**TESI MODULI MYDATA per app**

Repository GitHub: https://github.com/valentinaprotti/MyDataModuleTesi

Il mio lavoro di tesi si concentra sullo studio e lo sviluppo di un servizio modulare che possa consentire la gestione dei permessi di condivisione di dati personali, in riferimento inizialmente ad una particolare architettura (nella fattispecie quella di SMAll) e successivamente studiato e realizzato per essere esteso a servizio generico riutilizzabile in diverse circostanze in seguito ad opportuni adattamenti.

Nello studio iniziale e preventivo del progetto di tesi mi sono avvalsa della documentazione prodotta da una mia collega nel suo elaborato, nel quale è andata ad analizzare e progettare una base di partenza semplificata sotto forma di applicativo desktop di un complesso progetto, che risponde al nome MyData.

MyData è un progetto ormai entrato in fase avanzata di sviluppo, curato da un team di ricerca finlandese in collaborazione con il dipartimento del trasporto pubblico. In tal senso MyData nasce come concetto astratto, idea di un supporto per l’utente che oggi è sempre più sensibile al trattamento dei propri dati personali, sia da parte di quali entità o soggetti, sia nello specifico a quali aspetti della loro sfera personale queste entità abbiano accesso.

In questa ottica il lavoro di Giada ha portato all’analisi e al successivo sviluppo di alcune entità in gioco, ovvero (…). Dopo opportuna scrematura per rendere il proprio elaborato adatto ad una realtà ben più semplificata di quella studiata nel caso generale dal team di MyData, il prodotto finale ottenuto è quel che chiamerei un prototipo di funzionamento dell’intera architettura, codificato e comprensivo di tutte le funzionalità. Proprio per questa ragione, per la sua completezza, il prototipo da lei realizzato non può che consistere in un punto di partenza per successivo sviluppo, ad esempio per rendere l’architettura modulare e consentire che molteplici servizi non provenienti tutti da un’unica fonte (la stessa del prototipo, in questo caso) possano usufruire delle funzionalità di MyData, delegando perciò ad un’unica entità di elaborazione il trattamento personalizzato da parte dell’utente dell’accesso ai propri dati.

Il mio lavoro ha quindi previsto una fase iniziale di studio del funzionamento dell’architettura, in ottica di inclusione e rimodellamento al fine di renderla funzionale per il caso di studio desiderato, ovvero SMAll.

SMAll, Smart Mobility for All, è un progetto nato e sviluppato all’interno dell’Università di Bologna, e pensato in particolare per facilitare l’utilizzo di mezzi di trasporto pubblici (inizialmente della rete di trasporti bolognese, ma in ottica di generalità) a soggetti ipovedenti.

Nella sua realizzazione più immediata, ovvero quella di una applicazione per dispositivi mobili della piattaforma Android, SMAll (realizzata sotto il nome di GoBoBus) consente di (…).

Nell’ulteriore avanzamento del progetto si è pensato di decentralizzare lo scambio dei dati raccolti in tempo reale, in modo da alleggerire da parte delle task il single point of failure costituito dalla rete cloud centralizzata a cui fanno capo tutti i dispositivi interessati ad ottenere informazioni sul trasporto pubblico attraverso questa piattaforma. In questa ottica si è pensato fosse di interesse inserire un modulo, o una serie di moduli, che potessero frapporsi tra l’utente e l’effettivo uso dei dati da lui forniti (ritardo, attuale posizione del bus, propria posizione, ecc). In questo modo l’utente può avere completo controllo su quali dati permettere agli altri dispositivi di accedere, gestendo i propri consensi in maniera chiara e trasparente.

Lo sviluppo modulare ha quindi inizialmente assunto come unico servizio d’interesse quello dello scambio di dati tra dispositivi della rete di SMAll. Non avendo per ora accesso a quelli che saranno effettivamente i dati scambiati tra gli utilizzatori del servizio, ovvero tra i dispositivi, ho provveduto a sviluppare una applicazione di testing dalla quale estrarre al momento opportuno i moduli necessari ed effettuarne il porting con le dovute variazioni all’interno dell’applicazione vera e propria.

Il primo passo dello sviluppo è stato la scelta di quali moduli fossero strettamente necessari all’interno dell’applicazione e quali si potessero invece gestire a livello centralizzato tramite cloud, in modo da non appesantire inutilmente l’applicazione mobile.

Uno studio delle entità presenti nell’analisi progettuale di Giada ha portato a distinguere quattro circostanze separate e tra loro in collaborazione, quali:  
- la gestione e il matenimento di Utenti MyData (ovvero l’Operatore)  
- la registrazione dei Servizi presso un Registry  
- la gestione dei Consent (permessi) di diverse tipologie associati agli Account dei singoli Utenti presso i Servizi (Consent Manager)  
- il mantenimento dei dati effettivi da trattare in un Personal Data Vault

Come si può notare dalla sola descrizione degli ambiti di interesse delle singole funzionalità, esse si distinguono tra: contesto di utilizzo di dati pressoché immutabili dal punto di vista di un utilizzatore (i primi due punti); contesto di utilizzo piuttosto frequente di dati e potenziali accessi multipli in una stessa sessione di utilizzo (gli ultimi due punti).

