Sharp.Xmpp, a multiplatform .NET XMPP client library, and Android

Panagiotis (Takis) Stathopoulos

https://twitter.com/panstath
http://pgstath.me

Presentation supported by the
Greek Free/Open Source Software Society

XMPP

GFOSS
Greek Free / Open Source Software Society

Outline

- Part A: Sharp.Xmpp
  - https://github.com/pgstath/Sharp.Xmpp
  - C# . Net multiplatform client XMPP library
  - Fork of S22.Xmpp
  - MIT License
- Part B: Android, Xamarin, and XMPP
  - Tips, tricks and code for long running XMPP sessions
  - and background TCP connections

XMPP

GFOSS
Greek Free / Open Source Software Society

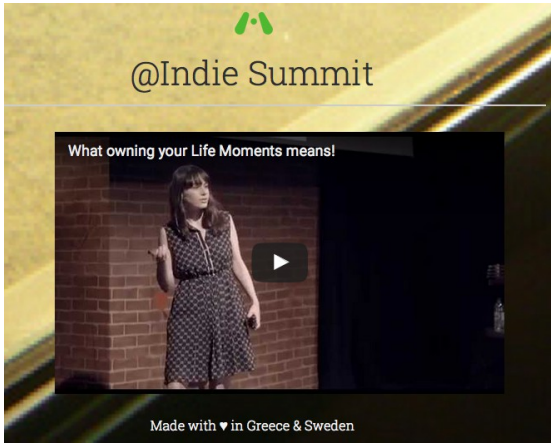
Why Xamarin and C#

Why XMPP?

Why yet another XMPP library?

Background

- Sharp.Xmpp mainly developed for use by Momentum.IM:
  - **Small-scale social networking app over XMPP**
  - Android/Windows/Linux
  - **App centric not web centric**
  - Check Georgia's presentation at Indie Summit: https://youtu.be/N7AP6HTjUlk



XMPP

GFOSS
Greek Free / Open Source Software Society

Why C# & Xamarin?

- A **controversial** subject but:
  - Built **native** apps in:
    - Android
    - Windows
    - (and Linux)
  - aka the **most widespread desktop and mobile platforms**
  - Client centric and not server centric technologies/app.
- Comes with a cost
- Mostly good (but not perfect) support of third party libraries

Why XMPP?

- Required a message distribution protocol:
  - Presence, subscription, message routing, etc.
  - Secure, **Private**. **OTR** support was must have.
- No **centralized control of messaging infrastructure** e.g. like:
  - Google Cloud Messaging or Apple Messaging Service
  - or closed browser specific technologies.
- Capacity to run **your own** messaging server
  - **Provide an app not a service**
- No need of any of IM specific functionality, but:
  - XMPP is not only for chat/messaging apps

XMPP

GFOSS
Greek Free / Open Source Software Society

Why yet another library?

- Modern technologies
- Clean and easy to extend
- MIT Licensed for most flexibility
- And finally:
  - Not so many C# libraries available

What's on the roadmap

- https://github.com/pgstath/Sharp.Xmpp
- So far: Improved disconnection detection, vCard Avatars support, DNS queries, port to Xamarin, bug fixes, etc.
- Third party has already added:
  - Message Carbons, Archive management, basic MUC , etc.
  - Will integrate in a next version
- Next step: OTR support.

Part B': XMPP, Android and Xamarin

- Android hates long running listening background TCP connections
  - Or background tasks
- Mobile OS with limited resources:
  - Kill Activities/Services on every opportunity
  - Independent from implementation of XEP-0198 stream management
- Already some good presentations by Smack during previous years
  - Provide Xamarin/C# perspective
  - **And a full sample project**

XMPP

GFOSS
Greek Free / Open Source Software Society

Show me the code

- https://github.com/pgstath/SharpXmppDemo
- pgstath.me and codeproject article coming just after FOSDEM2016
- Also check related XMPP resources http:bit.ly/1QMjS5n:
  - Securely register users from mobile app
  - Long running background process

Components

- A “**Sticky**” Android service
  - With Intent Service like API.
- Periodic (inexact) Android Alarms
- Wakelocks and WakefulBroadcastReceivers
- Network connectivity monitoring events
- XMPP features:
  - Ping
  - Server ping before disconnection

XMPP

GFOSS
Greek Free / Open Source Software Society

Sticky Intent-like Service

- Place your XMPP connection object in a StickyIntentService
  - XMPP must be placed in a Sticky Service, listening TCP connection even if no work is done.
- Provide StickyIntentService
  - A class with an IntentService compatible API & START_STICKY behavior
  - The simple Intent Service is not “Sticky”
  - See http://bit.ly/1PZv1er for details

XMPP

GFOSS
Greek Free / Open Source Software Society

Alarms, Wakeful Receivers and Wakelocks

- Set an **inexact** Android Alarm
  - Ping, to keep connection alive, or connect if network is present.
- "**Wakeful**" Broadcast Alarm receiver
  - OS might sleep even before connection is made
  - Wakelocks are a mechanism to prevent OS from sleeping
    - Required for Alarm Broadcast receivers
    - Long running tasks, e.g. **receiving** a file transfer

Network monitoring, built in XMPP

- Make use of Android network connectivity events
  - Set broadcast receivers
- Make use of built in XMPP features:
  - XMPP ping when an alarm is fired
  - Server side ping before disconnection
- **Results for 15min alarms:**
  - Very low battery background usage, e.g. %
  - Persistent network connection
    - Easily pushing messages, with out XEP push notifications!

XMPP

GFOSS
Greek Free / Open Source Software Society

SharpXmppDemo Classes

- **BackgroundService**, **StickyIntentService**, **Utils**
- **SharpComms** & **IUIThreadDispatcher**, multiplatform wrappers:
  - Detecting and managing reconnection timers
  - Wakelocks
  - Debug messages
  - Run actions on the UI Thread

Next Steps

- SharpXmppDemo as a separate library?
- Future announcements about:
  - Middleware for **end point controlled messages replication** (provisional name: TrustVillageNet)
- Store or replicate nothing in the server!
- Minimize assumptions about server functionality!

Thank you, stay tuned at:

https://github.com/pgstath