TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Systems Architecture, Chair of Computer Networks

APPLICATION DEVELOPMENT FOR MOBILE AND UBIQUITOUS COMPUTING

Second Presentation

Group No. 15: Jasim Al-Krdy and Danny Kiefner

Dresden, 17.12.2010

Content

Why HappyDroid?

- Find HappyHours next to you
- See friends at HappyHour-locations
- Jabber with friends
- Rate and watch ratings for locations

Current Status

Current Status

- Build up connection between client and server ✓

Current Status

- Build up connection between client and server ✓
- Define communication-behavior and packets ✓

Current Status

- Build up connection between client and server ✓
- Define communication-behavior and packets ✓
- Test communication while using dummy-data ✓

Current Status

- Build up connection between client and server ✓
- Define communication-behavior and packets ✓
- Test communication while using dummy-data ✓
- Design client-views x

Overview XMPP

Overview XMPP

General:

- Open technology
- Mainly for IM and presence-information
- Identifier JID (node@domain/resource)
- Many Extensions (XEP)

Overview XMPP

General:

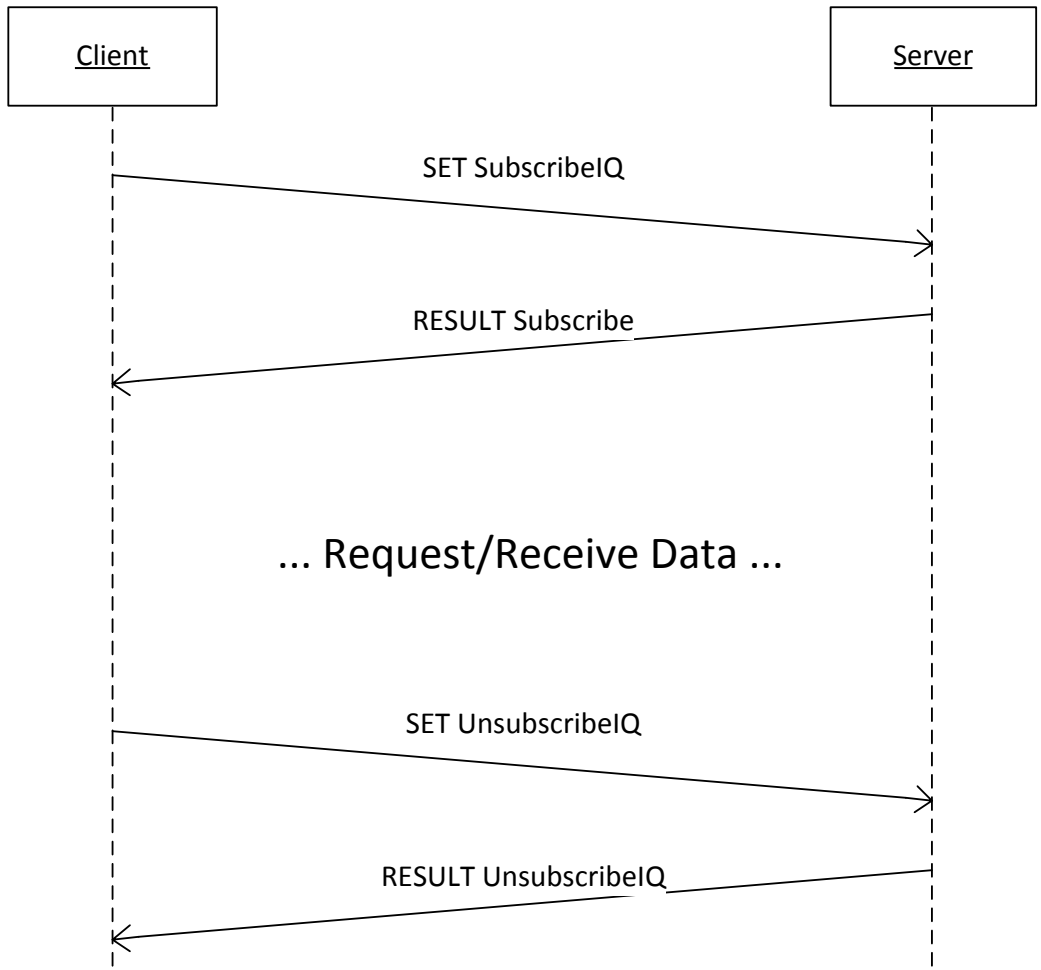
- Open technology
- Mainly for IM and presence-information
- Identifier JID (node@domain/resource)
- Many Extensions (XEP)

