati. Questo garantisce tracciabilità, riduce il rischio di frodi e protegge i dati sensibili dei pazienti.

Hyperledger Fabric



Private

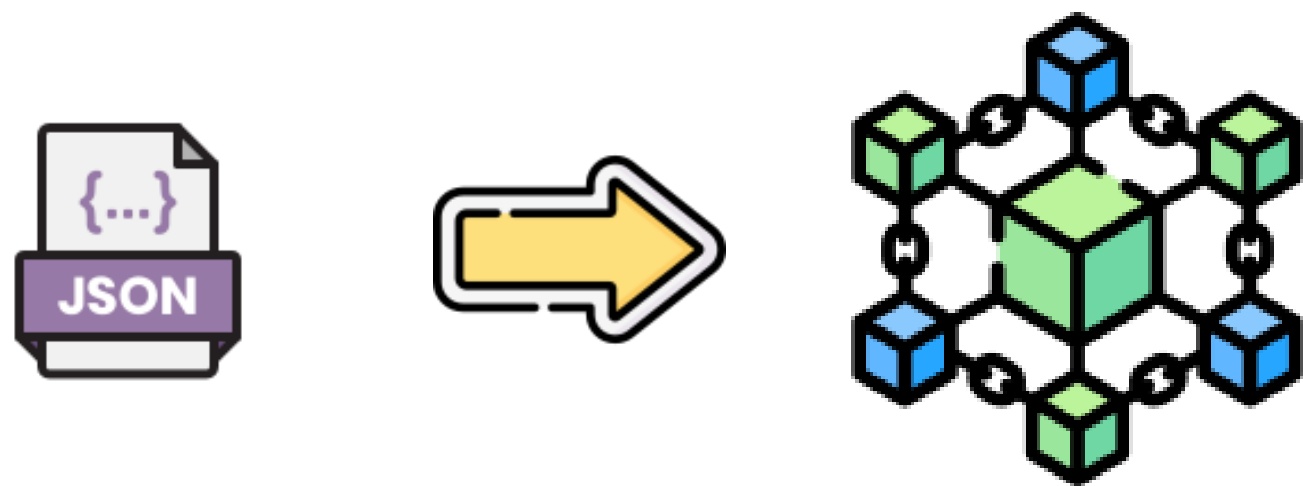
E' accessibile solo a un gruppo ristretto di partecipanti, appartenenti a una stessa organizzazione o consorzio; inoltre i partecipanti possono mantenere la riservatezza di transazioni e dati (sensibili).

Permissioned

Ogni partecipante deve ottenere il permesso per entrare nel network, permette di stabilire ruoli e responsabilità specifici per ciascun partecipante, migliorando la sicurezza e l'affidabilità del sistema.

Hyperledger Labs: Fablo

Configurare e gestire una rete personalizzata di Hyperledger Fabric è difficile. L'obiettivo principale di Fablo è rendere questo processo facile, diretto e accessibile. Trasforma un singolo file di configurazione in una rete funzionante.



Non sono necessari ulteriori flag, comandi o altri aggiustamenti successivi (contiene tutto come è stato dichiarato)

È flessibile quanto lo è Fabric. E' possibile avere molteplici organizzazioni, canali, chaincode e combinarli in vari modi



HL7 FHIR

Fast Healthcare Interoperability Resources

03.

Descrizione

E' uno standard per lo scambio di dati sanitari elettronici, sviluppato da Health Level Seven International

Obiettivo

Facilitare l'interoperabilità tra sistemi sanitari diversi, migliorando la condivisione e l'accesso ai dati sanitari

Modularità

Utilizza risorse modulari che rappresentano concetti clinici specifici, come pazienti, incontri, osservazioni e prescrizioni

Flessibilità

Adattabile a vari contesti e necessità cliniche, supportando sia dati strutturati che non strutturati

Applicazioni

Sorveglianza della salute pubblica, Supporto alle decisioni cliniche, Condivisione di documenti sanitari e Ricerca clinica

Funzionalità del Sistema

04.



