



**Notifra**

Powerful Distributed Notification Framework  
for Android

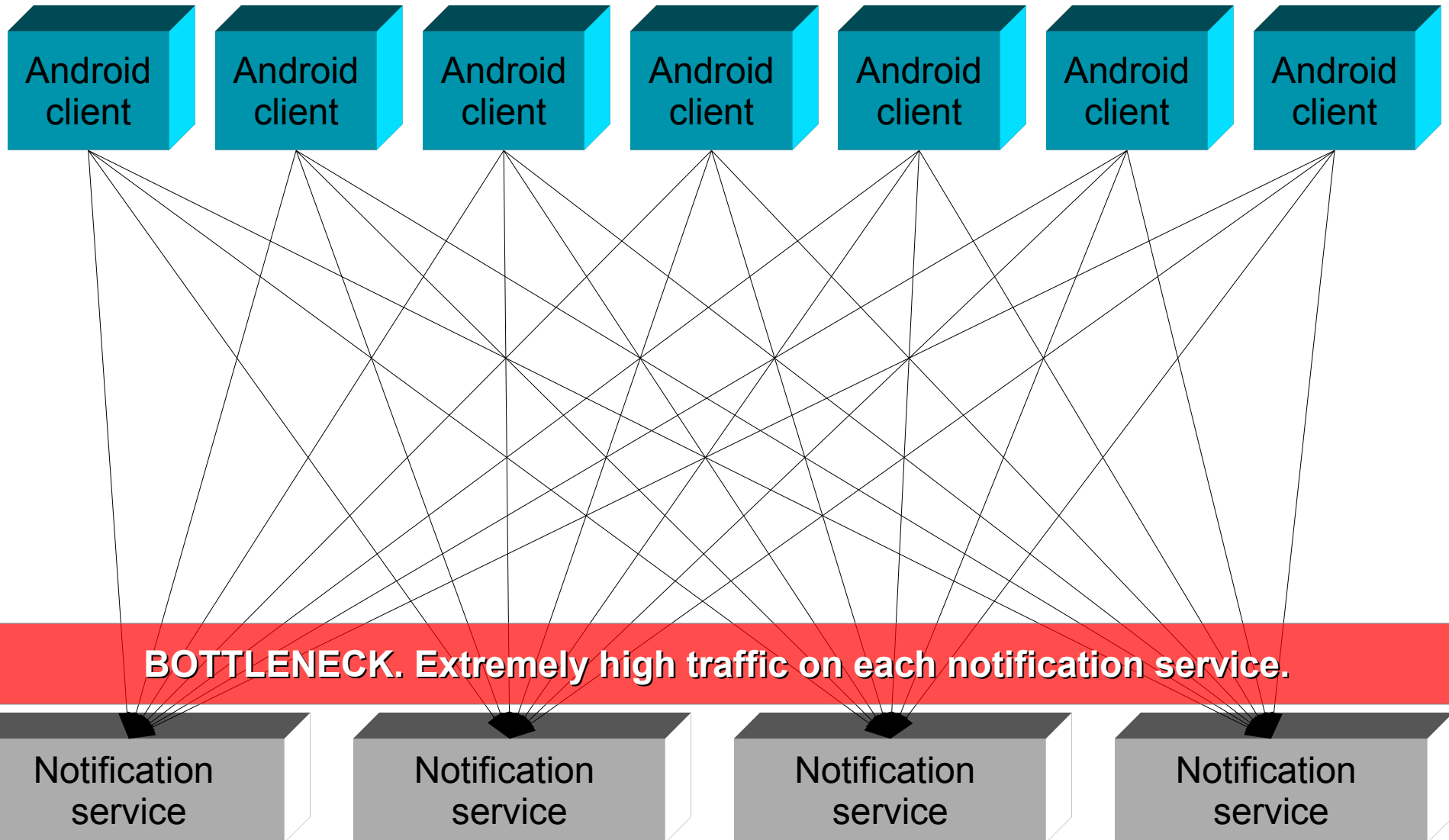
Tomasz Mikolajewski  
30/01/3012

# Outline

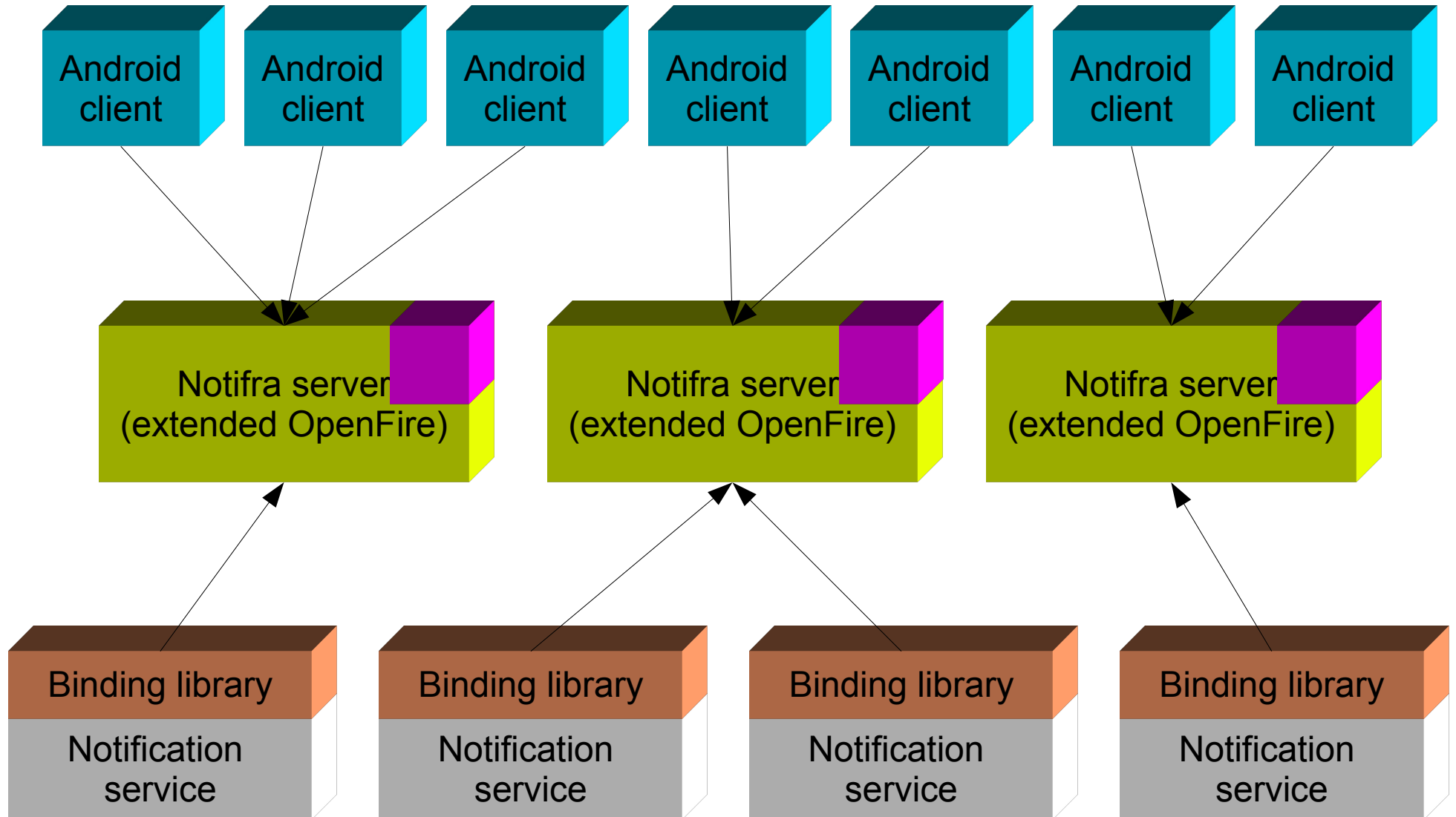
- Notifra is a complete set of tools to write real-time notification software (eg. about earthquakes, new messages, server problems, ...) which is:
  - Scalable
  - Efficient
  - Reliable
  - Fast
  - Easy to use (*out of the box*)
- Backend is based on XMPP (Extensible Messaging and Presence Protocol), communication in XML
- Client is for Android platform
- Services may be written in any language, for any platform (PHP and Java bindings are provided as for now)
