

**UNIVERSITATEA POLITEHNICA BUCURESTI,
FACULTATEA DE AUTOMATICA SI CALCULATOARE**

Aplicatie colaborativa bazata pe contacte oportuniste

- Arhitectura aplicatiei -

Branescu Adrian

Grupa 341C3

1. Introducere

Proiectul propune dezvoltarea unei aplicatii Android bazata exclusiv pe comunicatie descentralizata in medii wireless. Informatiile trimise din aceasta aplicatie nu vor trece printr-un server, ci va fi utilizat transferul peer-to-peer. Concret, am extins functionalitatea framework-ului AllJoyn cu functiile unei aplicatii de chat (la nivelul a 2 utilizatori).

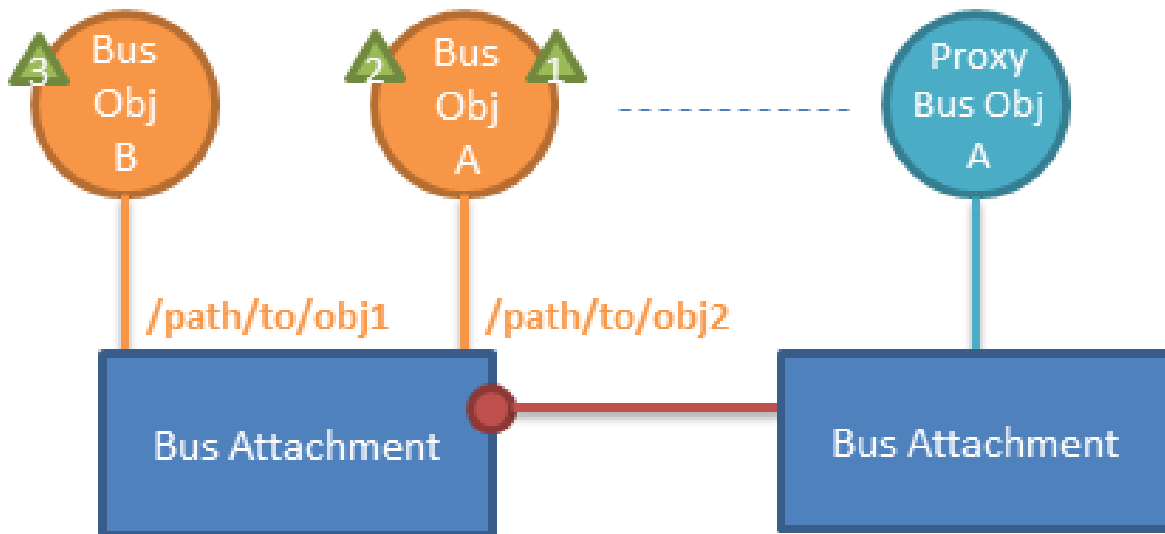
2. AllJoyn Framework

La baza framework-ului AllJoyn stau doua notiuni: aplicatie si router. Comunicatia intre aplicatii se realizeaza prin intermediul router-ului, aceasta fiind componenta care asigura conectivitatea intre dispozitive.

Arhitectura AllJoyn poate fi privita ca o magistrala (software) distribuita, comunicatia intre aplicatii facandu-se prin aceasta magistrala. Modul in care aplicatiile se conecteaza la magistrala este ad-hoc, iar interactiunea cu alte aplicatii are ca prim pas descoperirea dispozitivelor din apropiere conectate la aceeasi retea wireless (prin ip multicast).

Biblioteca AllJoyn Core pune la dispozitia dezvoltatorilor un mecanism similar RMI, serviciu prin care se creeaza obiecte proxy pentru a putea invoca metode ale obiectelor ce se afla la distanta. Acest mecanism este alcatuit din interfetele asociate

serviciilor expuse de fiecare aplicatie si din magistrala distribuita care asigura invocarea la distanta a metodelor obiectelor atasate la aceasta.