Packets:

- XML-standardized
- 3 XML-stanzas: message, presence, iq
- 4 stanza-types: set, get, result, error

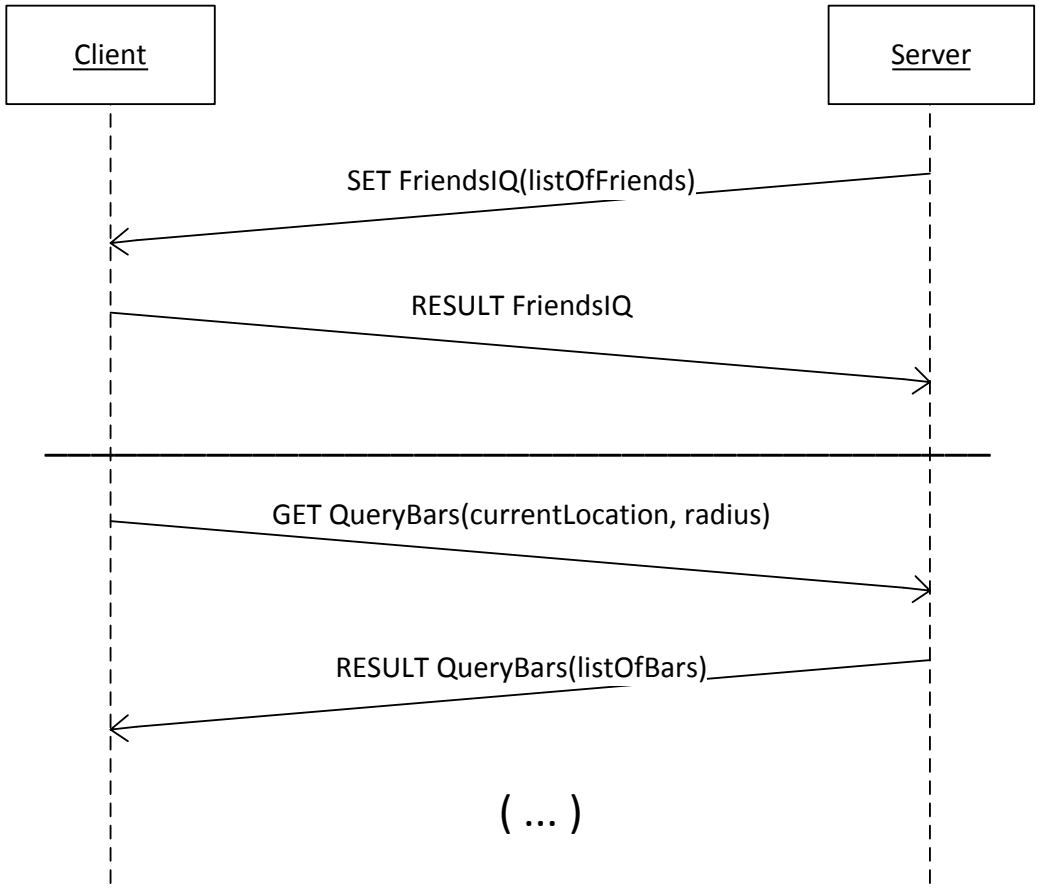
Communication and Architecture

Communication - Sequences (Subscription)



