Android fortsättning

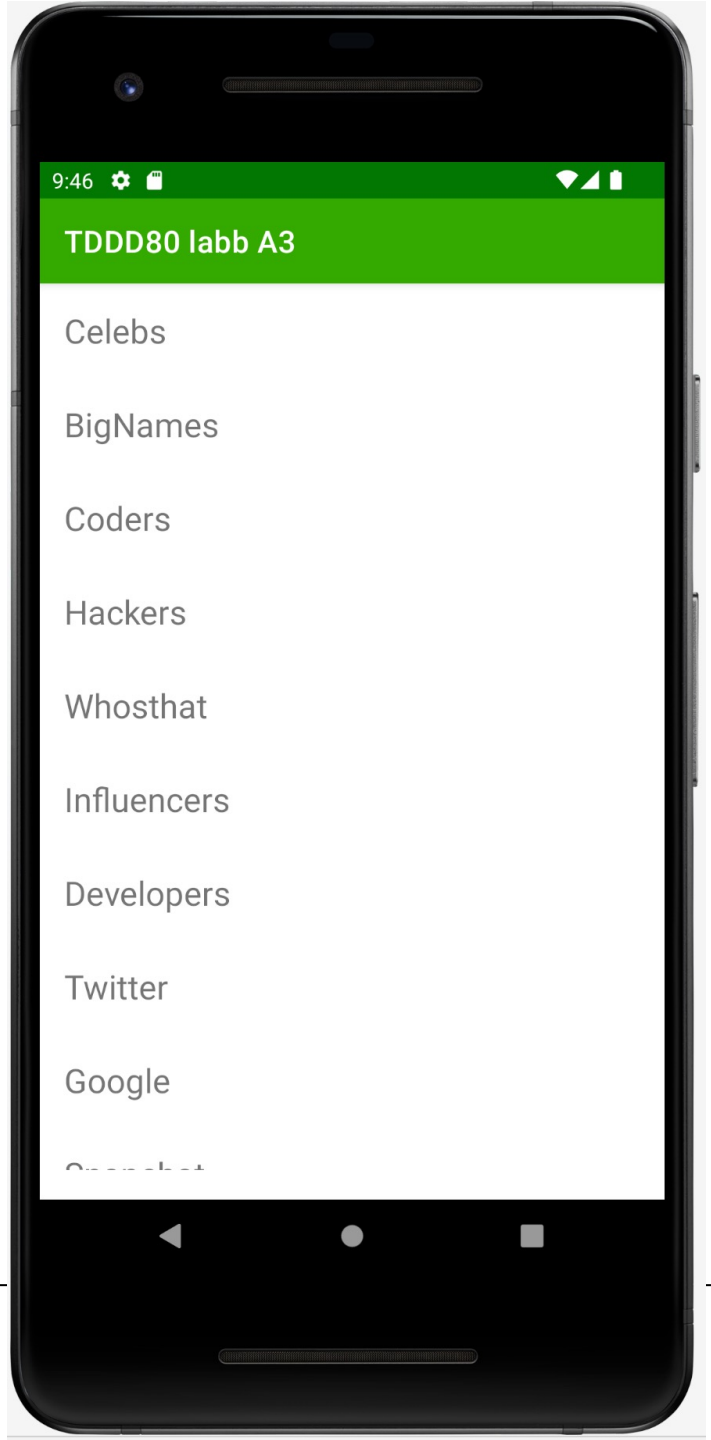
TDDD80 Mobila och sociala applikationer

Krav labb A2 och A3

- Labb A2: Scrollbar lista + detalj-vy
  - Navigation
  - Safe Args
  - Binding
- Labb A3: Nätverksanrop
  - Hämta data från server
  - Hantera JSON

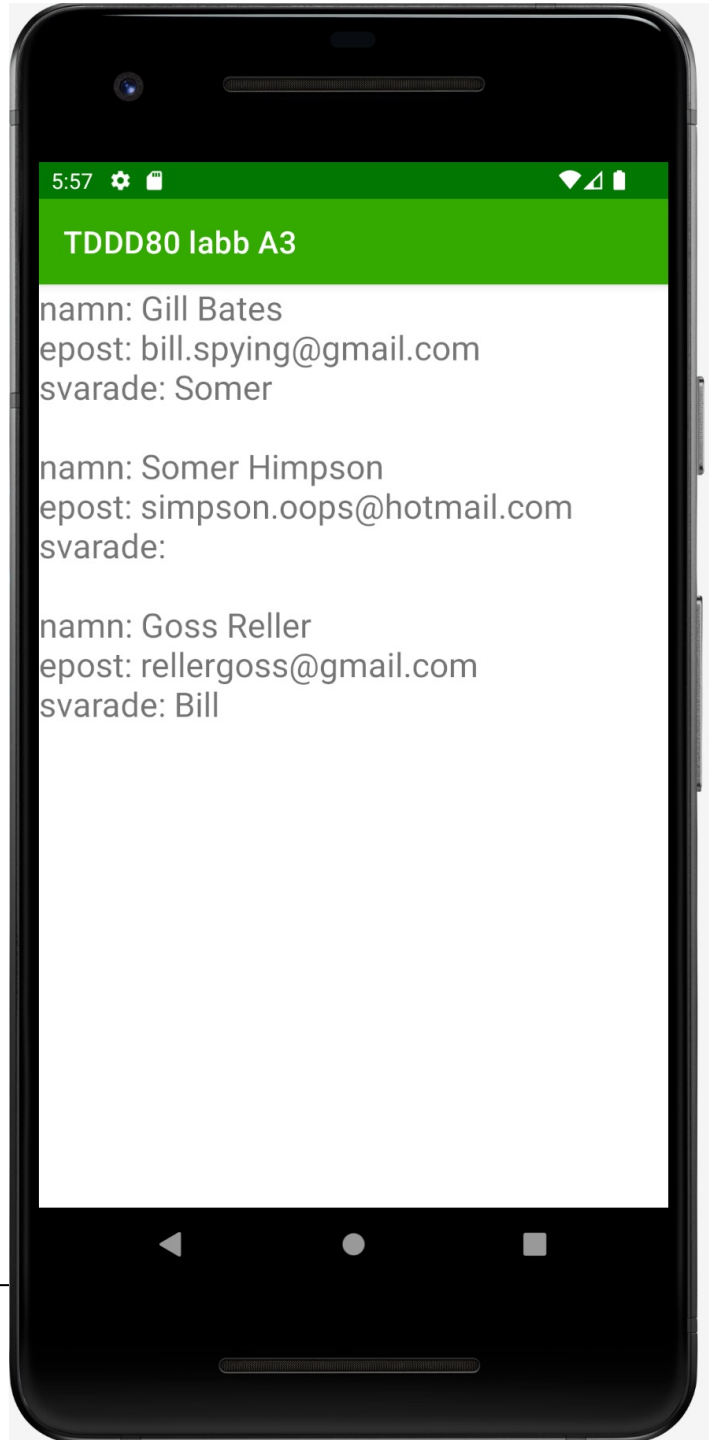
Labb A2

- Scrollbar lista



Labb A2

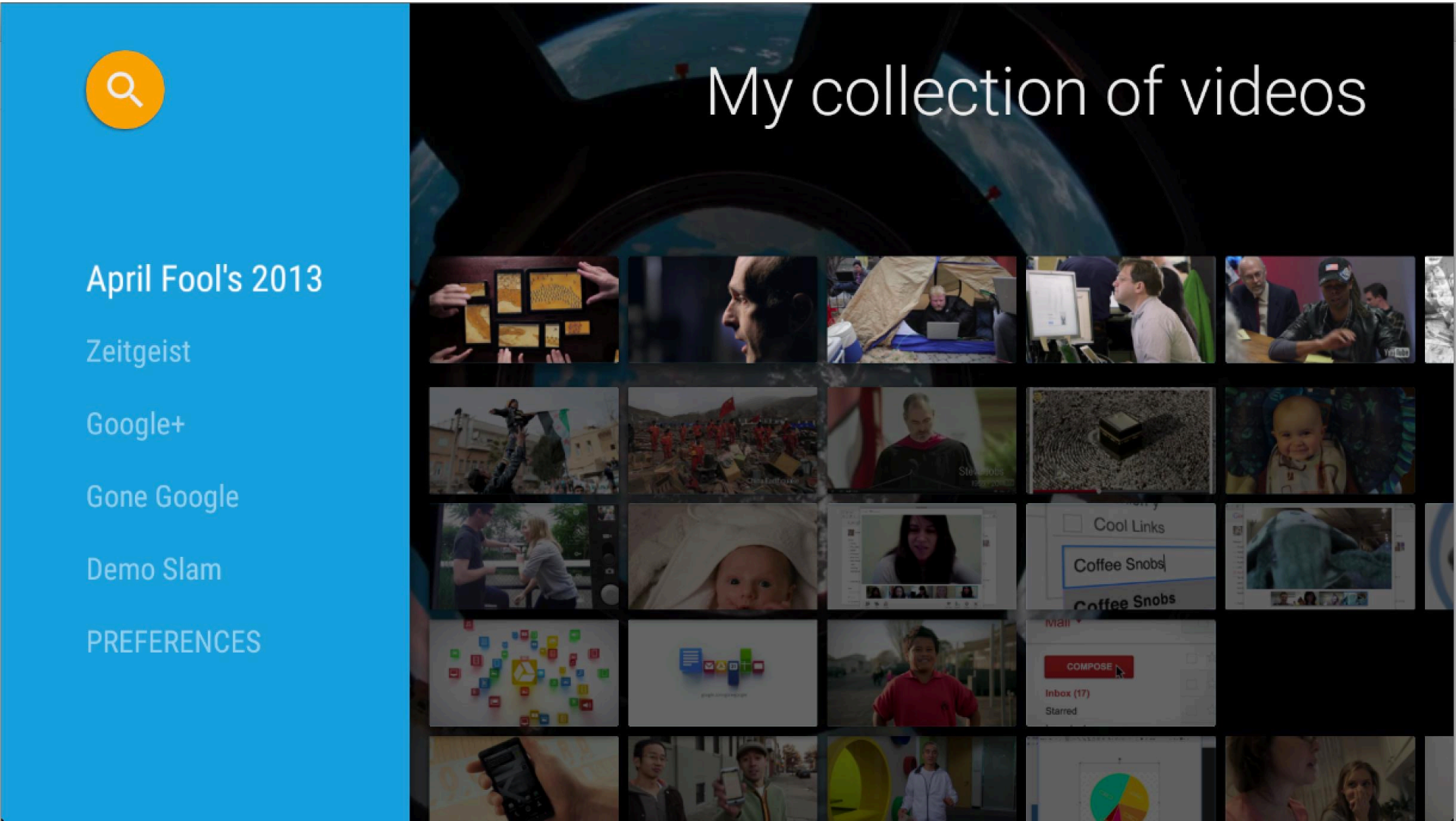
- Efter val: detalj-vy



Ska vi ha två Activities?

