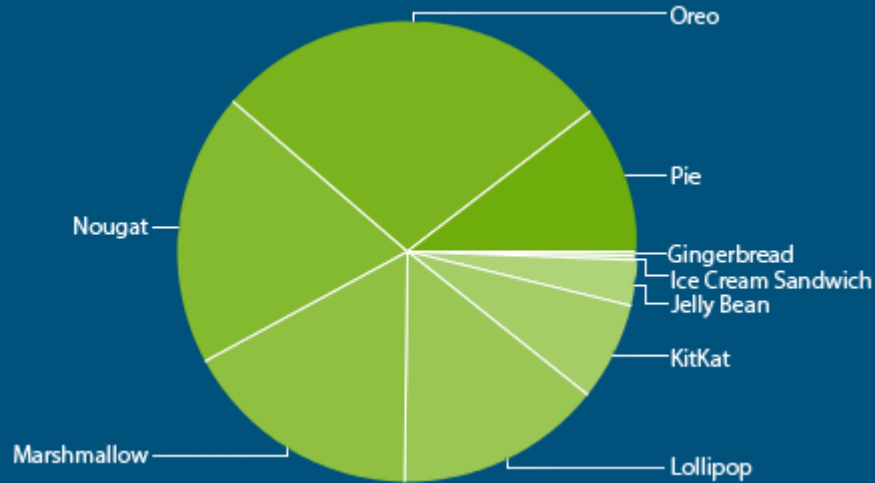


Invivo testing di applicazioni Android

Relatore: Leonardo Mariani
Co-relatore: Matteo Orrù

Problema della frammentazione di Android



Il problema della frammentazione di Android riguarda la diffusione non uniforme delle varie versioni del sistema operativo.

La disomogeneità e la numerosità delle versioni del sistema operativo ha reso difficile da parte degli sviluppatori lo sviluppo e test delle applicazioni.

In particolare, dato l'alta configurabilità del sistema operativo e delle applicazione che vi girano è quasi impossibile riuscire a testare tutte le possibili configurazioni.

Una possibile soluzione: il Framework in-vivo

Il framework in-vivo è nato con l'idea di aiutare gli sviluppatori nel rivelare e testare particolari bug che avvengono a runtime.

La fase di testing può avvenire direttamente lato client, quindi sul dispositivo dell'utente solo se è possibile eseguire un meccanismo di isolamento.

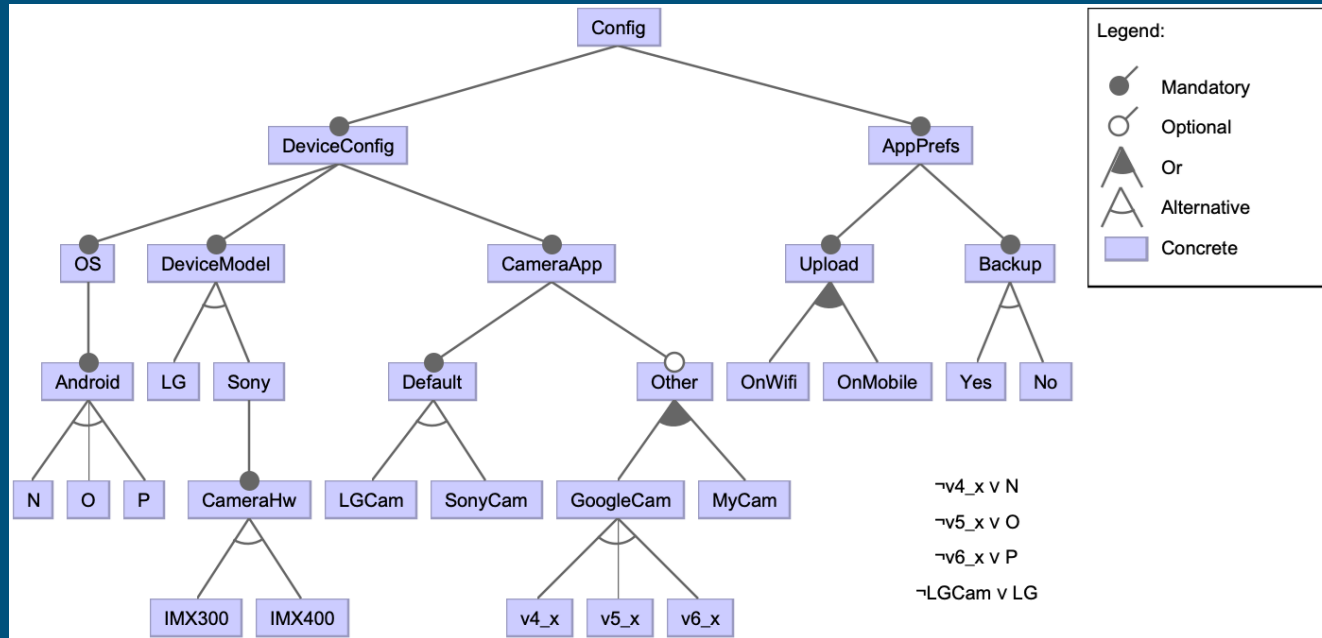
Per riuscire a rivelare i bug prima citati si è pensato all'uso delle configurazioni.