Communication and Architecture

Communication - Sequences (Query Bars/Friends)



Communication and Architecture

Communication - Packets

Packet	From Client	From Server
SubscribeIQ	SET	RESULT
UnsubscribeIQ	SET	RESULT
QueryBarsIQ	GET	RESULT
FriendsIQ	RESULT	SET
AddFriendIQ	SET	RESULT
DeleteFriendIQ	SET	RESULT
CheckInIQ	SET	RESULT
RateBarIQ	SET	RESULT

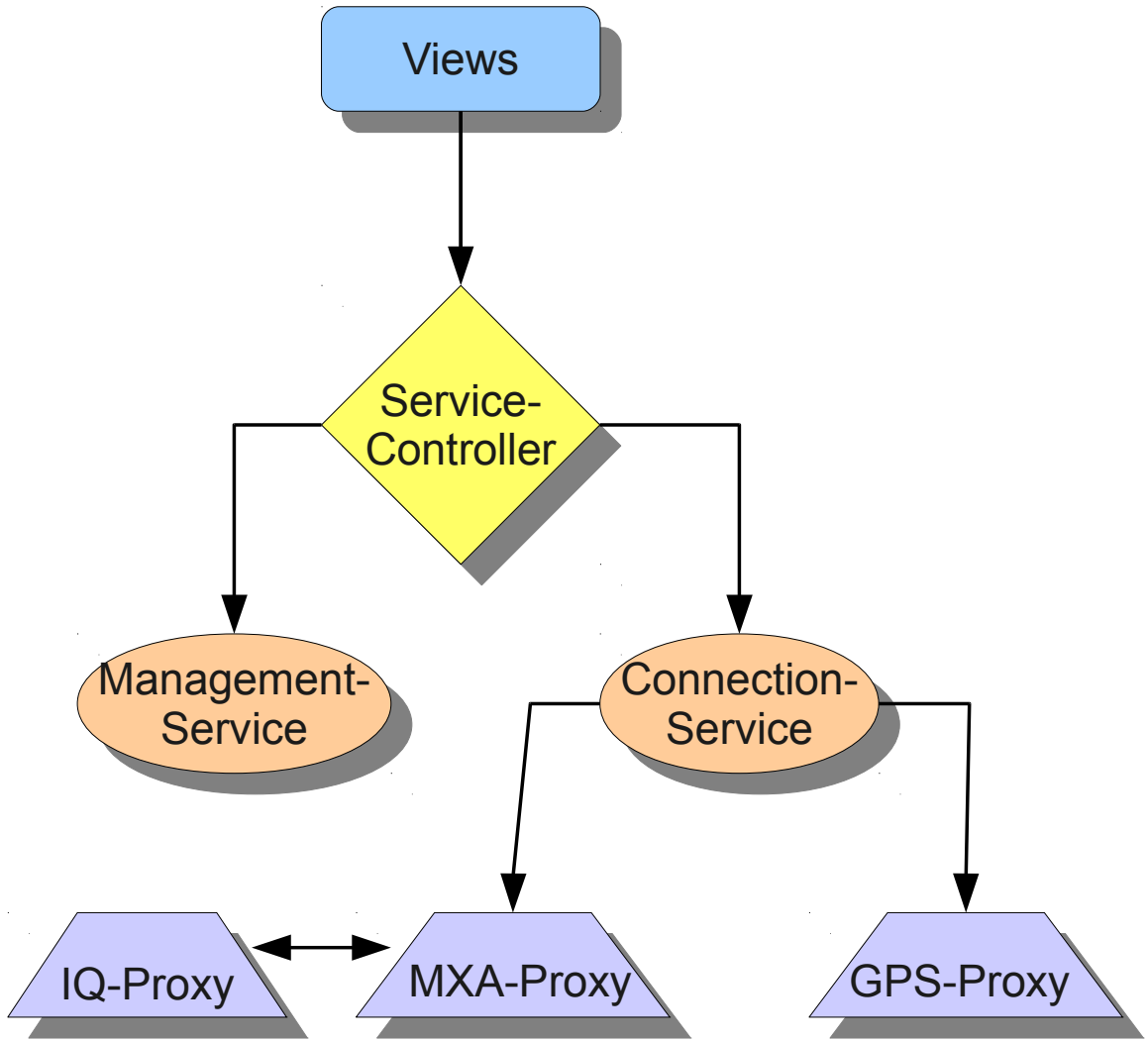
Communication and Architecture

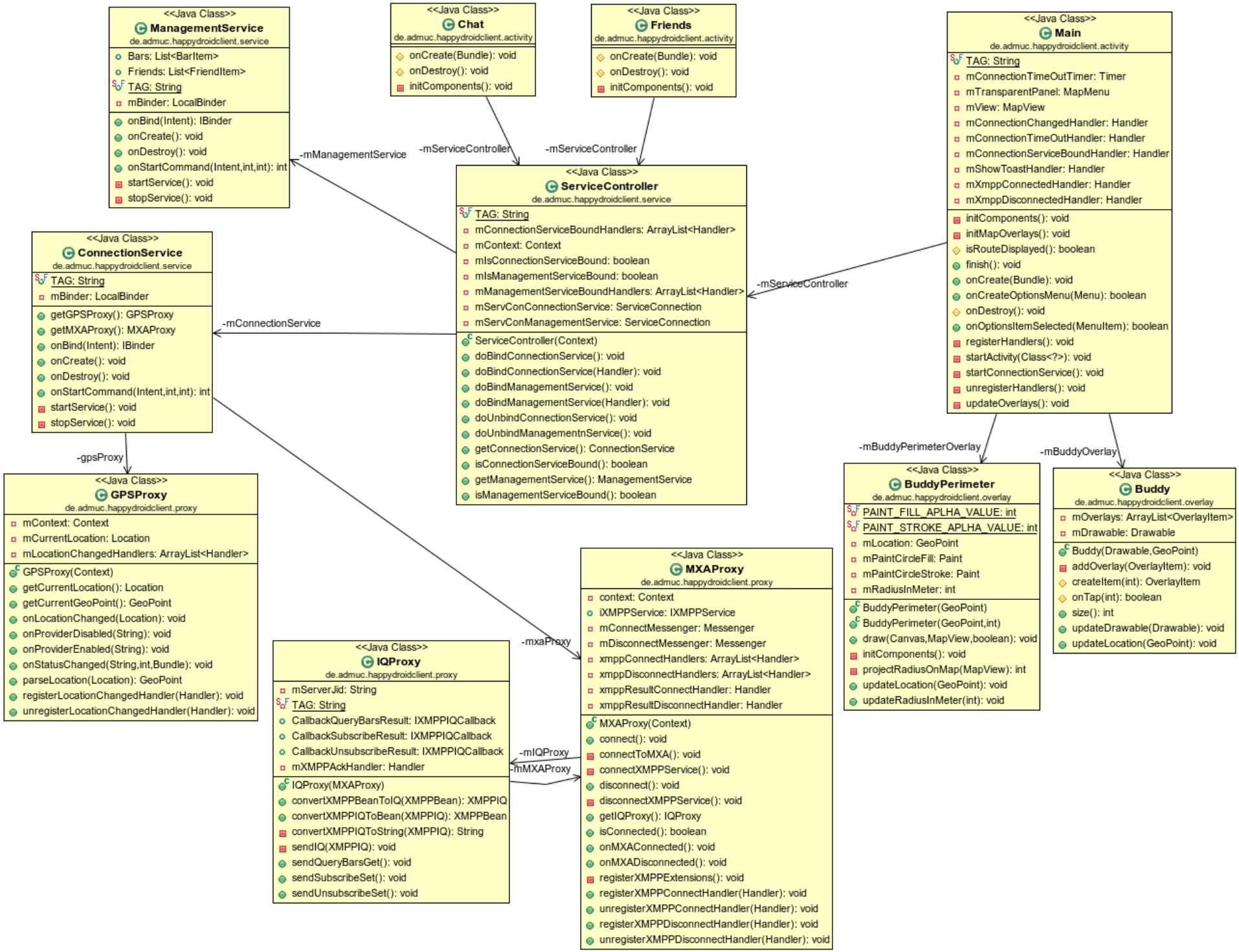
Communication - QueryBarsIQ(RESULT)

```
<iq type='result' from='server@domain/resource' to='client@domain/resource'>
        <query xmlns="happydroid:iq:querybarsresult">
                <bar>
                        <bid>1</bid>
                        <name>SKY</name>
                        <lat>51041501</lat>
                        <lon>13732792</lon>
                        <avgrating>3.5</avgrating>
                        <numvoters>11</numvoters>
                        <hhtimes>
                                <hhtime>
                                        <hstart>17</hstart>
                                        <hend>19</hend>
                                        <mday>1</mday>
                                </hhtime>
                                <hhtime>
                                        <hstart>23</hstart>
                                        <hend>1</hend>
                                        <mday>2</mday>
                                </hhtime>
                        </hhtimes>
                </bar>
                ...
        </query>
</iq>
```