Records



Users



Prescrizioni



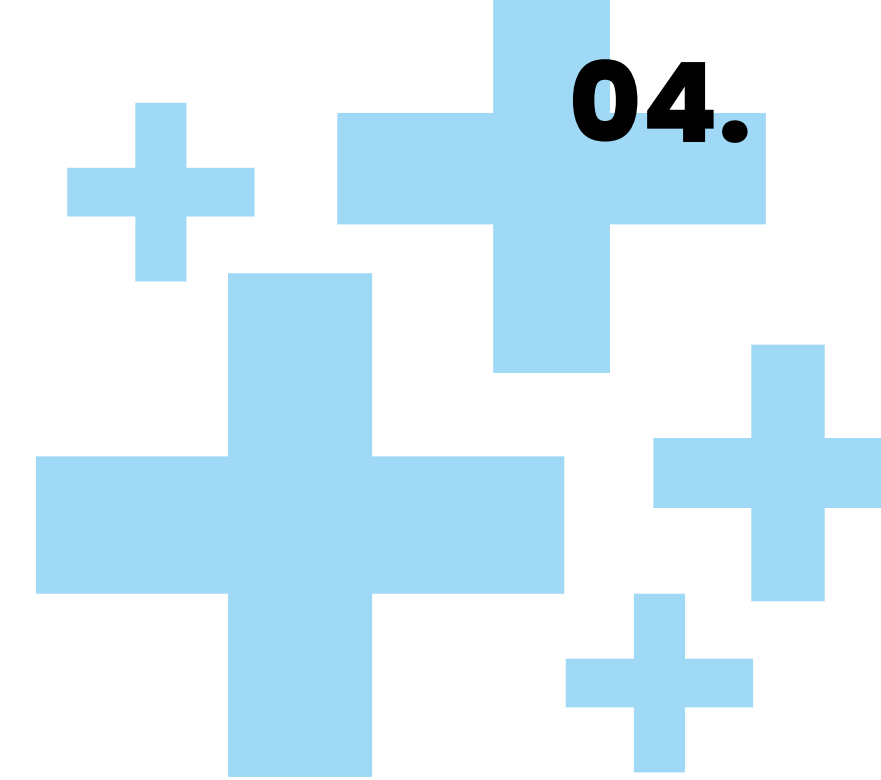
Risultati (Lab)