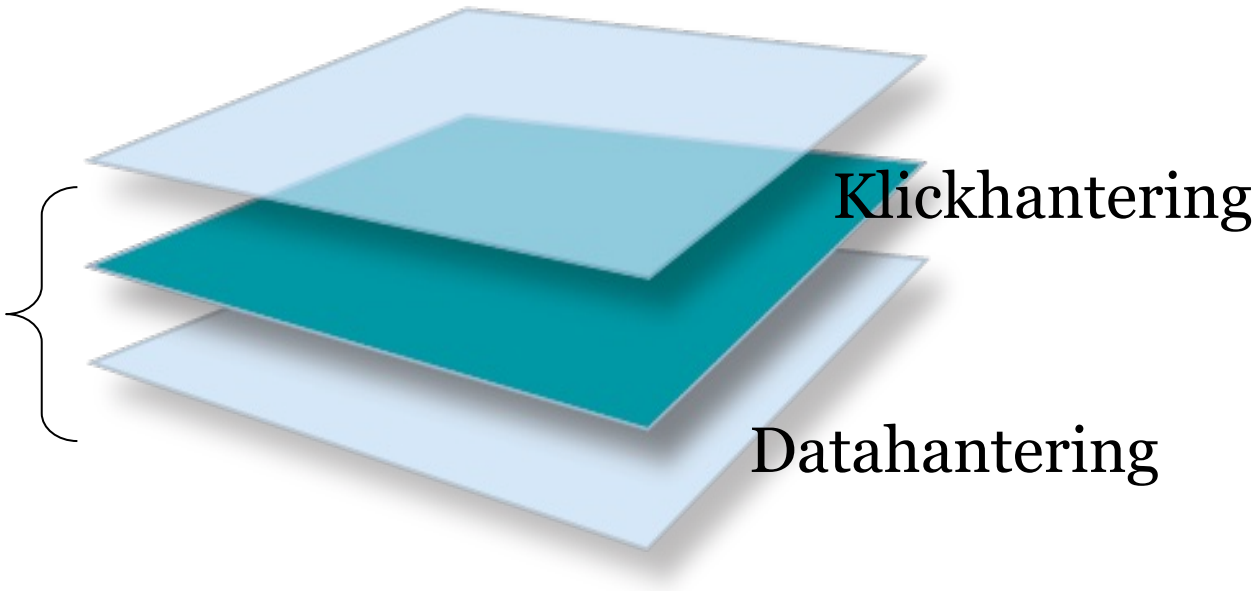
- Nackdelar:
  - Activities är speciella:
    - Har direktkontakt med Android
    - Kan inte läggas två på samma skärm
    - Kan inte nästlas (läggas inne i varandra)

Vill ha flyttbara, återanvändbara skärmbitar

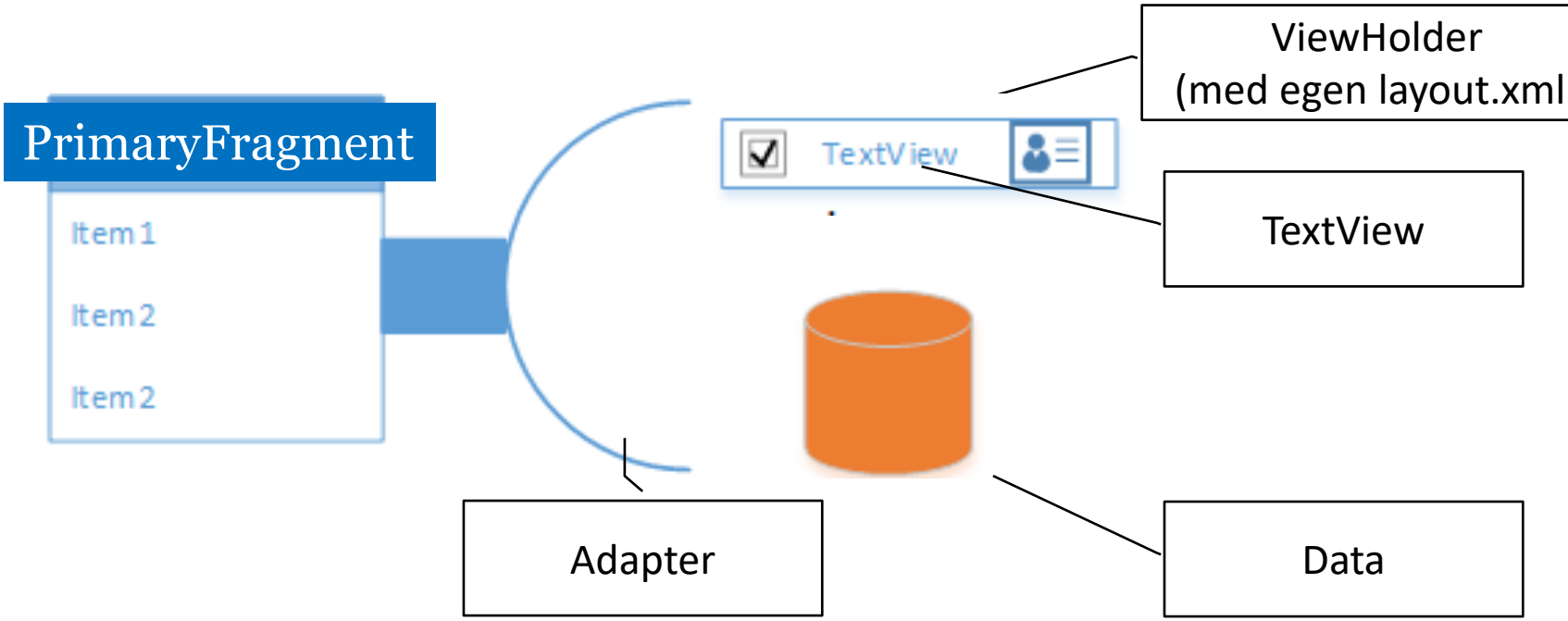


Återanvändbara skärmbitar: Fragment

- Återanvända både
  - Layout
  - Beteende

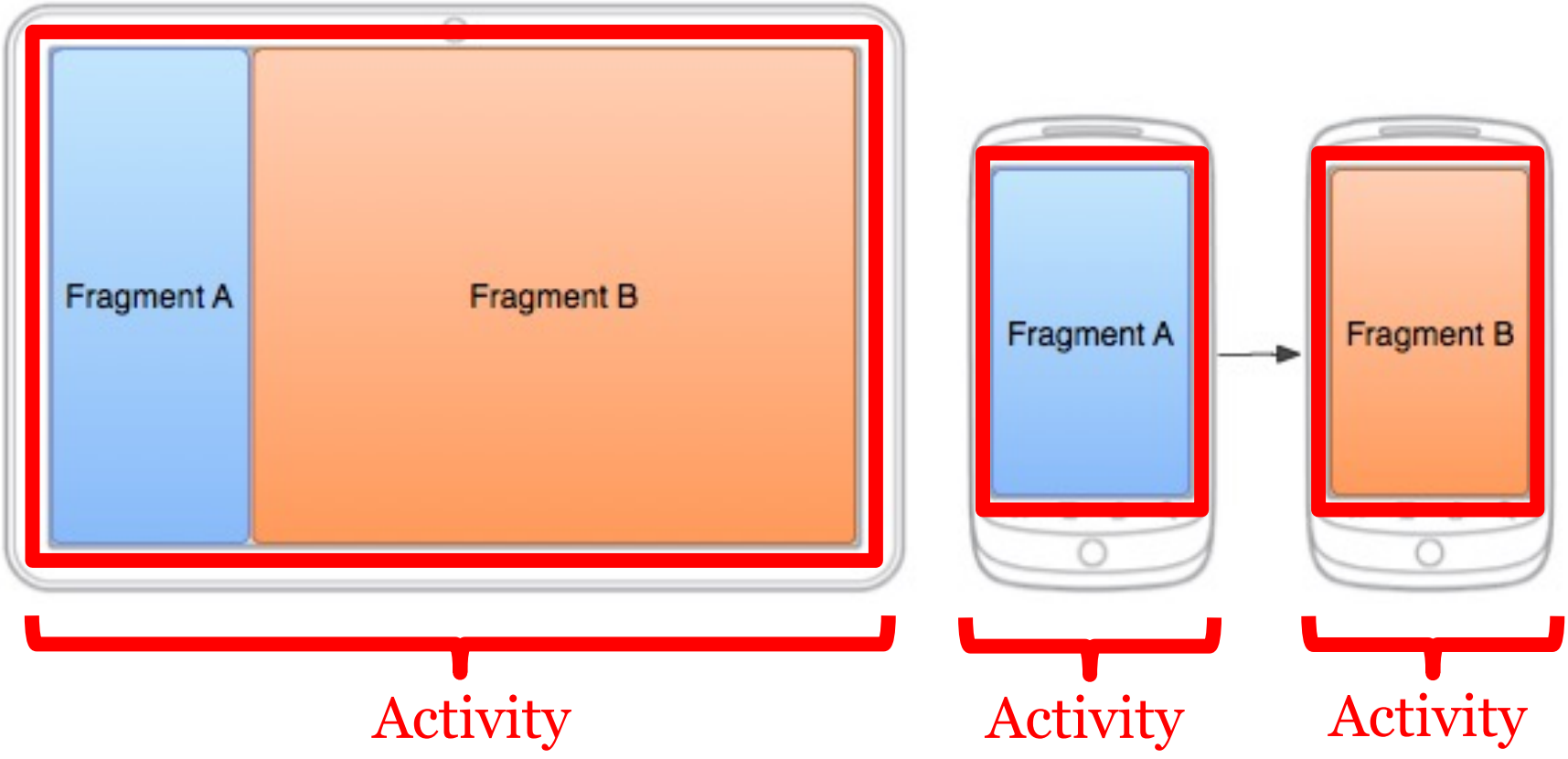


Adapter: matar innehåll när listan scrollas



RecyclerView.Adapter återanvänder ViewHolders och TextViews:
Lägger in ny text i gamla Views

En Activity, flera utbytbara fragment



MainActivity

```
public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

    public void onListItemClick(View v, int position) {
        ...
```

Exempel: DetailFragment

```
public class DetailFragment extends Fragment {

    private FragmentDetailBinding binding;
    private int group_idx;

    public DetailFragment() {}

    @Override
    public void onCreate(…) {
        super.onCreate(…);
        group_idx = getArguments().getInt("group_idx");
    }
```

Namn på xml-fil:
fragment_detail.xml

Packa upp argument

Exempel: DetailFragment

```
@Override
public View onCreateView(@NonNull LayoutInflater inflater,
                         ViewGroup container,
                         Bundle savedInstanceState) {
    binding = FragmentDetailBinding.inflate(inflater,
                                            container,
                                            false);
    return binding.getRoot();
}
```

Bygg layout

Utför binding

Returnera rot-vyn

Exempel: PrimaryFragment (listan)

```
@Override
public void onViewCreated(…) {
    super.onViewCreated(…);
    binding.list.setAdapter(new MyRecyclerAdapter());
}
```

Är redan uppe på skärmen, kan komma åt delkomponenter (t.ex. listan)