Communication and Architecture

Architecture




<<Java Class>>
ManagementService
de.admuc.happydroidclient.service
Bars: List<BarItem>
Friends: List<FriendItem>
TAG: String
mBinder: LocalBinder
onBind(Intent): IBinder
onCreate(): void
onDestroy(): void
onStartCommand(Intent,int,int): int
startService(): void
stopService(): void
<<Java Class>>
Chat
de.admuc.happydroidclient.activity
onCreate(Bundle): void
onDestroy(): void
initComponents(): void
<<Java Class>>
Friends
de.admuc.happydroidclient.activity
onCreate(Bundle): void
onDestroy(): void
initComponents(): void
<<Java Class>>
Main
de.admuc.happydroidclient.activity
TAG: String
mConnectionTimeOutTimer: Timer
mTransparentPanel: MapMenu
mView: MapView
mConnectionChangedHandler: Handler
mConnectionTimeOutHandler: Handler
mConnectionServiceBoundHandler: Handler
mShowToastHandler: Handler
mXmppConnectedHandler: Handler
mXmppDisconnectedHandler: Handler
initComponents(): void
initMapOverlays(): void
isRouteDisplayed(): boolean
finish(): void
onCreate(Bundle): void
onCreateOptionsMenu(Menu): boolean
onDestroy(): void
onOptionsItemSelected(MenuItem): boolean
registerHandlers(): void
startActivity(Class<?>): void
startConnectionService(): void
unregisterHandlers(): void
updateOverlays(): void
-mManagementService
-mServiceController
-mServiceController
<<Java Class>>
ServiceController
de.admuc.happydroidclient.service
TAG: String
mConnectionServiceBoundHandlers: ArrayList<Handler>
mContext: Context
mIsConnectionServiceBound: boolean
mIsManagementServiceBound: boolean
mManagementServiceBoundHandlers: ArrayList<Handler>
mServConConnectionService: ServiceConnection
mServConManagementService: ServiceConnection
ServiceController(Context)
doBindConnectionService(): void
doBindConnectionService(Handler): void
doBindManagementService(): void
doBindManagementService(Handler): void
doUnbindConnectionService(): void
doUnbindManagementnService(): void
getConnectionService(): ConnectionService
isConnectionServiceBound(): boolean
getManagementService(): ManagementService
isManagementServiceBound(): boolean
-mServiceController
<<Java Class>>
ConnectionService
de.admuc.happydroidclient.service
TAG: String
mBinder: LocalBinder
getGPSProxy(): GPSProxy
getMXAProxy(): MXAProxy
onBind(Intent): IBinder
onCreate(): void
onDestroy(): void
onStartCommand(Intent,int,int): int
startService(): void
stopService(): void
-mConnectionService
-gpsProxy
-mBuddyPerimeterOverlay
-mBuddyOverlay
<<Java Class>>
GPSProxy
de.admuc.happydroidclient.proxy
mContext: Context
mCurrentLocation: Location
mLocationChangedHandlers: ArrayList<Handler>