Il framework in-vivo fa parte di un progetto di ricerca, in particolare mi sono occupato di:

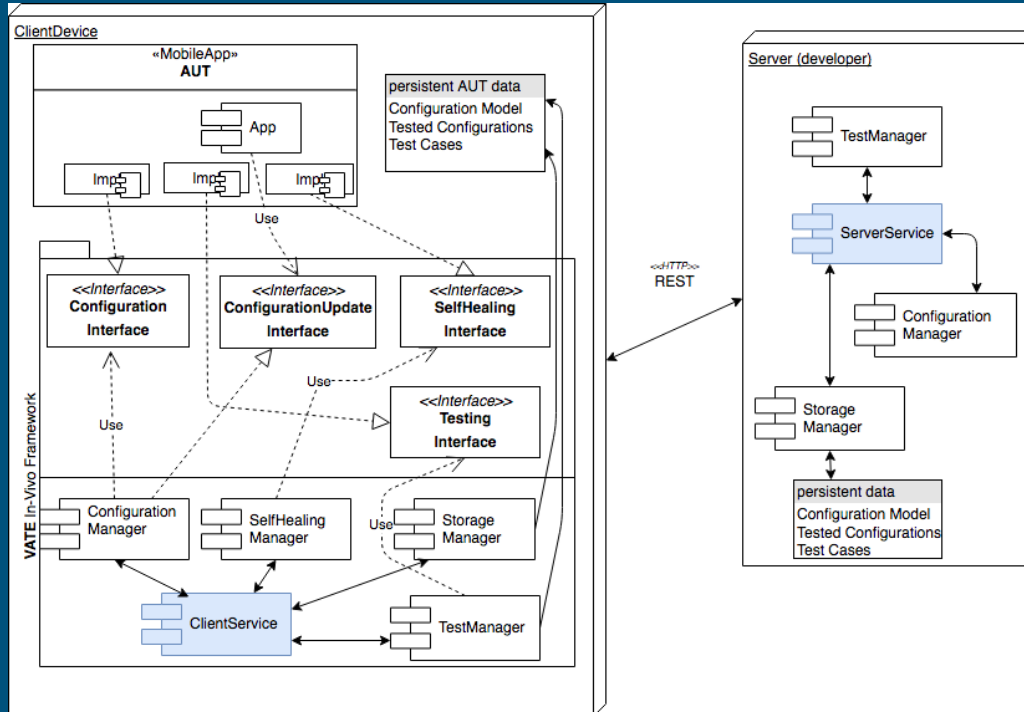
- Risolvere alcuni problemi di cui il framework era affetto e che sono emersi durante l'attività di debugging.
- Fare una prima esperienza del framework con un'applicazione reale.
- Testare il corretto funzionamento del framework.