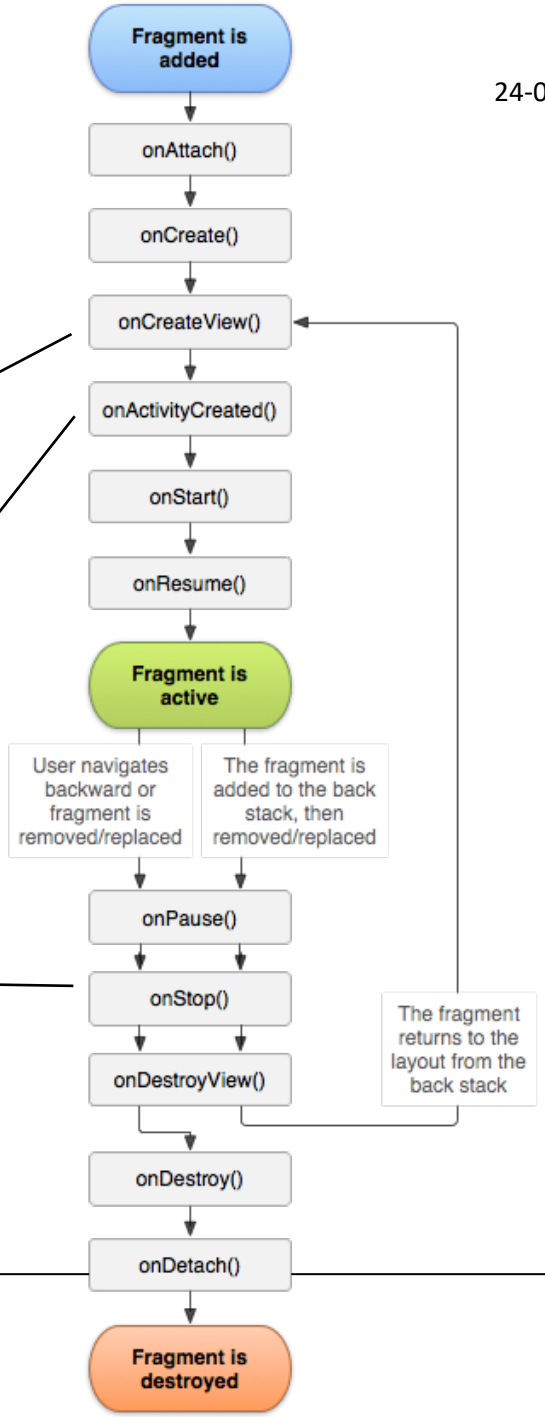
Sätt adapter

Fragmentets lifecycle

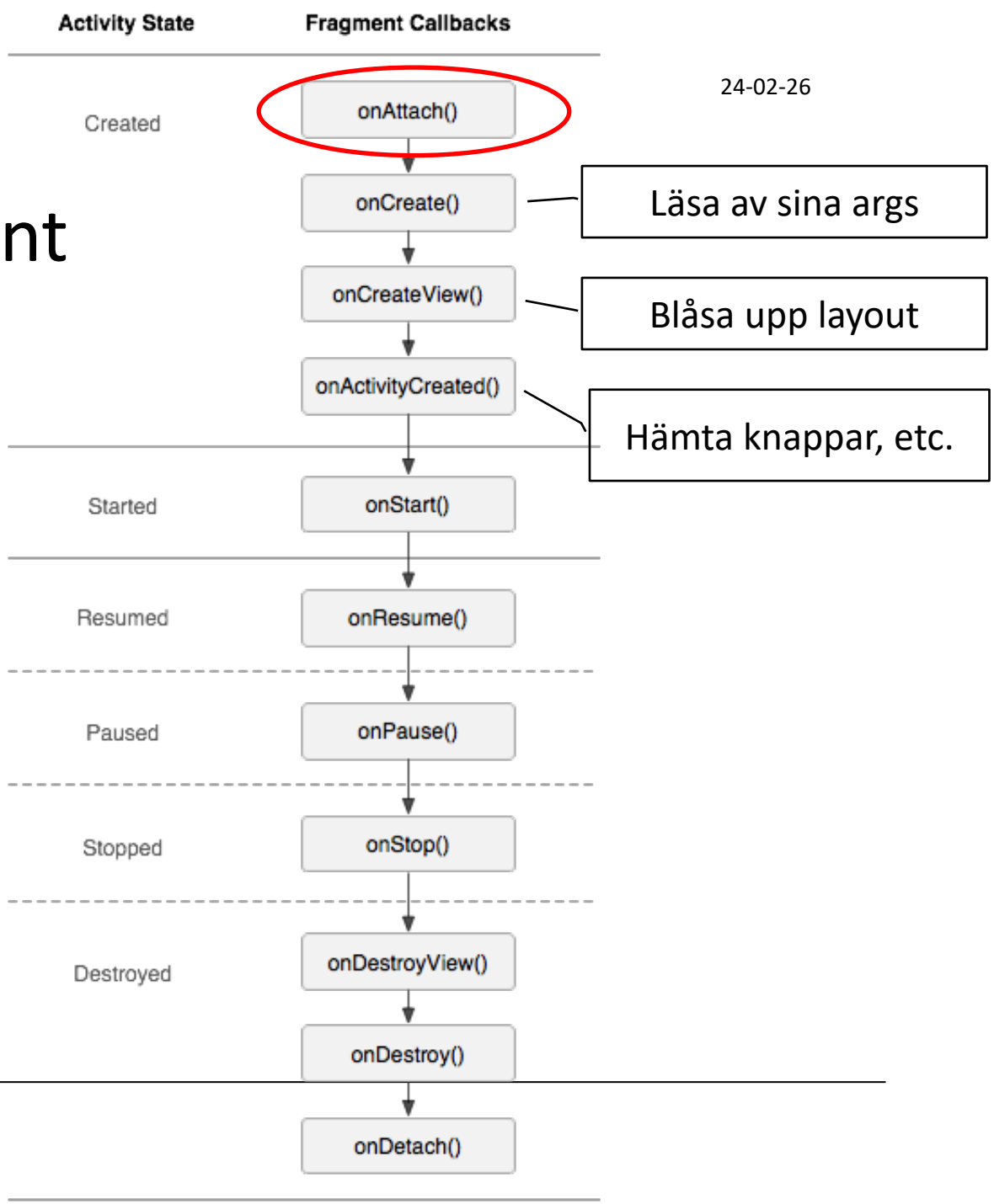
Fyll med innehåll

Är uppe på skärmen. Det går nu att komma åt komponenter, knappar, etc.

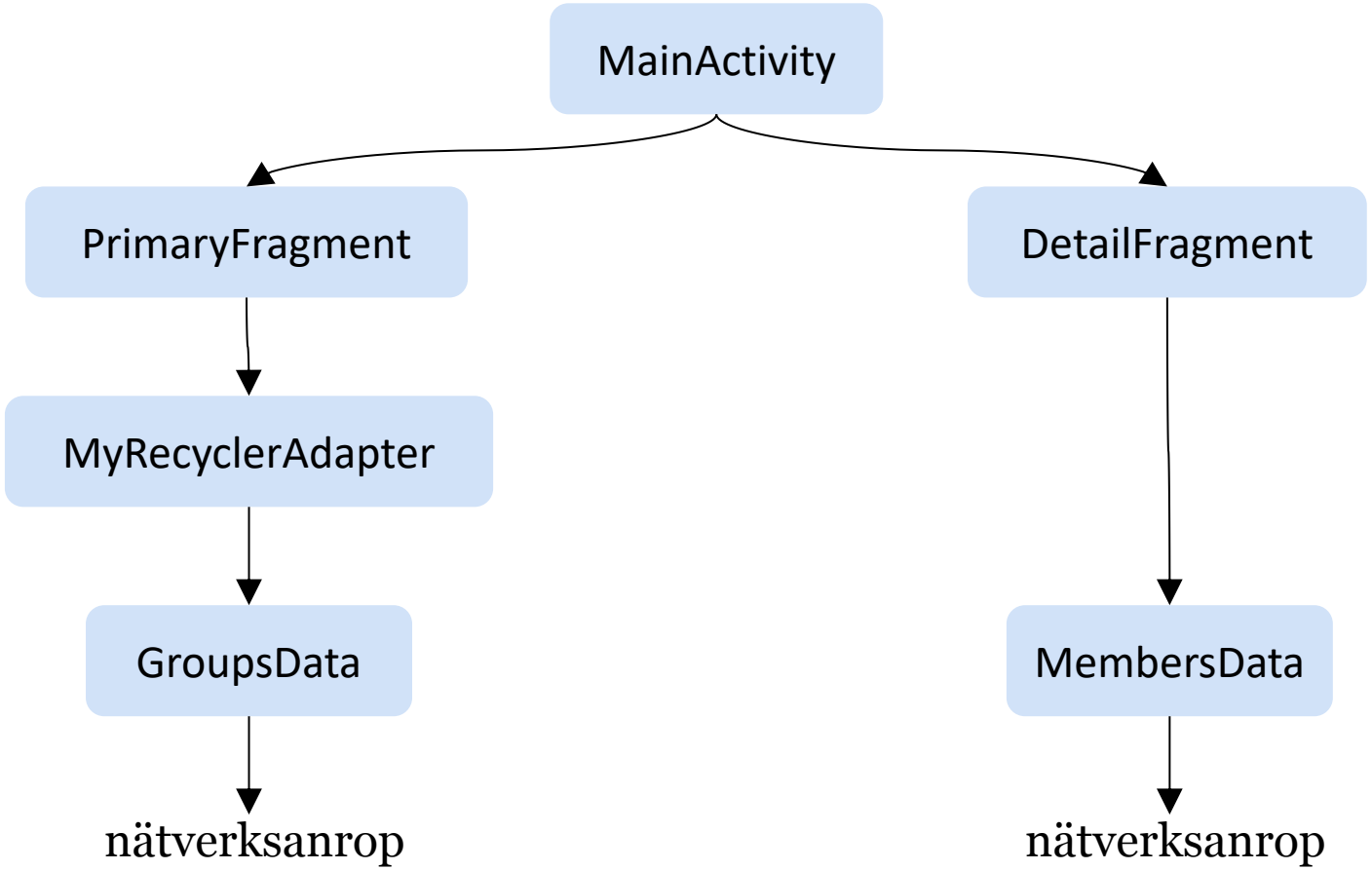
rotation



Activity - Fragment



App-struktur



Arbetsfördelning: Fragments

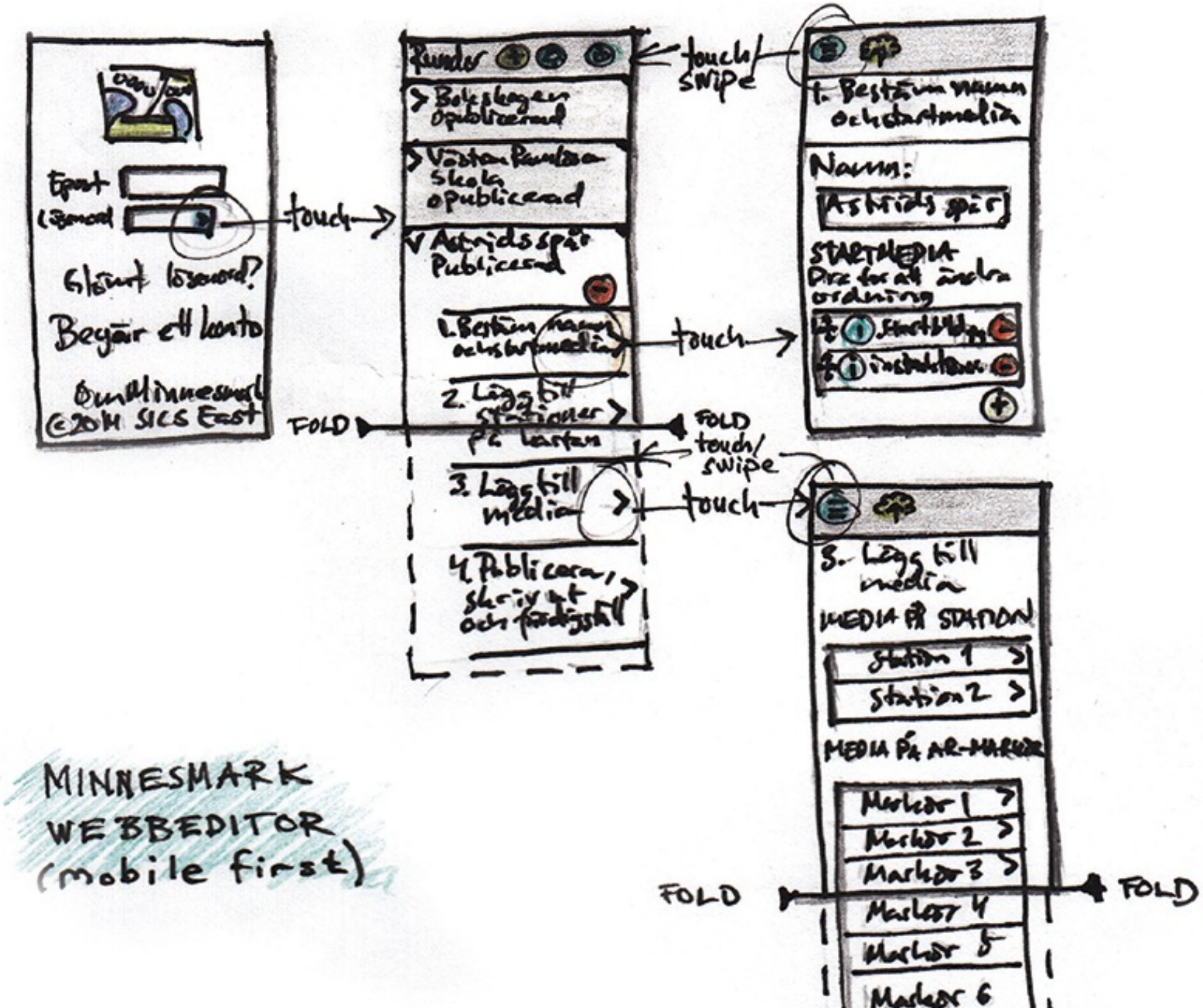
- Visar innehåll
- Tar emot klickar, skickar upp till Activity
- Synkar sina data mot server (nätverksanrop)

- Ska inte veta om vilka andra fragment som finns på skärmen just nu
  - Eftersom detta kan variera