3. Arhitectura aplicatiei

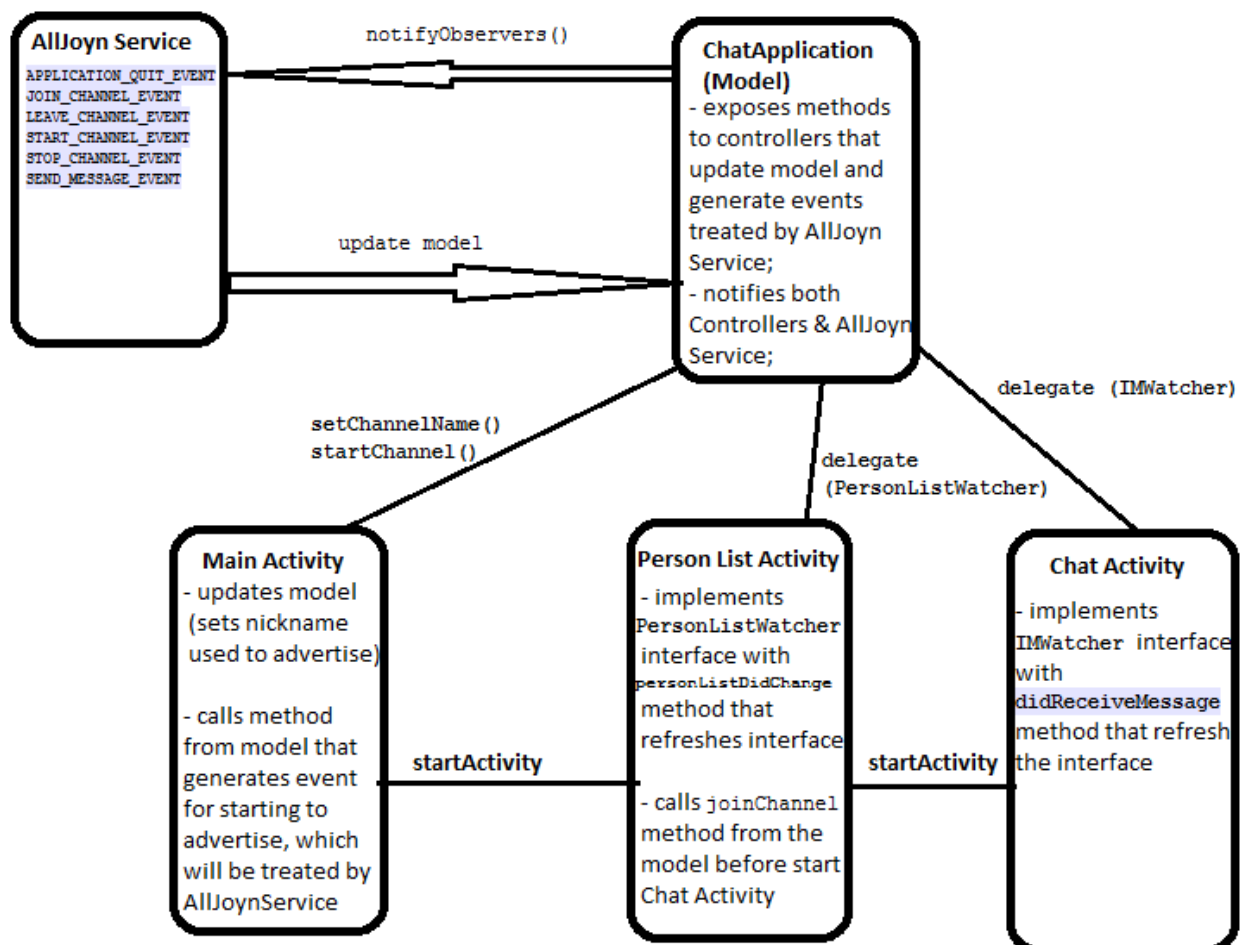
Aplicatia urmeaza design-ul Model-View-Controller (MVC) si are urmatoarele componente de baza:

- **Model:**
 - **ChatApplication** - subclasa a clasei *Application* din Android – componenta a carei durata de viata coincide cu durata de viata a aplicatiei, folosita pentru stocarea persistenta pe durata aplicatiei a conexiunilor AllJoyn, a numelui utilizatorului, a mesajelor etc. (deoarece Activity nu e persistent – e recreat la fiecare -tranzitie background-foreground a aplicatiei)

- **AllJoynService** – subclasa a clasei *Service* din Android – componenta care asigura conectivitatea prin AllJoyn intre aplicatii – este pornit la lansarea aplicatiei, se inregistreaza ca observer pentru ChatApplication pentru a trata evenimentele acestuia: descoperire peers, creare sesiune, advertise de sesiune, join de sesiune, leave de sesiune , transmitere mesaje etc.
- **View-Controller:**
 - **MainActivity** - setare nickname folosit in cadrul wellknown-name-ului (prefix aplicatie + nickname) pentru advertise si lansarea de catre tap gesture-ul utilizatorului a doua evenimente ce vor fi tratate de AllJoynService: creare sesiune, descoperire peers.
 - **PersonListActivity** – populare lista cu peers, inregistrare ca delegat pentru ChatApplication pentru a fi anuntat cand se modifica lista de peers descoperiti si lansarea de catre tap

gesture-ul utilizatorului a unui eveniment ce va fi tratat de AllJoynService: join de sesiune.

- **ChatActivity** – activitatea in care se vor primi, trimite mesaje – inregistrare ca delegat pentru ChatApplication pentru a fi anuntat cand se modifica lista de mesaje si lansarea unui eveniment ce va fi tratat de AllJoynService: trimiterea de mesaj in cadrul sesiunii.



Ca design pattern-uri folosite pentru modularizarea si reutilizarea codului amintim: MVC, Observer si Delegate.

MVC reprezinta chiar scheletul aplicatiei, asa cum am mentionat mai sus.

Observer este folosit de catre serviciul ce asigura operatiile realizate prin AllJoyn pentru a trata evenimentele generate de aplicatie.

Delegate este folosit de catre view-controllere pentru a actualiza elementele de interfata (view-urile) atunci cand se modifica modelul.