Feature Model

Per tenere traccia di tutte le possibili configurazioni viene usato il feature model.



Struttura Framework in-vivo



Strumenti utilizzati



Git appartiene alla famiglia di software per fare versionamento del codice. In particolare il sistema adottato da git è un sistema decentralizzato.



Docker è uno strumento per la gestione e sviluppo dei container all'interno dei quali è possibile eseguire un'applicazione.

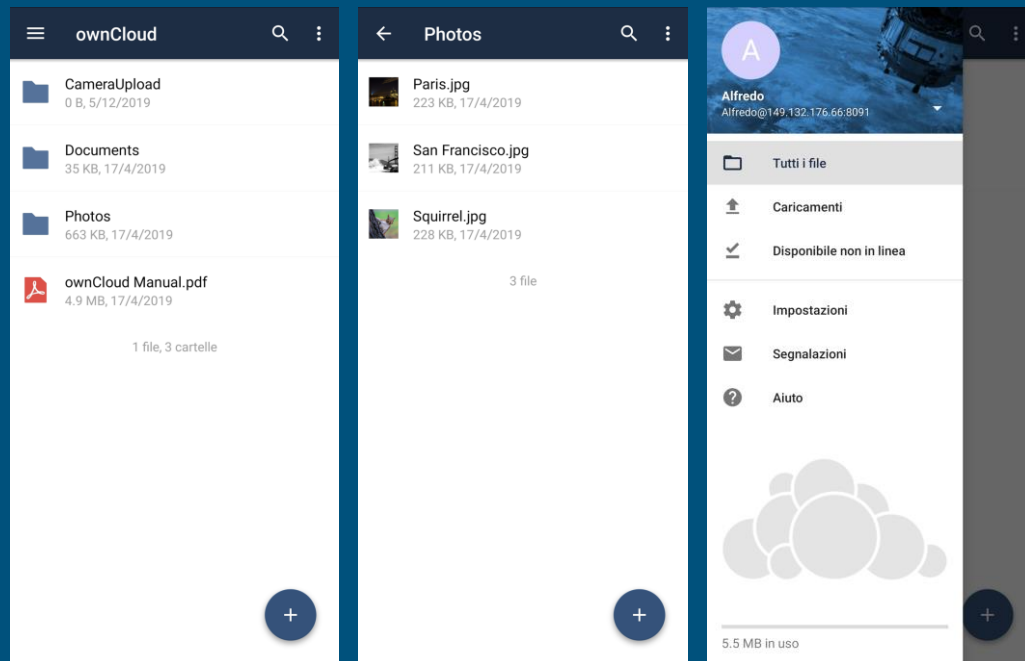


Maven è un tool per l'automazione dei processi di compilazione dei progetti, di esecuzione dei test e della creazione della documentazione.

Presentazione ownCloud



È stata scelta tale applicazione per poterla integrare con le funzionalità offerte dal framework in-vivo.

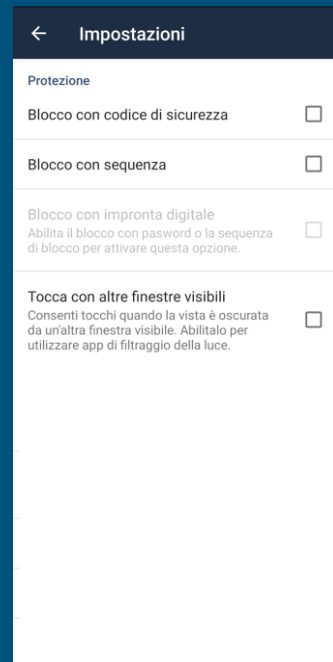
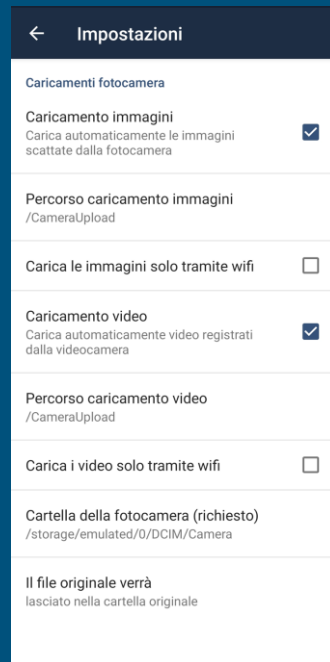