Arbetsfördelning: MainActivity

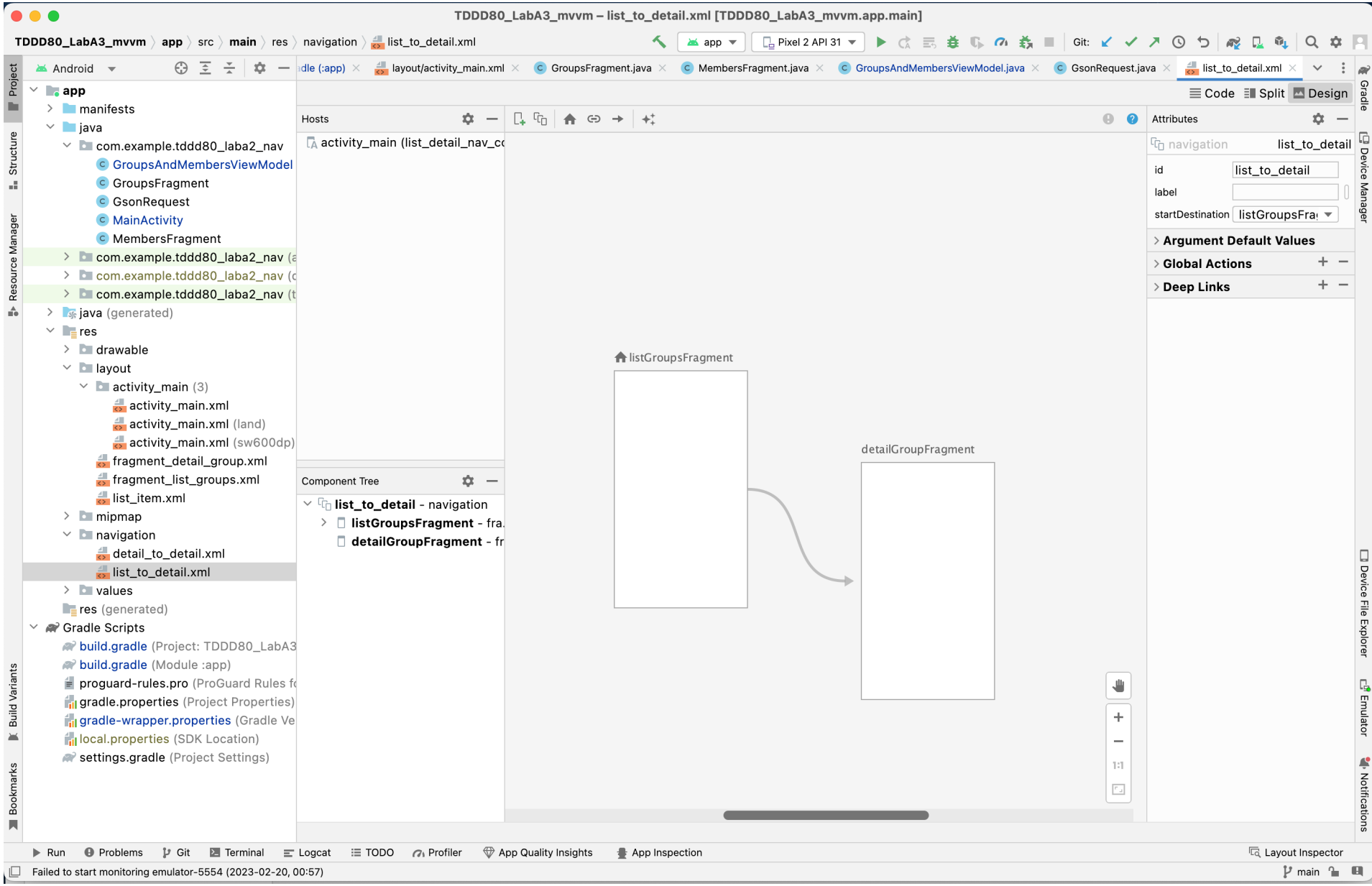
- Direktkontakt med Android OS
  - Kan anropas “utifrån” (från andra appar)
- Hanterar klick: Navigerar mellan fragment

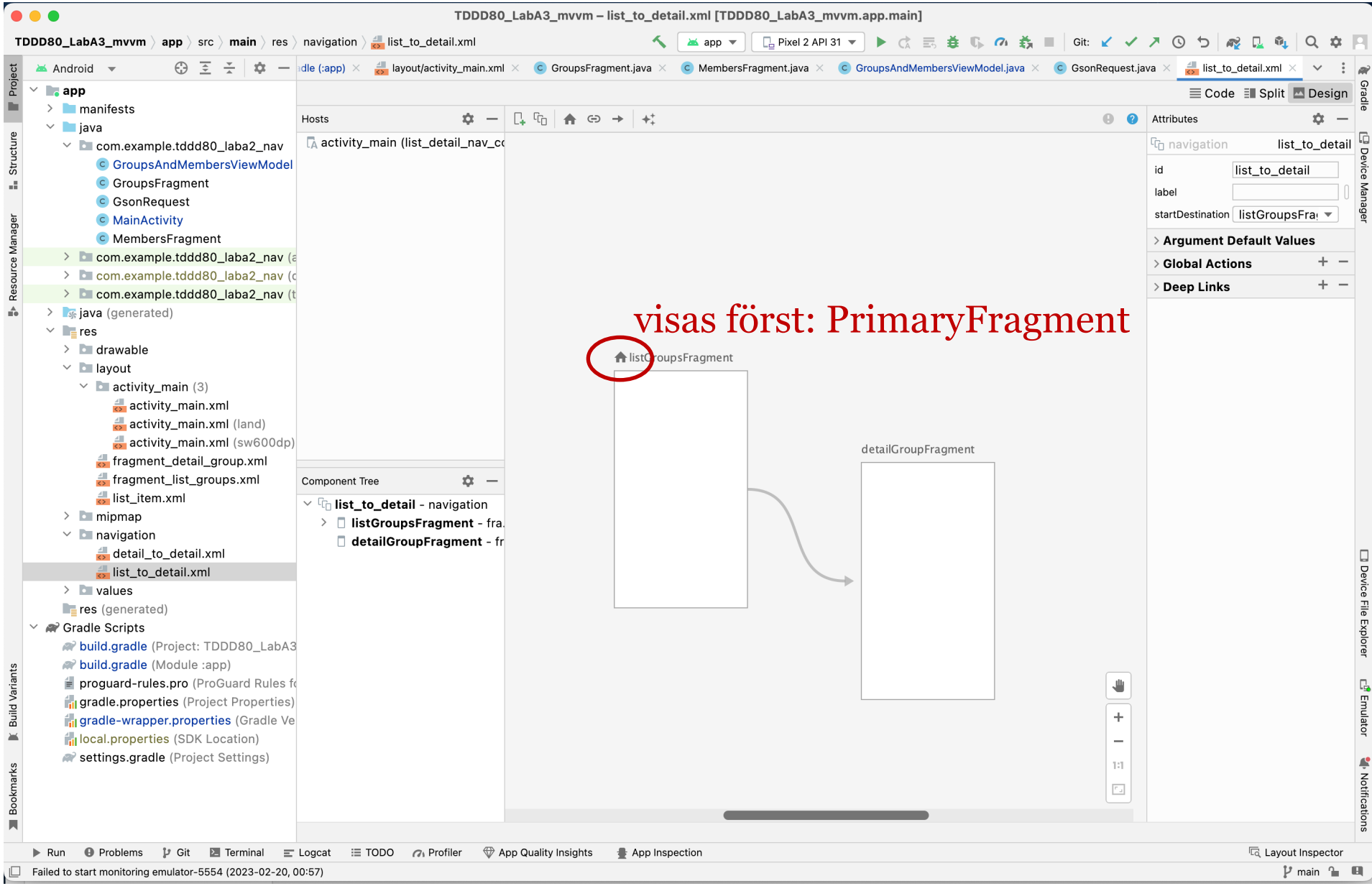
Navigering

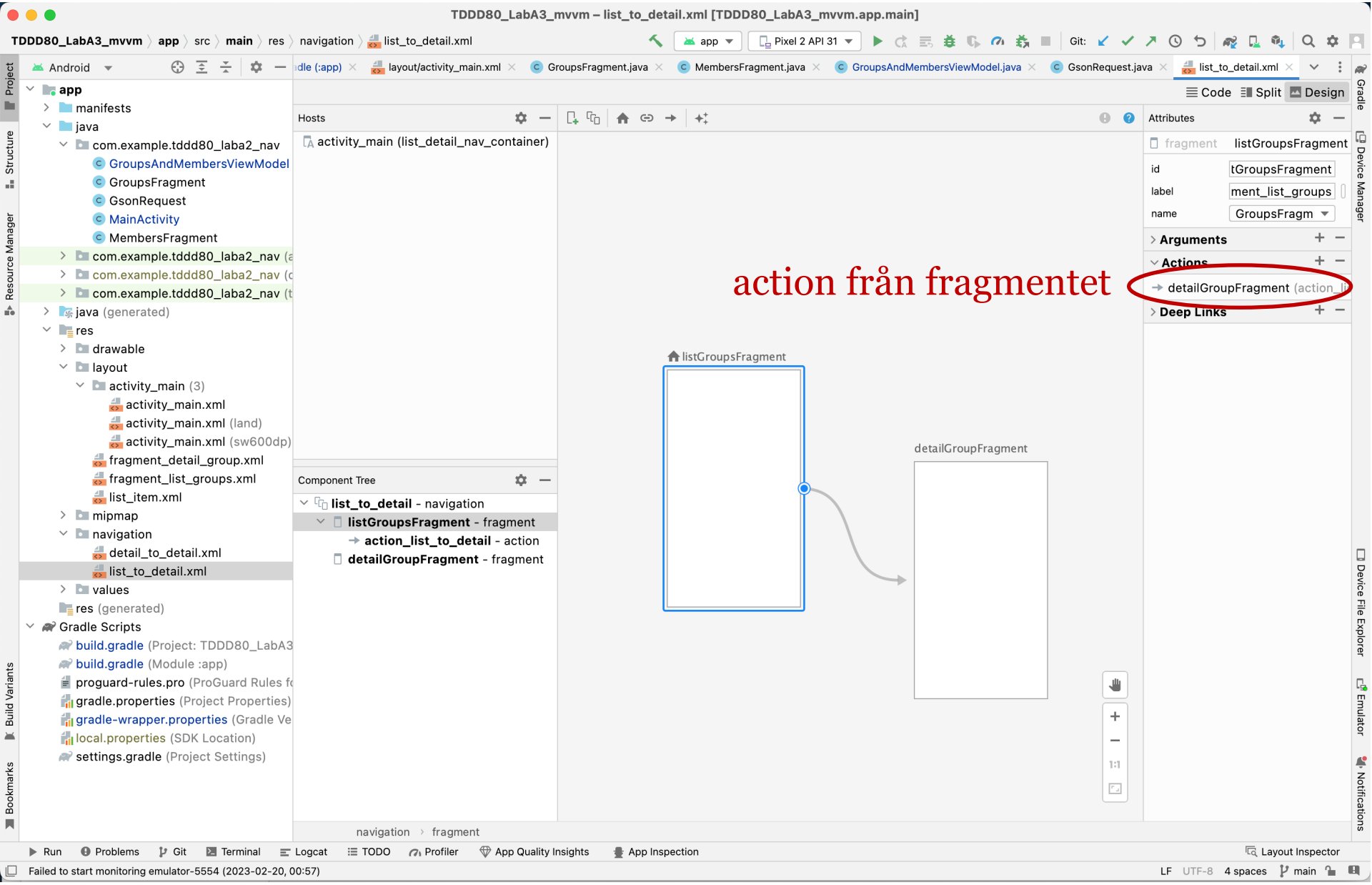

touch
touch/
swipe
touch
FOLD
FOLD
touch/
swipe
touch
FOLD
FOLD
MINNESMARK
WEBBEDITOR
(mobile first)