Ho perciò ipotizzato in prima battuta di mantenere su cloud le funzionalità caratteristiche dell’Operatore MyData, che conserva i dati dei propri Utenti, e del Service Registry, che consente invece la registrazione presso la rete dei Servizi che abbiano interesse a trattare i dati utilizzati mediante il supporto di MyData. Come da precedenti premesse, nel nostro singolo caso di studi (salvo successivi sviluppi) l’intero pool di Utenti interessati è quello degli Utenti SMAll (o GoBoBus), ovvero coloro che fanno utilizzo dell’applicazione di mobilità, e conseguentemente l’unico servizio interessato da MyData è proprio quello di SMAll (GoBoBus).

Allo stesso tempo si è inizialmente ipotizzato che si potesse regolare la concessione dei permessi (Consent) tra Account degli utenti e Servizi, tramite Consent Manager, a livello centralizzato. Se da un lato, come si vedrà meglio grazie alla più approfondita distinzione tra i due fondamentali tipi di Consent esistenti (...), si può pensare che anche la concessione e la successiva gestione dei ServiceConsent sia più opportunamente da delegarsi ad un servizio cloud, la necessità di interazioni più rapide e frequenti per la concessione di DataConsent durante le singole interazioni ha fatto sì che si optasse in questo caso per uno sviluppo modulare all’interno dell’applicazione stessa. Mantenuto localmente anche il Personal Data Vault, ovvero la vera e propria base di dati dell’utente che i servizi hanno interesse a scambiare (nel nostro caso, ad esempio, la posizione GPS dell’utente).

Lato fondamentale da tenere in considerazione nello sviluppo dei moduli è il formato in cui i dati vengono rappresentati: nel nostro caso definire a priori il tipo di dato scambiato non costituirebbe un limite, essendo quello per cui è costruito inizialmente l'unico servizio da considerare, ma in ottica di estendibilità e generalizzazione si è optato per mantenere la scelta implementativa di raccogliere i dati in DataSource, affiancando a ciascun dato una stringa che rappresentasse la classe o il tipo primitivo in esso contenuto.

A livello di interfaccia grafica si è deciso di mantenere lo stile di interfaccia dell'applicazione GoBoBus, preservando sia la facilità di utilizzo che lo stile scelto, in ottica di utilizzo da parte di utenti ipovedenti (quindi adottando soluzioni come Google TalkBack, come all'interno del resto dell'applicazione). L'accesso alla gestione dei Consent è stato quindi pensato per essere inserito come pulsante aggiuntivo nella schermata di impostazioni del proprio profilo, e per contenere al suo interno i meccanismi per i quali l'utente ha la possibilità di scegliere di disattivare tutti o parte dei consensi alla condivisione dei dati, oppure revocare completamente tali permessi. Nel predisporre l'interfaccia grafica si è scelto in prima battuta di salvare le preferenze di condivisione dei singoli dati tra le shared preferences, dal momento che non esiste un modo semplice di estrapolare tale informazione tramite, ad esempio, i Data Consent emessi: ciò accade perché un utente può voler modificare le proprie preferenze in tal senso in modo "asincrono" rispetto allo scambio effettivo dei dati, rendendo così plausibile che le informazioni contenute nell'ultimo Data Consent registrato siano nel mentre diventate obsolete.

Ho così provveduto a realizzare quattro activities: nella prima, MainActivity, è presente soltanto un pulsante tramite il quale si accede ai veri e propri moduli di gestione MyData; in particolare l'utente potrebbe avere o meno un account (attivo o disabilitato) di MyData per quel servizio, e per tale motivo l'utente potrebbe essere presentato con la schermata NewAccountActivity oppure UserProfileActivity. Nel primo caso la schermata consisterà in un pulsante (+) che, previa conferma, porterà a creare l'account richiesto e il relativo Service Consent, e condurrà alla seconda activity citata, UserProfileActivity, la quale al suo interno conterrà la logica effettiva di gestione dei permessi (Consent); in particolare: un elenco di switch per i dati in dettaglio da voler condividere (la cui preferenza viene mantenuta nelle Shared Preferences, e sempre da esse si attingerà alla necessità di emettere Data Consent), un pulsante per disabilitare il Service Consent (stato da cui può comunque essere riabilitato) e un pulsante per revocare il Service Consent (che equivale all'assenza di un account presso quel servizio). Per fini di testing ho deciso di includere anche un pulsante che permetta di visionare i Data Consent precedentemente scambiati per quel Service Consent.