Incontri



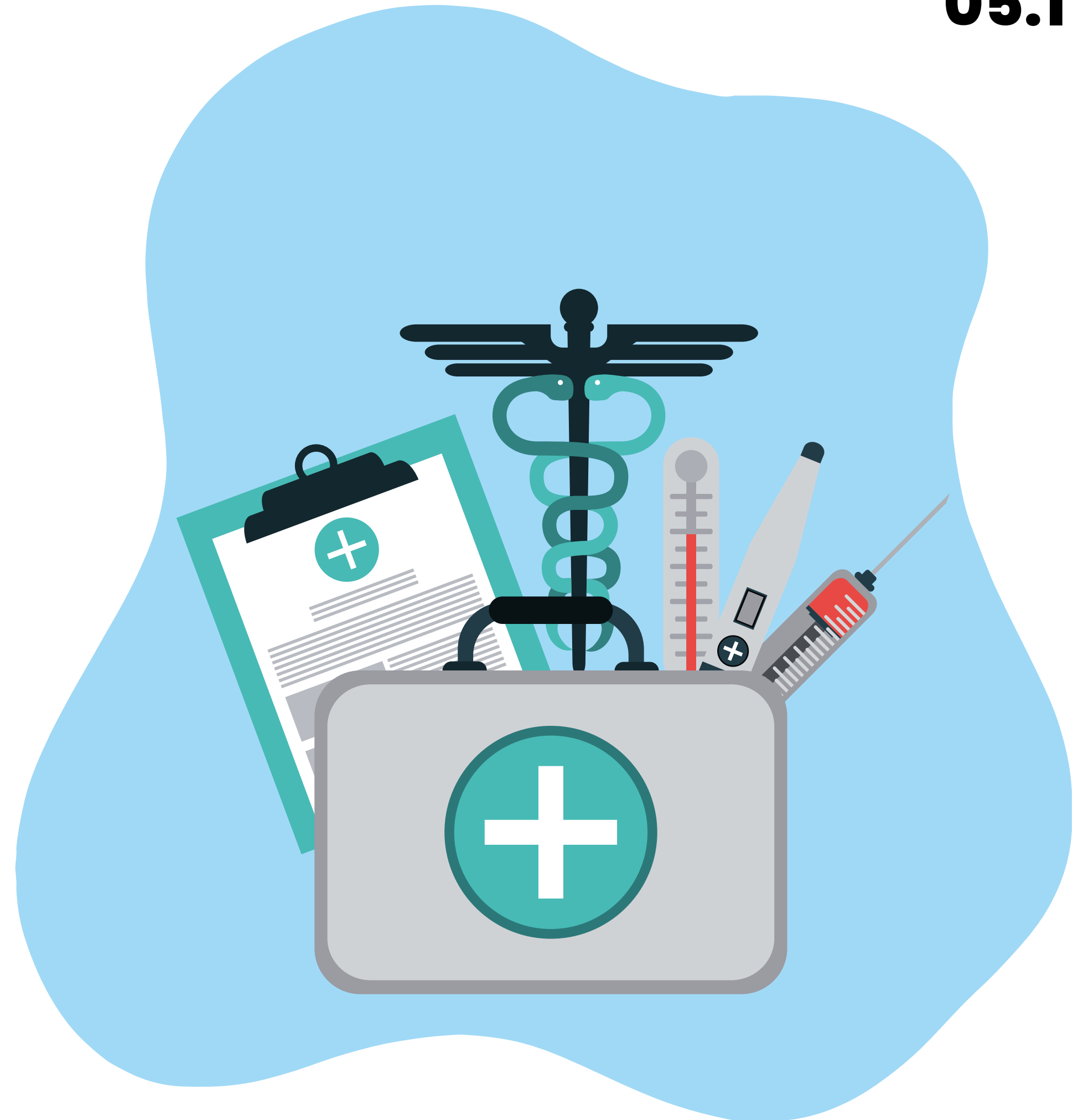
Autorizzazioni



Medical Records

Questo chaincode implementa un sistema per la gestione delle cartelle cliniche sulla blockchain e ne permette la creazione, lettura, aggiornamento e cancellazione (rispettando la relativa struttura), nonché la gestione in modalità batch per condizioni, procedure e allergie.

Quest' ultima è necessaria poichè vengono mantenute anche le sottostrutture delle cartelle cliniche. Ciò permette di gestire un insieme di dati complessi, garantendo consistenza e atomicità delle transazioni.





```
type MedicalRecords struct {  
    RecordID      string           `json:"identifier,omitempty"`  
    PatientID     string           `json:"patientID,omitempty"`  
    Allergies     []AllergyIntolerance `json:"allergies,omitempty"`  
    Conditions    []Condition       `json:"conditions,omitempty"`  
    Procedures    []Procedure       `json:"procedures,omitempty"`  
    Prescriptions []MedicationRequest `json:"prescriptions,omitempty"`  
    LabResultsIDs []string          `json:"labResultsIDs,omitempty"`  
}
```



```
CreateMedicalRecords(ctx contractapi.TransactionContextInterface, medicalRecordJSON string) (string, error)  
  
ReadMedicalRecords(ctx contractapi.TransactionContextInterface, recordID string) (string, error)  
  
UpdateMedicalRecords(ctx contractapi.TransactionContextInterface, recordID string, updatedMedicalRecordJSON string) (string, error)  
  
DeleteMedicalRecords(ctx contractapi.TransactionContextInterface, recordID string) (string, error)  
  
SearchMedicalRecords(ctx contractapi.TransactionContextInterface, queryString string) (string, error)
```

Patient

Questo chaincode implementa la gestione dei dati dei pazienti sulla blockchain e ne consente la creazione, lettura, aggiornamento e cancellazione, nonché la gestione delle autorizzazioni per l'accesso ai dati contenuti nella struttura fhir corrispondente.



```
type Authorization struct {  
    PatientID      string      `json:"patientId"`  
    Authorized     map[string]bool  `json:"authorized"`  
    AuthorizedOrgs map[string]bool  `json:"authorizedOrgs"`  
}
```





```
type Patient struct {
    ID                *Identifier    `json:"identifier"`
    Active            bool           `json:"active,omitempty"`
    Name              *HumanName     `json:"name,omitempty"`
    Telecom           []ContactPoint `json:"telecom,omitempty"`
    Gender            *Code          `json:"gender,omitempty"`
    BirthDate         time.Time      `json:"date,omitempty"`
    Deceased          bool           `json:"deceased,omitempty"`
    Address           []Address      `json:"address,omitempty"`
    MaritalStatus     *CodeableConcept `json:"maritalstatus,omitempty"`
    MultipleBirth     []int          `json:"multiplebirth,omitempty"`
    Photo             *Attachment    `json:"photo,omitempty"`
    Contact           []Contact      `json:"contact,omitempty"`
    Communication     []Communication `json:"communication,omitempty"`
    GeneralPractitioner *Reference     `json:"generalpractitioner,omitempty"`
    ManagingOrganization *Reference     `json:"managingorganization,omitempty"`
}
```




```
CreatePatient(ctx contractapi.TransactionContextInterface, patientJSON string) (string, error)

ReadPatient(ctx contractapi.TransactionContextInterface, patientID string) (string, error)

UpdatePatient(ctx contractapi.TransactionContextInterface, patientJSON string) (string, error)

DeletePatient(ctx contractapi.TransactionContextInterface, patientID string) (string, error)

SearchPatients(ctx contractapi.TransactionContextInterface, queryString string) (string, error)
```