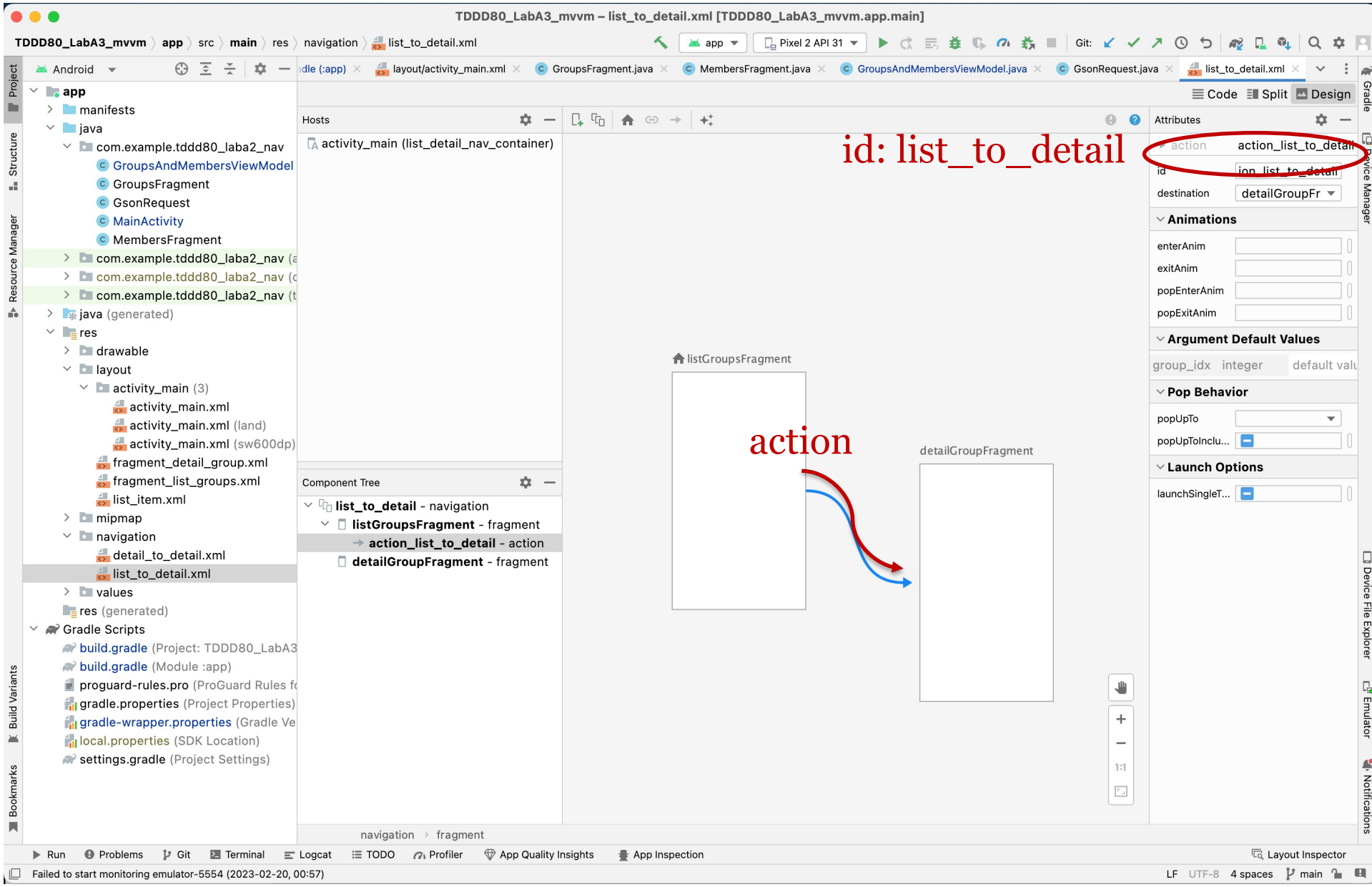
Utgå ifrån navigeringsskiss

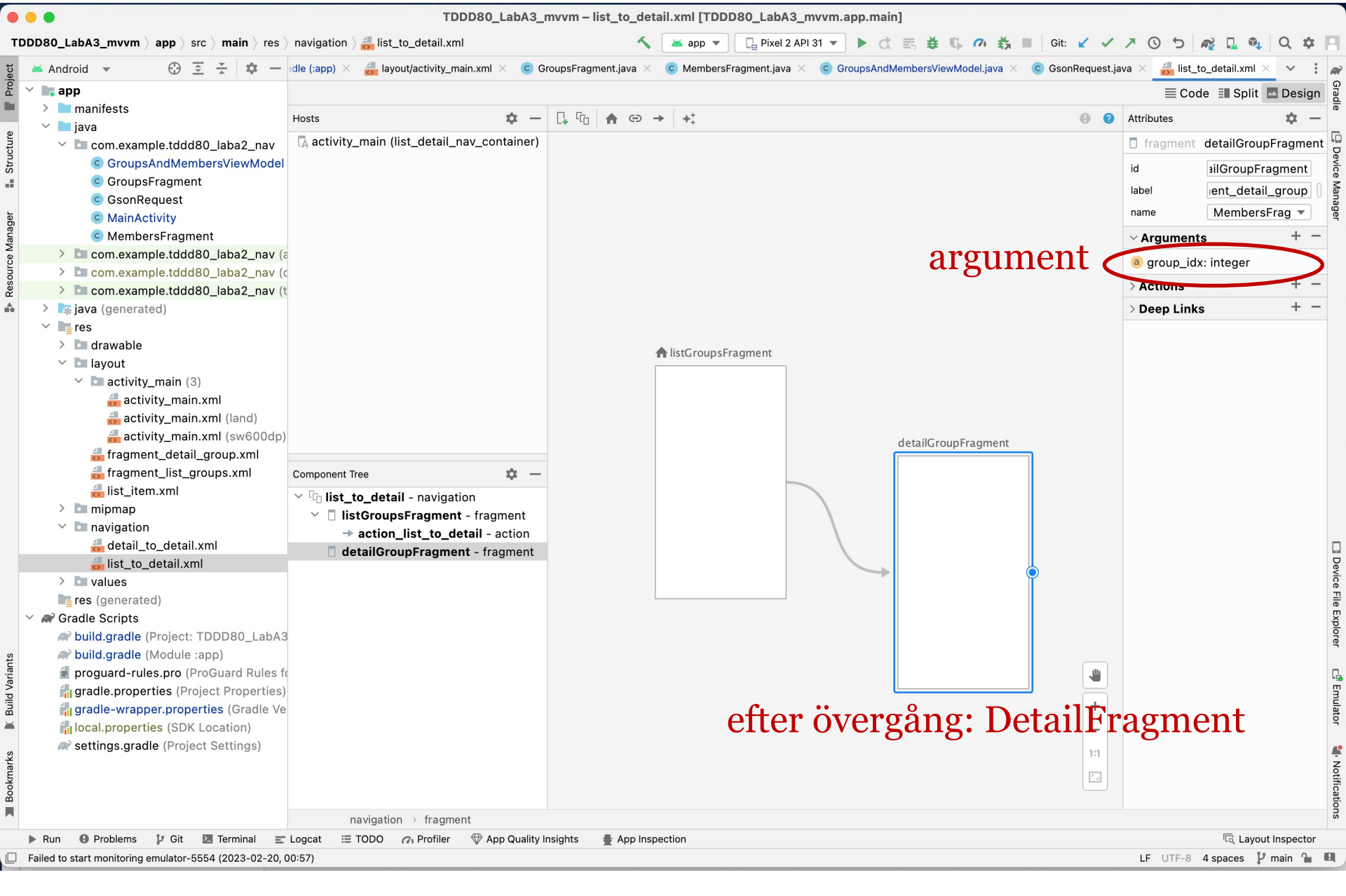
- Inkludera bibliotek för navigering i build.gradle (app)
- Skapa ny navigation graph
  - Högerklicka på res-mappen -> New Resource File -> Navigation Graph
- Skapa destinations och övergångar (actions)
  - Övergång från PrimaryFragment till DetailFragment

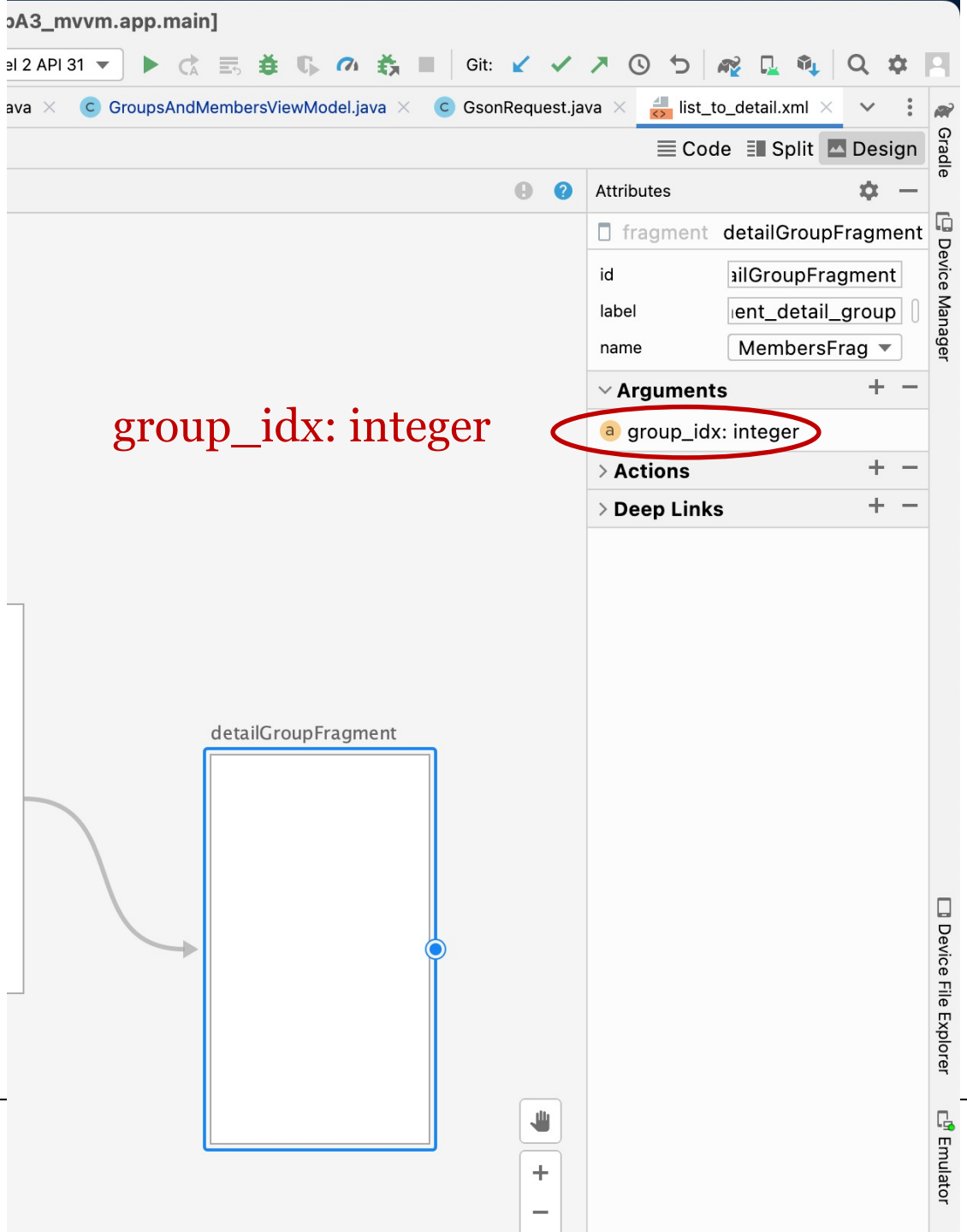

TDDD80_LabA3_mvvm – list_to_detail.xml [TDDD80_LabA3_mvvm.app.main]
TDDD80_LabA3_mvvm 〉 app 〉 src 〉 main 〉 res 〉 navigation 〉 list_to_detail.xml
app
Pixel 2 API 31
Git:
Android
app
manifests
java
com.example.tddd80_laba2_nav
GroupsAndMembersViewModel
GroupsFragment
GsonRequest
MainActivity
MembersFragment
com.example.tddd80_laba2_nav (a
com.example.tddd80_laba2_nav (c
com.example.tddd80_laba2_nav (t
java (generated)
res
drawable
layout
activity_main (3)
activity_main.xml
activity_main.xml (land)
activity_main.xml (sw600dp)
fragment_detail_group.xml
fragment_list_groups.xml
list_item.xml
mipmap
navigation
detail_to_detail.xml
list_to_detail.xml
values
res (generated)
Gradle Scripts
build.gradle (Project: TDDD80_LabA3
build.gradle (Module :app)
proguard-rules.pro (ProGuard Rules f
gradle.properties (Project Properties)
gradle-wrapper.properties (Gradle Ve
local.properties (SDK Location)
settings.gradle (Project Settings)
dle (:app)
layout/activity_main.xml
GroupsFragment.java
MembersFragment.java
GroupsAndMembersViewModel.java
GsonRequest.java
list_to_detail.xml
Code
Split
Design
Hosts
activity_main (list_detail_nav_c
Component Tree
list_to_detail - navigation
listGroupsFragment - fra
detailGroupFragment - fr
listGroupsFragment
detailGroupFragment
Attributes
navigation
list_to_detail
id
list_to_detail
label
startDestination
listGroupsFra
Argument Default Values
Global Actions
Deep Links
Project
Structure
Resource Manager
Build Variants
Bookmarks
Gradle
Device Manager
Device File Explorer
Emulator
Notifications
Run
Problems
Git
Terminal
Logcat
TODO
Profiler
App Quality Insights
App Inspection
Layout Inspector
Failed to start monitoring emulator-5554 (2023-02-20, 00:57)
main