GPSProxy(Context)
getCurrentLocation(): Location
getCurrentGeoPoint(): GeoPoint
onLocationChanged(Location): void
onProviderDisabled(String): void
onProviderEnabled(String): void
onStatusChanged(String,int,Bundle): void
parseLocation(Location): GeoPoint
registerLocationChangedHandler(Handler): void
unregisterLocationChangedHandler(Handler): void
<<Java Class>>
BuddyPerimeter
de.admuc.happydroidclient.overlay
PAINT_FILL_APLHA_VALUE: int
PAINT_STROKE_APLHA_VALUE: int
mLocation: GeoPoint
mPaintCircleFill: Paint
mPaintCircleStroke: Paint
mRadiusInMeter: int
BuddyPerimeter(GeoPoint)
BuddyPerimeter(GeoPoint,int)
draw(Canvas,MapView,boolean): void
initComponents(): void
projectRadiusOnMap(MapView): int
updateLocation(GeoPoint): void
updateRadiusInMeter(int): void
<<Java Class>>
Buddy
de.admuc.happydroidclient.overlay
mOverlays: ArrayList<OverlayItem>
mDrawable: Drawable
Buddy(Drawable,GeoPoint)
addOverlay(OverlayItem): void
createItem(int): OverlayItem
onTap(int): boolean
size(): int
updateDrawable(Drawable): void
updateLocation(GeoPoint): void
-mxaProxy
<<Java Class>>
MXAProxy
de.admuc.happydroidclient.proxy
context: Context
iXMPPService: IXMPPService
mConnectMessenger: Messenger
mDisconnectMessenger: Messenger
xmppConnectHandlers: ArrayList<Handler>
xmppDisconnectHandlers: ArrayList<Handler>
xmppResultConnectHandler: Handler
xmppResultDisconnectHandler: Handler
MXAProxy(Context)
connect(): void
connectToMXA(): void
connectXMPPService(): void
disconnect(): void
disconnectXMPPService(): void
getIQProxy(): IQProxy
isConnected(): boolean
onMXAConnected(): void
onMXADisconnected(): void
registerXMPPExtensions(): void
registerXMPPConnectHandler(Handler): void
unregisterXMPPConnectHandler(Handler): void
registerXMPPDisconnectHandler(Handler): void
unregisterXMPPDisconnectHandler(Handler): void
<<Java Class>>
IQProxy
de.admuc.happydroidclient.proxy
mServerJid: String
TAG: String
CallbackQueryBarsResult: IXMPPIQCallback
CallbackSubscribeResult: IXMPPIQCallback
CallbackUnsubscribeResult: IXMPPIQCallback
mXMPPAckHandler: Handler
IQProxy(MXAProxy)
convertXMPPBeanToIQ(XMPPBean): XMPPIQ
convertXMPPIQToBean(XMPPIQ): XMPPBean
convertXMPPIQToString(XMPPIQ): String
sendIQ(XMPPIQ): void
sendQueryBarsGet(): void
sendSubscribeSet(): void
sendUnsubscribeSet(): void
-mIQProxy
-mMXAProxy

Next Steps

Next Steps

Client:

- Finalize client-UI (friends, ratings)
- Location-based data-requesting

Next Steps

Client:

- Finalize client-UI (friends, ratings)
- Location-based data-requesting

Server:

- Data-storage using SQL
- Data-scrapping function for Bartime.de

Thank you for your attention!