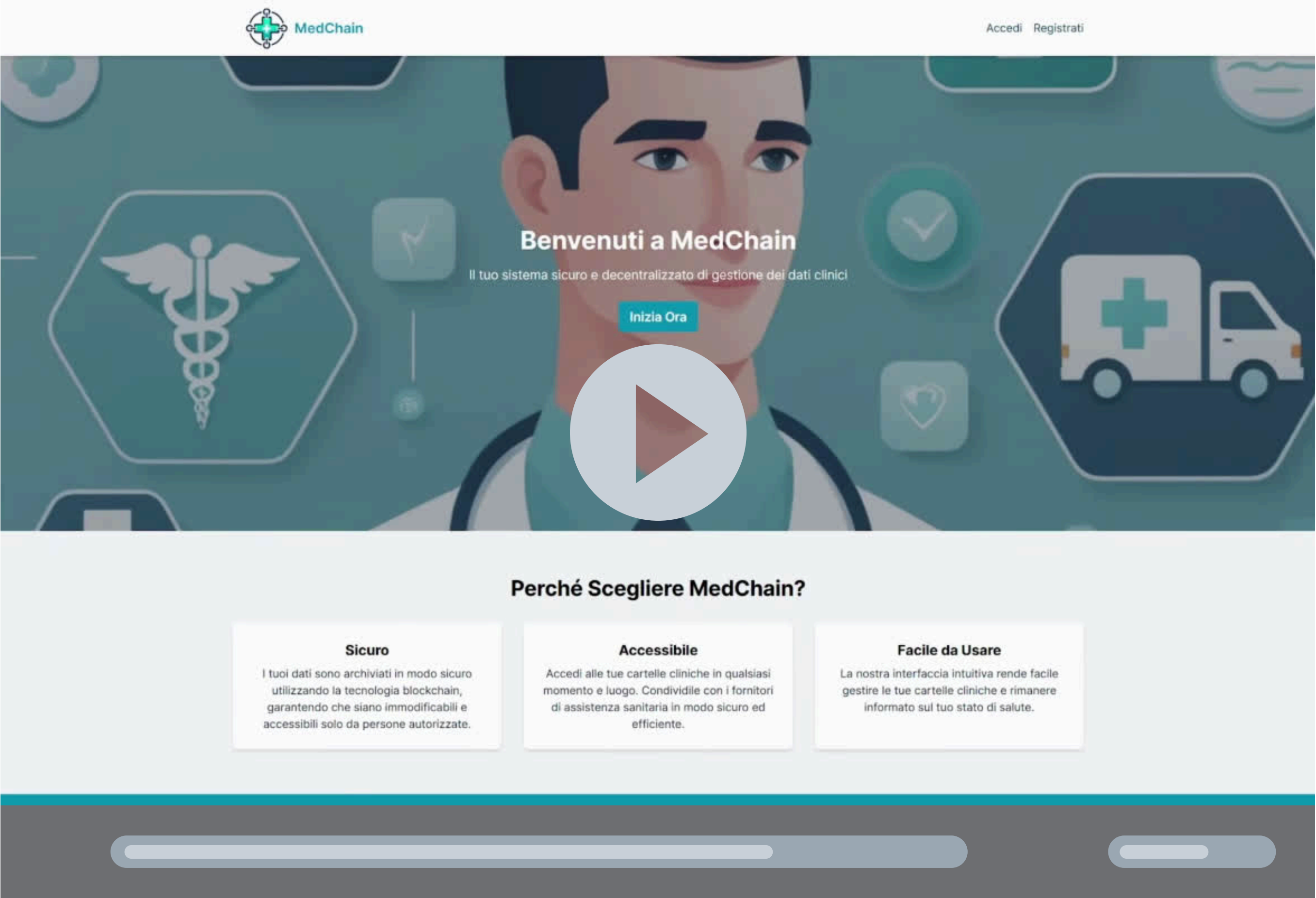


```
IsAuthorized(ctx contractapi.TransactionContextInterface, patientID string, clientID string) (bool, error)

RevokeAccess(ctx contractapi.TransactionContextInterface, patientID string, requesterID string, isOrg bool) (string, error)

RequestAccess(ctx contractapi.TransactionContextInterface, patientID string, requesterID string, isOrg bool) (string, error)

GrantAccess(ctx contractapi.TransactionContextInterface, patientID string, requesterID string, isOrg bool) (string, error)
```



Considerazioni

Complessità strutture dati

Nell'uso delle strutture FHIR una volta integrata l'interfaccia utente, in quanto è stato verificato più volte (in coincidenza ad eventuali modifiche) la correttezza degli input rispetto alla specifica.

Applicazione Labs: Fablo

E' stato necessario capire bene il funzionamento di Fablo per la generazione automatica della rete e definire una topologia di rete adatta allo scopo del progetto senza esagerare con la dimensione



Conclusione

Grazie per l'attenzione!



Medchain



Fhir Standard



Hyperledger
FABRIC

HLF