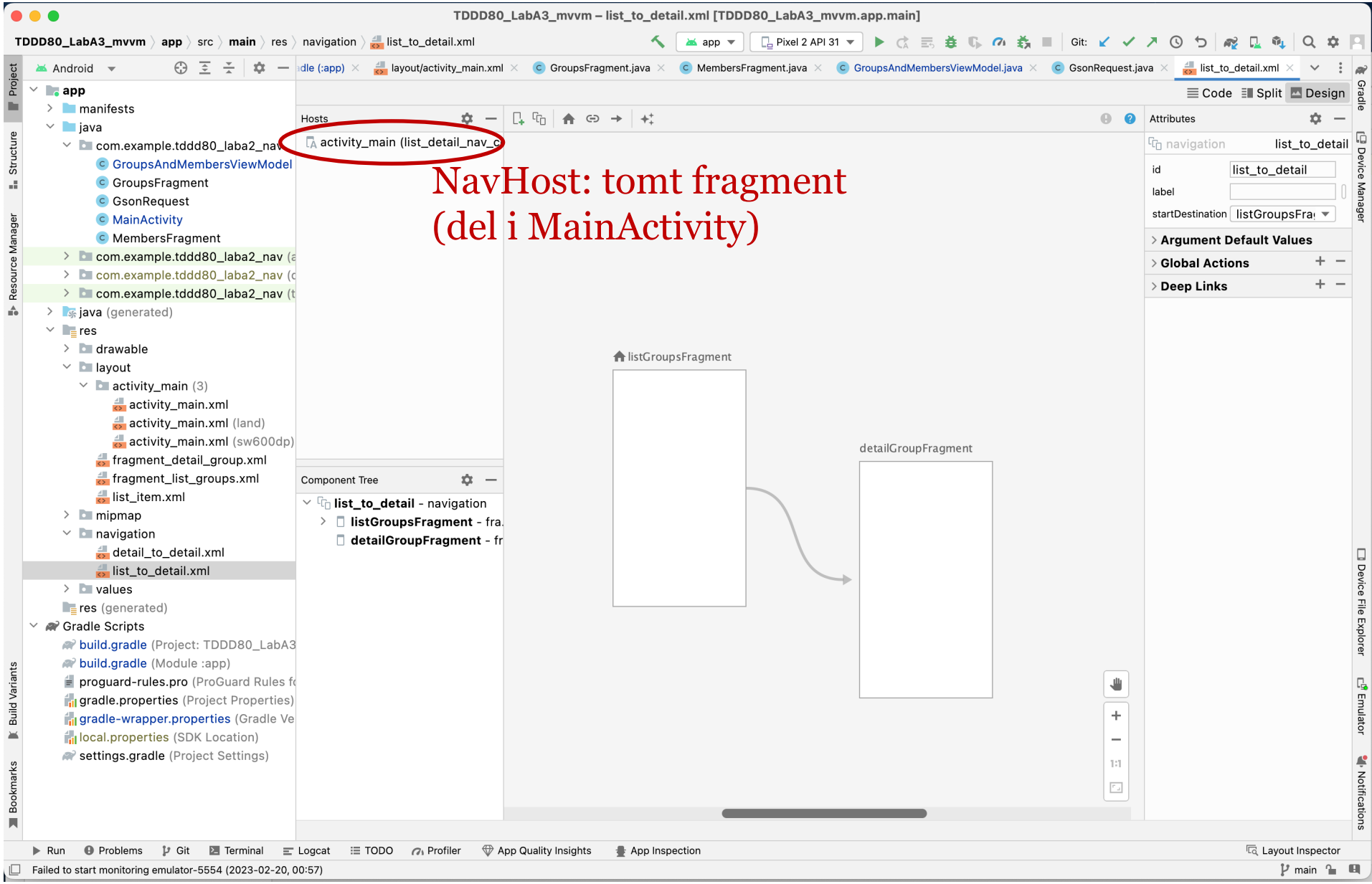

TDDD80_LabA3_mvvm – list_to_detail.xml [TDDD80_LabA3_mvvm.app.main]
visas först: PrimaryFragment
listGroupsFragment
detailGroupFragment


TDDD80_LabA3_mvvm – list_to_detail.xml [TDDD80_LabA3_mvvm.app.main]
action från fragmentet
listGroupsFragment
detailGroupFragment
Hosts
activity_main (list_detail_nav_container)
Component Tree
list_to_detail - navigation
listGroupsFragment - fragment
action_list_to_detail - action
detailGroupFragment - fragment
Attributes
Actions
detailGroupFragment
Deep Links
navigation › fragment


TDDD80_LabA3_mvvm – list_to_detail.xml [TDDD80_LabA3_mvvm.app.main]
id: list_to_detail
action
listGroupsFragment
detailGroupFragment
Hosts
activity_main (list_detail_nav_container)
Component Tree
list_to_detail - navigation
listGroupsFragment - fragment
action_list_to_detail - action
detailGroupFragment - fragment
Attributes
action
action_list_to_detail
id
destination
detailGroupFr
Animations
enterAnim
exitAnim
popEnterAnim
popExitAnim
Argument Default Values
group_idx
integer
Pop Behavior
popUpTo
popUpToInclu...
Launch Options
launchSingleT...
navigation › fragment


TDDD80_LabA3_mvvm – list_to_detail.xml [TDDD80_LabA3_mvvm.app.main]
argument
group_idx: integer
listGroupsFragment
detailGroupFragment
efter övergång: DetailFragment


oA3_mvvm.app.main]
el 2 API 31
Git:
ava
GroupsAndMembersViewModel.java
GsonRequest.java
list_to_detail.xml
Code
Split
Design
Attributes
fragment detailGroupFragment
id ailGroupFragment
label ent_detail_group
name MembersFrag
Arguments
group_idx: integer
Actions
Deep Links
group_idx: integer
detailGroupFragment
Gradle
Device Manager
Device File Explorer
Emulator


TDDD80_LabA3_mvvm – list_to_detail.xml [TDDD80_LabA3_mvvm.app.main]
TDDD80_LabA3_mvvm 〉 app 〉 src 〉 main 〉 res 〉 navigation 〉 list_to_detail.xml
Android
app
manifests
java
com.example.tddd80_laba2_nav
GroupsAndMembersViewModel
GroupsFragment
GsonRequest
MainActivity
MembersFragment
com.example.tddd80_laba2_nav
com.example.tddd80_laba2_nav
com.example.tddd80_laba2_nav
java (generated)
res
drawable
layout
activity_main (3)
activity_main.xml
activity_main.xml (land)
activity_main.xml (sw600dp)
fragment_detail_group.xml
fragment_list_groups.xml
list_item.xml
mipmap
navigation
detail_to_detail.xml
list_to_detail.xml
values
res (generated)
Gradle Scripts
build.gradle (Project: TDDD80_LabA3
build.gradle (Module :app)
proguard-rules.pro (ProGuard Rules f
gradle.properties (Project Properties)
gradle-wrapper.properties (Gradle Ve
local.properties (SDK Location)
settings.gradle (Project Settings)
layout/activity_main.xml
GroupsFragment.java
MembersFragment.java
GroupsAndMembersViewModel.java
GsonRequest.java
list_to_detail.xml
Code
Split
Design
Hosts
activity_main (list_detail_nav_c
NavHost: tomt fragment
(del i MainActivity)
Component Tree
list_to_detail - navigation
listGroupsFragment - fra
detailGroupFragment - fr
Attributes
navigation
list_to_detail
id
list_to_detail
label
startDestination
listGroupsFra
Argument Default Values
Global Actions
Deep Links
listGroupsFragment
detailGroupFragment
Run
Problems
Git
Terminal
Logcat
TODO
Profiler
App Quality Insights
App Inspection
Layout Inspector
Failed to start monitoring emulator-5554 (2023-02-20, 00:57)
main

activity_main.xml

Måste läggas till manuellt i xml

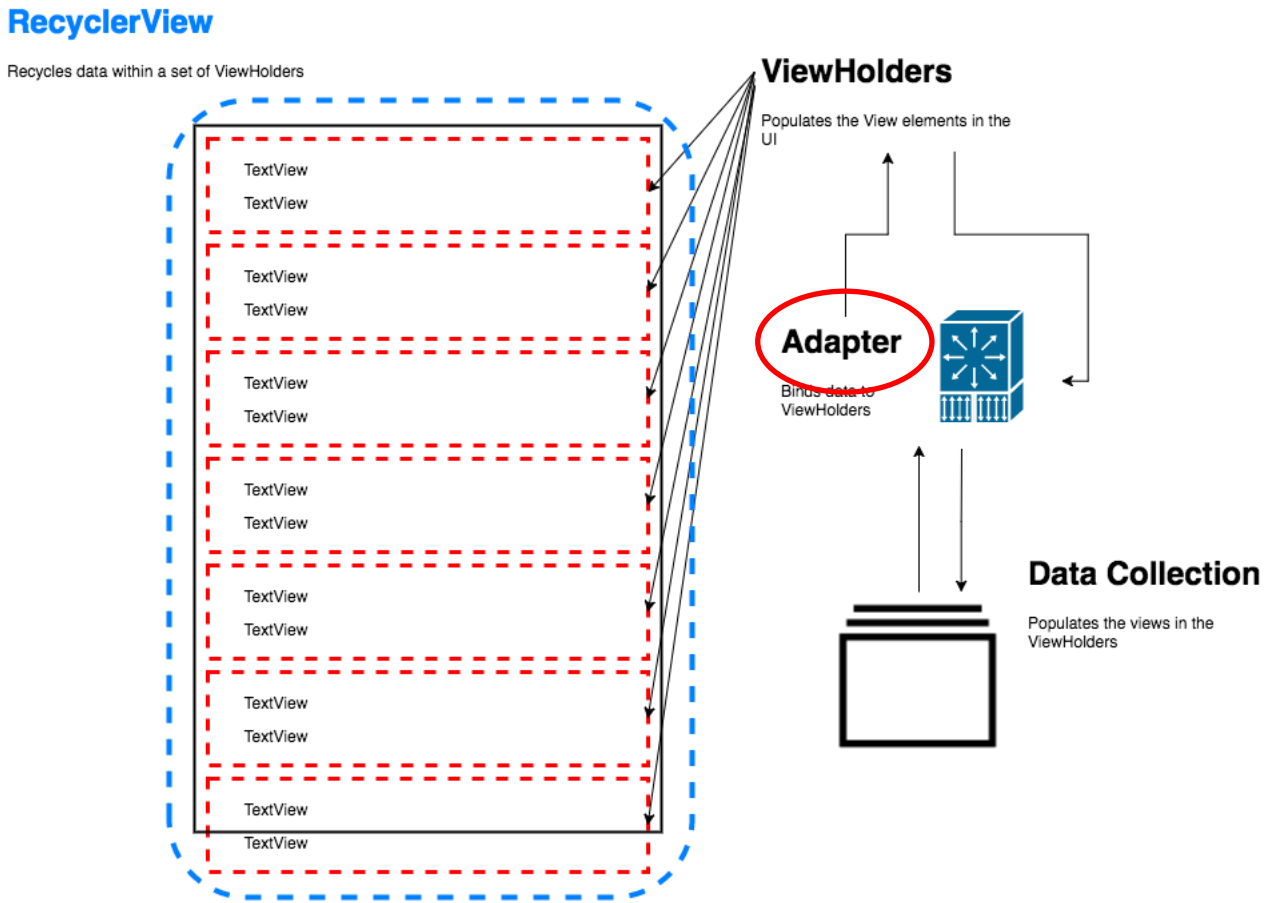
```
<?xml version="..." ...ng="utf-8"?>
<andr...ntlayout.widget.ConstraintLayout ...>

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/nav_container"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:defaultNavHost="true"
        app:navGraph="@navigation/list_to_detail" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Måste lägga till manuellt i xml

- FragmentContainerView med ett (tomt) NavHost fragment
- Läs mer här:
  - https://developer.android.com/guide/navigation/navigation-getting-started#add-navhostfragment

Genomföra action

En ViewHolder i listan fångar upp klick



Skicka upp klick till Activity

Ska jobba med denna ViewHolder