Lavoro su ownCloud

Per poter integrare ownCloud con le funzionalità del framework in-vivo c'era bisogno di recuperare le configurazioni dell'applicazione.

Queste vengono recuperate una volta che l'utente entra e successivamente esce dalle impostazioni.

Il recupero delle informazioni relative alle scelte fatte nella impostazioni è stata possibile grazie alle SharedPreferences.



Lavoro svolto sull'in-vivo server

L'in-vivo server ha il compito di processare le configurazioni che gli vengono inviate e classificarle.

Le configurazioni devono avere una precisa forma per poter venire correttamente processate dall'in-vivo server. La forma in questione è la seguente:

fname1=fvalue1,fvalue2;fname2=fvalue1; ...; fname-n=fvalue-n;

Tuttavia nella classificazione questo presentava due criticità che non gli permettevano di classificare in modo esatto le configurazioni.

Queste sono:

- Configurazione classificata come untested quando tested.
- Configurazione unknown non sempre riconosciuta.

Simulazione degli scenari

Per testare il corretto funzionamento del framework in-vivo dopo aver integrato ownCloud con le sue funzionalità sono stati simulati 4 scenari.

```
preferences_pattern_lock=preferences_pattern_lock_false;
DeviceModel=Galaxy S10e;
preferences_video_uploads=preferences_video_uploads_false;
preferences_pincode_lock=preferences_pincode_lock_false;
preferences_picture_uploads=preferences_picture_uploads_false;
Android=9;
```

1. Generazione della prima configurazione.

```
preferences_pattern_lock=preferences_pattern_lock_false;
DeviceModel=Galaxy S10e;
preferences_video_uploads=preferences_video_uploads_false;
preferences_pincode_lock=preferences_pincode_lock_false;
preferences_picture_uploads=preferences_picture_uploads_false;
Android=9;
```

3. Generazione di una configurazione uguale alla prima.

2. Generazione di una configurazione diversa dalla precedente.

```
preferences_pattern_lock=preferences_pattern_lock_false;
DeviceModel=Galaxy S10e;
preferences_video_uploads=preferences_video_uploads_false;
preferences_pincode_lock=preferences_pincode_lock_false;
preferences_picture_uploads_via_wifi_only=preferences_picture_uploads_via_wifi_only_false;
preferences_picture_uploads=preferences_picture_uploads_true;
Android=9;
```

4. Generazione di una configurazione unknown.

```
preferences_pattern_lock=preferences_pattern_lock_false;
DeviceModel=Xiaomi Mi9;
preferences_video_uploads=preferences_video_uploads_false;
preferences_pincode_lock=preferences_pincode_lock_false;
preferences_picture_uploads=preferences_picture_uploads_false;
Android=9;
```

Conclusioni

L'in-vivo framework rappresenta una valida opzione per supportare gli sviluppatori nel classificare e testare le configurazioni che vengono scoperte.

Tramite appunto la loro classificazione si riesce capire se un bug avvenuto a runtime può dipendere dal cambiamento di specifiche impostazioni da parte dell'utente.

L'integrazione di ownCloud con le funzionalità del framework è un esempio di quello che gli sviluppatori devono fare per poter usufruire dei vantaggi del framework.

Grazie per l'attenzione