```
public class MyRecyclerAdapter extends
                        RecyclerView.Adapter<MyRecyclerAdapter.GroupNameHolder> {
    …
    public class GroupNameHolder extends RecyclerView.ViewHolder {…}

@Override
public void onBindViewHolder(…, int position) {
  …
  holder.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
      …
      ((MainActivity) requireContext()).onListItemClick(v, position);
    }
  });
}
```

Inre klass i MyRecylerAdapter

Ber Android OS om den Activity som fragmentet är bundet till (och cast:ar)

I Activity: förbered och genomför action

```
public void onListItemClick(View v, int position) {
    …
    PrimaryFragmentDirections.ActionListToDetail action =
                        PrimaryFragmentDirections.actionListToDetail(position);
    action.setGroupIdx(position);
    Navigation.findNavController(this, binding.nav_container).navigate(action);
```

Utan Safe Args

Med Safe Args

- Autogenererade klasser och metoder
  - Utifrån klassnamn och id:n i nav_graph.xml

Läs mer på

- https://developer.android.com/guide/navigation/navigation-getting-started

Hämta data från servern

Nätverksanrop

- Volley
  - Googles egna bibliotek
  - http://developer.android.com/training/volley/index.html
- Retrofit
  - Inbyggd JSON-avkodning
  - https://square.github.io/retrofit/

build.gradle (app)

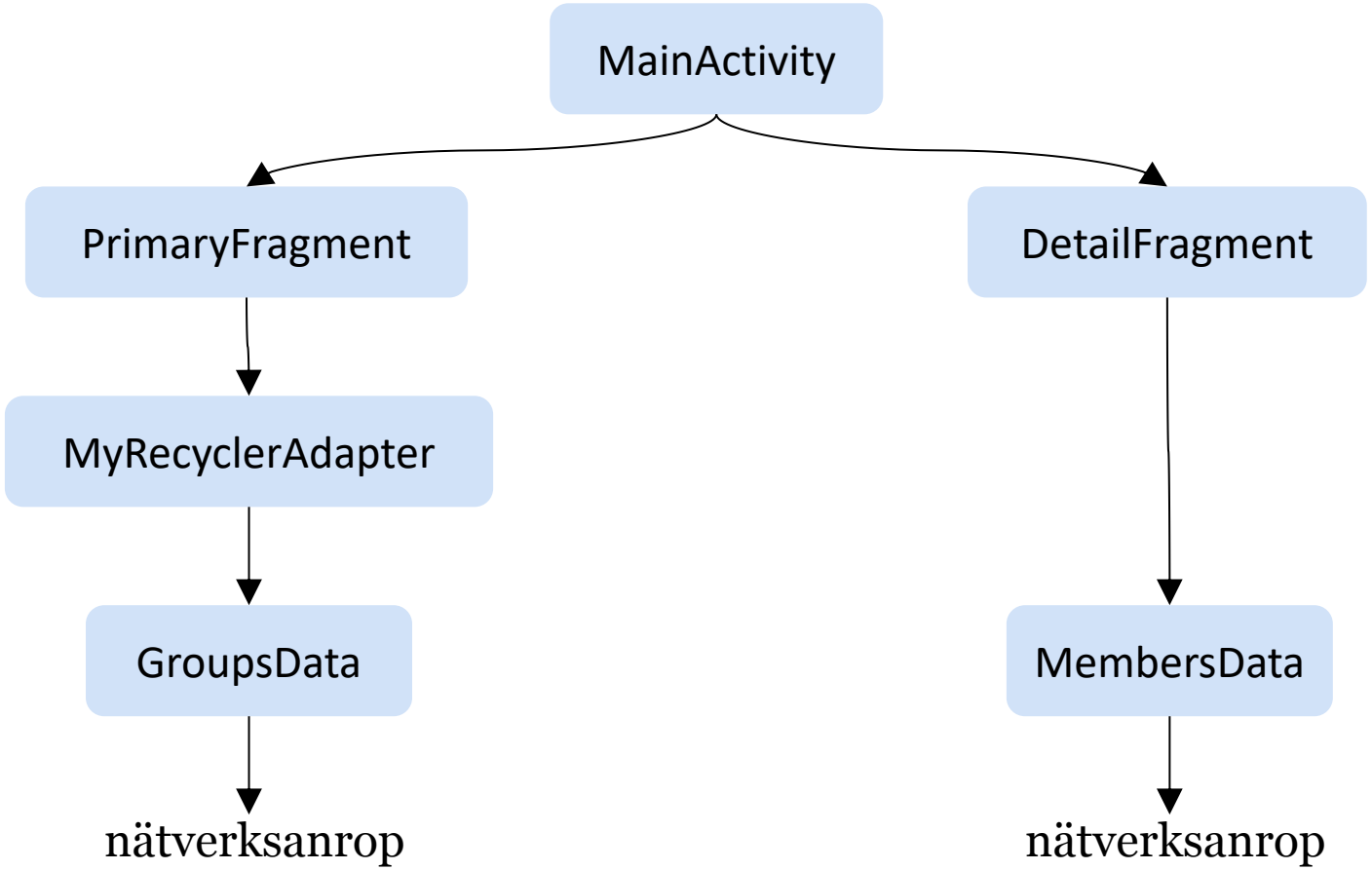
```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.8.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    implementation 'com.android.volley:volley:1.2.1'
    implementation 'com.google.code.gson:gson:2.10.1'
```

* Använd alltid senaste biblioteken

I build.gradle (project)

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
buildscript {
    repositories {
        google()
    }
    dependencies {
        def nav_version = "2.5.3"
        classpath "androidx.navigation:navigation-safe-args-gradle-plugin:$nav_version"
    }
}

plugins {
    id 'com.android.application' version '7.4.1' apply false
    id 'com.android.library' version '7.4.1' apply false

}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

App-struktur



Exempel på request (t.ex. i GroupsData)

```
request = new MyGsonRequest<GroupsNames>(
        Method.GET
        ”http://.../grupper”,
        GroupsNames.class,
        null,  // no special headers
        null,  // no body
        …,    // olika call-backs, dvs. lyssnare)


…
requestQueue.add(request);
```

Packa upp JSON-responsen till ett objekt av typen GroupsNames

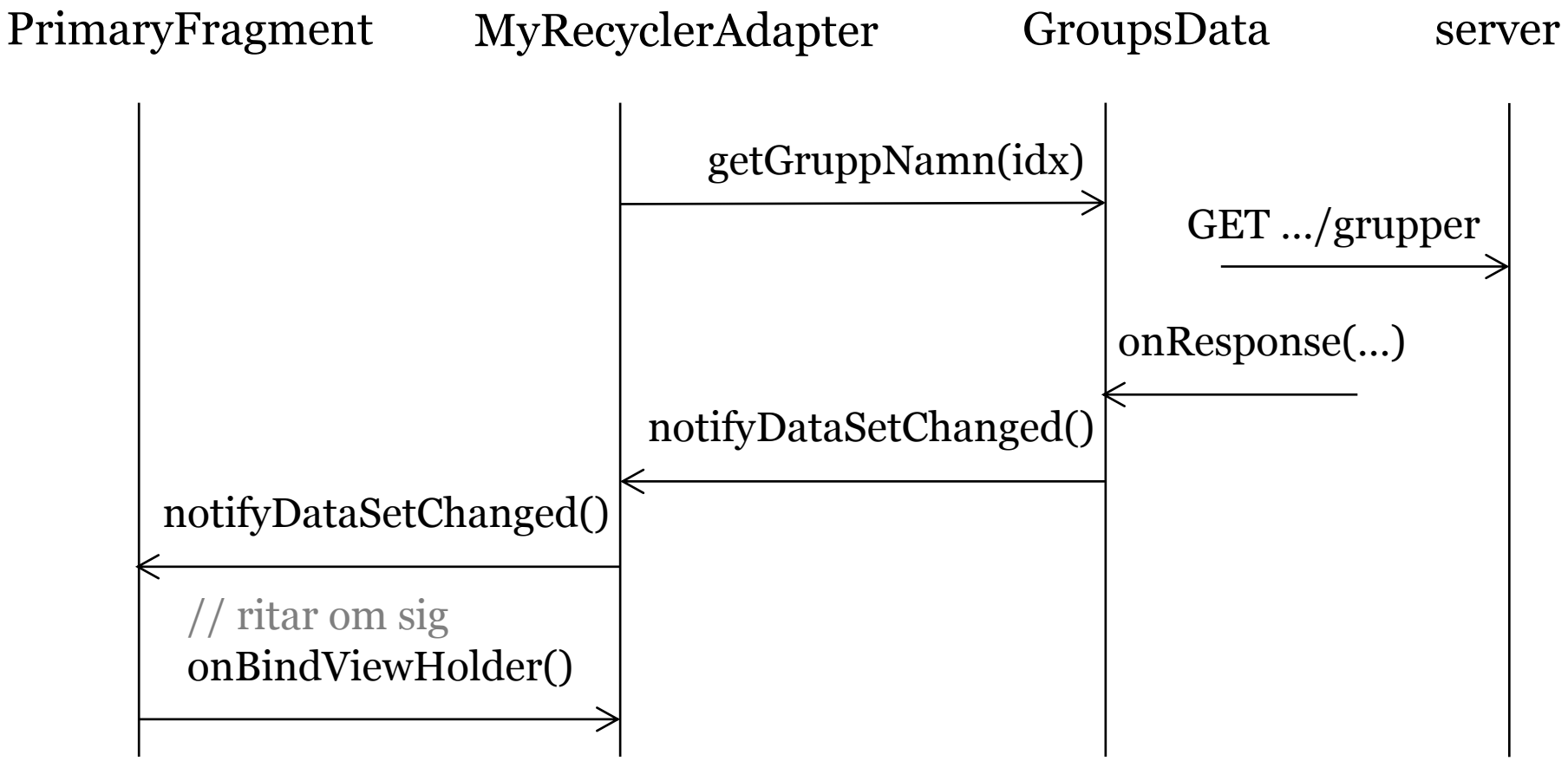
Exempel på request (forts)

```
request = new MyGsonRequest<GroupNames>(
        … // method, URL, etc.
        new Response.Listener<GroupNames>() {
            @Override
            public void onResponse(GroupNames groupNames) {
                updateGroups(groupsNames);
            }
        },
        …

private void updateGroups(GroupNames groupNames) {
    this.groupNames = groupNames;
    notifyDataSetChanged();
```

Anropas när det finns respons från servern

Respons redan avkodad: JSON -> Groups!

Sekvensdiagram



Skicka JSON i request, vid inloggning

```
headers = new HashMap<String, String>();
headers.put(”content-type”, ”application/json”);
payLoad = new MyLoginData(userEmail, userPassword);
body = gson.toJson(payLoad);

request = new GsonRequest<>(Method.POST, …, headers, body, …);
requestQueue.add(request);
```

Ta emot token

```
request = new GsonRequest<MyTokenHolder>(
    Method.GET,
    MyTokenHolder.class,
    headers,
    body,
    …);
```

Packa upp response till en instans av den här klassen

Response from server

```
{ "access_token": ”server-generated.jwt.here",
  "token_type": ”Bearer",
  "expires_in": 3600
}
```

Skicka token i headers

```
headers = new HashMap<String, String>();
accessToken = myTokenHolder.getToken()
headers.put(”Authorization”, ”Bearer ” + accessToken);

request = new GsonRequest<>(…, headers, …);
requestQueue.add(request);
```

Best practice

- Man kan med fördel skapa en basklass MyGsonRequest
- Subklasser för olika typer av requests man vill göra
  - GET, POST, etc.
  - Dessa kommer att vilja ha olika typer av parametrar vid anrop

GsonRequest

Bygger ihop requests – packar upp response

Definiera GsonRequest

```
public class MyGsonRequest<T> extends Request<T> {

    private final Gson gson = new Gson();

    public MyGsonRequest(int method, String url, Class<T> clazz, …)  {
        super(method, url, …);
        this.clazz = clazz;
        …
    }
```

Skapa en intern översättare

Vilken data-klass responsen ska packas upp till

GsonRequest (uppackning av respons)

```
@Override
protected Response<T> parseNetworkResponse(NetworkResponse response)
{
   try {
      json = new String(response.data,
                        // encoding info from headers);
      return Response.success(
            gson.fromJson(json, clazz),
            // get response code from headers);
   }
```

Biblioteket Gson

- Googles egna bibliotek för att parsa JSON direkt till Java objekt (av en viss klass)
- Man anger att man vill ha objekt av typen Groups.class när man instansierar ett GsonRequest
- Gson sköter nerpackning och uppackning av data till/från Groups

- Kräver att Groups är en Java Bean

Java Bean (standardiserad java klass)

- Null-konstruktor
- Var. namn matchar JSON-nycklarna
- Finns getters och setters för alla var.

- Möjliggör automatisk uppackning av JSON till objektet

Mer om GsonRequest (med kodexempel)

- https://google.github.io/volley/request-custom.html#example-gsonrequest

Retrofit (alternativ till Volley)

Bibliotek med liknande syntax som Flask

Server-routes (Python/Flask):

```
@app.route(’/users/new', methods=['POST’])
def messages_post():
    …
@app.route(’/group/<id>/users', methods=[’GET’])
def messages_post(id):
    …
```

Definiera ett interface med alla routes

```
public interface MyRequestService {
    @POST("users/new")
    Call<User> createUser(@Body User user);

    @GET("group/{id}/users")
    Call<List<User>> getGroupMem(@Path("id") int groupId);
}
```

Exempel på användning

```
// Initiera Retrofit med inställningar
Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("https://TDDD80.myserver.com/")
        .build();

// Skapa definitioner av alla metoder i MyRequestService
RequestService service = retrofit.create(MyRequestService.class);

// Sätt ihop request
Call<List<User>> call = service.getGroupMem(”hackers");
```

Exempel på användning (forts)

```
…
Call<List<User>> call = service.getGroupMem(”hackers”);
call.enqueue(new Callback<List<User>>() {
        @Override
        public void onResponse(Call<List<User>> call,
                                        Response<List<User>> resp) {
                myData.setValue(resp.body());
        }
}
```

Skicka med access token i header

```
// dynamisk header
// TODO: dubbelkolla att ‘Bearer’ har skickats med
@GET("user")
Call<User> getUser(@Header("Authorization") String authToken)

// key-value headers, t.ex. HashMap, motsv. Pyhon dictionary
@GET("user")
Call<User> getUser(@HeaderMap Map<String, String> headers)
```

För smidig avkodning med Gson

I er app-nivå build.gradle

```
dependencies {
    implementation 'com.google.code.gson:gson:2.10.1’
    implementation 'com.squareup.retrofit2:retrofit:2.9.0’
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0’ }
```

* Ta alltid senaste versionerna av alla bibliotek

För att avkoda med Gson

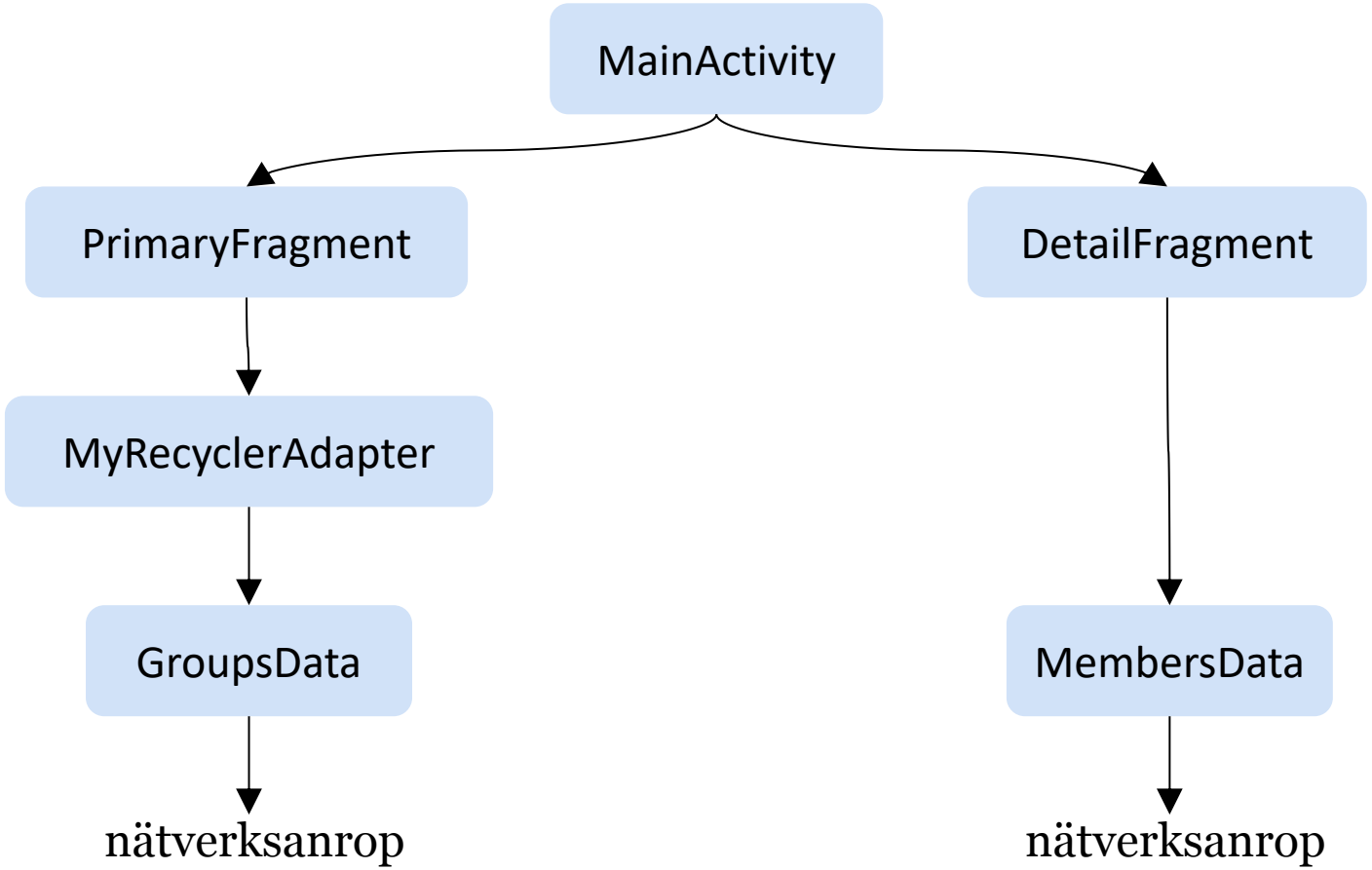
```
// Initiera med mina inställningar
Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("https://TDDD80.myserver.com")
        .addConverterFactory(GsonConverterFactory.create())
        .build();

// Skapa instans av er request-service som vanligt
RequestService service = retrofit.create(MyRequestService.class);
```

Läs mer på…

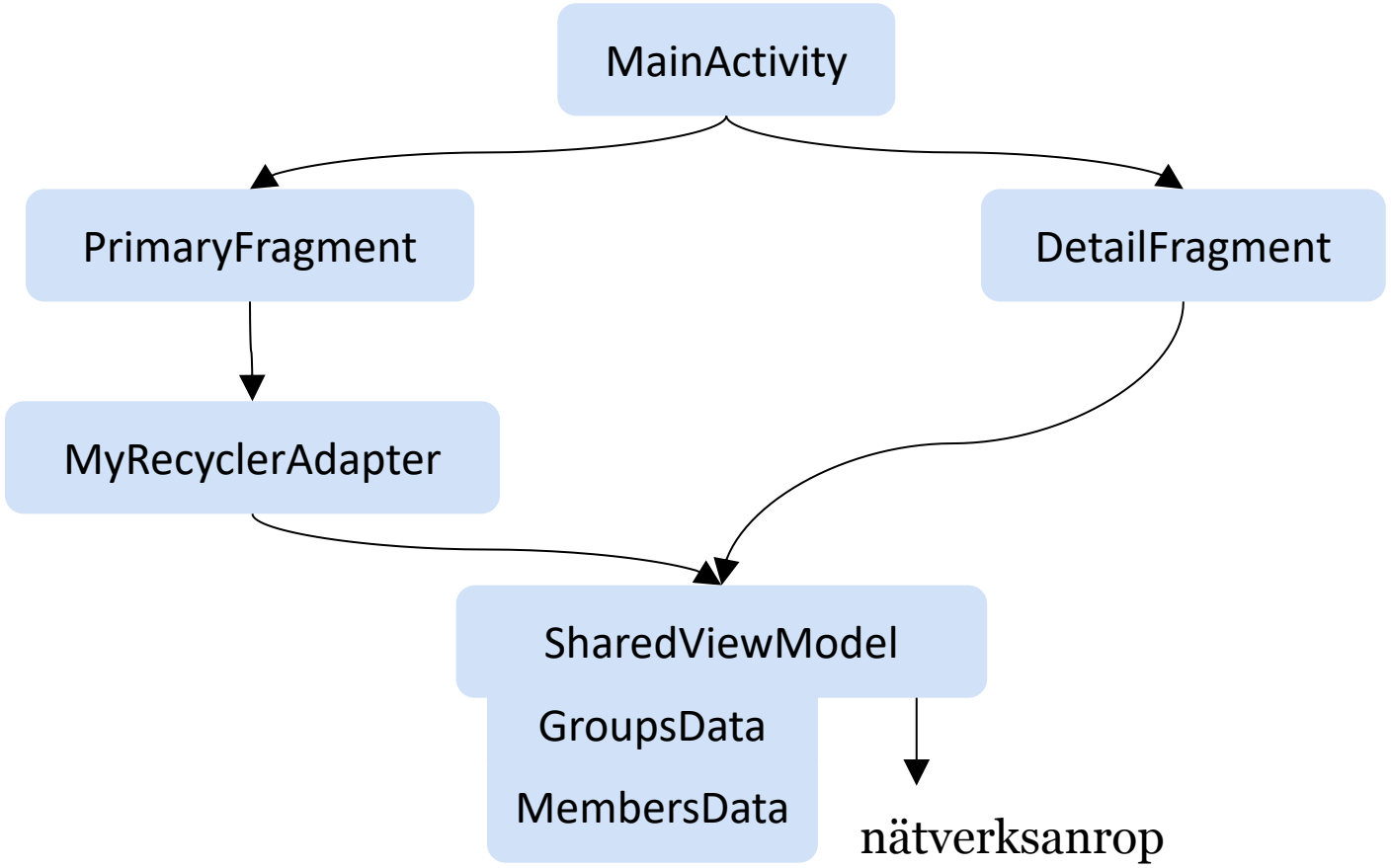
http://square.github.io/retrofit/

Bra att ha för projektet: ViewModel

App-struktur



Bättre app-struktur



ViewModel kapslar in data

```
public class SharedViewModel extends ViewModel {
    public MutableLiveData<ArrayList<String>> groups = new
                                        MutableLiveData<ArrayList<String>>();
    public MutableLiveData<String> members = new
                                        MutableLiveData<String>();
    private int selectedIdx = 0;
    private RequestQueue queue;
```

Shared ViewModel

- Scopa ViewModel till MainActivity från båda fregmenten, så att ViewModel får samma ägare
  - Då får vi samma instans av ViewModel

```
mViewModel = new ViewModelProvider(requireActivity()).
                                        get(SharedViewModel.class);
```

Exempel: PrimaryFragment

```
public class MyAdapter extends RecyclerView.Adapter<…> {
    private ArrayList<String> data;

    public MyAdapter() {
        mViewModel = new …;
        mViewModel.groups.observe(…, new Observer<…>() {
            @Override
            public void onChanged(ArrayList<String> groups) {
                this.data = groups;
                notifyDataSetChanged();
            }
        });
    }
```

LiveData i ViewModel har ändrats (t.ex. nätverksanrop är klart)

Meddela Fragmentet att dags att rita om sig

Fragmenten kommunicerar via ViewModel

```
public class MyRecyclerAdapter extends


@Override
public void onBindViewHolder(…, int position) {
    …
    holder.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mViewModel.setSelectedIdx(position);
            ((MainActivity) requireContext()).onListItemClick(v, position);
        }
    });
}
```

Läs vidare på

https://developer.android.com/jetpack/docs/guide

https://developer.android.com/topic/libraries/architecture/viewmodel#java

www.liu.se