- It's free, it's open-source!

ARCHITECTURE

# Typical (bad) architecture



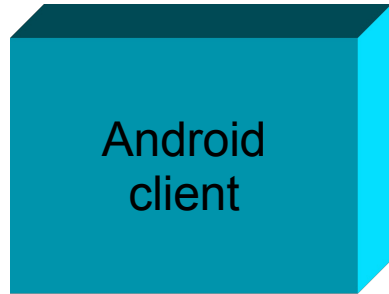
# Notifra distributed (good) architecture



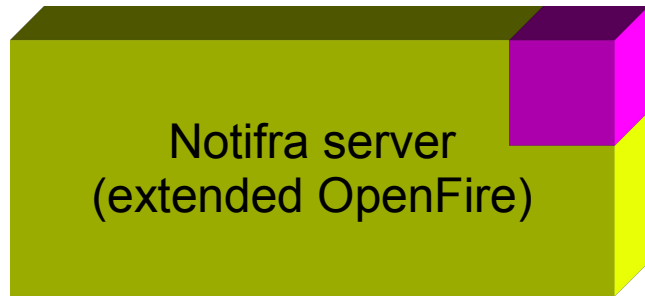
# Distributed architecture

- Distributed on each stage. There are no bottlenecks even for lots of subscribers and services
- Highly extendable and efficient
  - Anyone can add new server to this cloud
  - Anyone can write new services without starting own server and it will work for millions of subscribers
- Autonomous - does not require maintaining

# Components



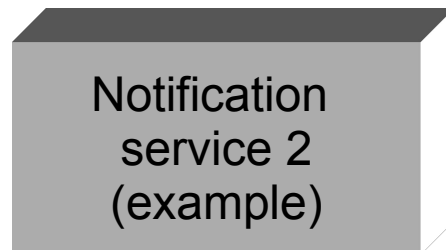
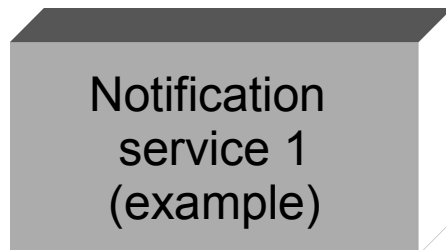
Client layer



Backend layer



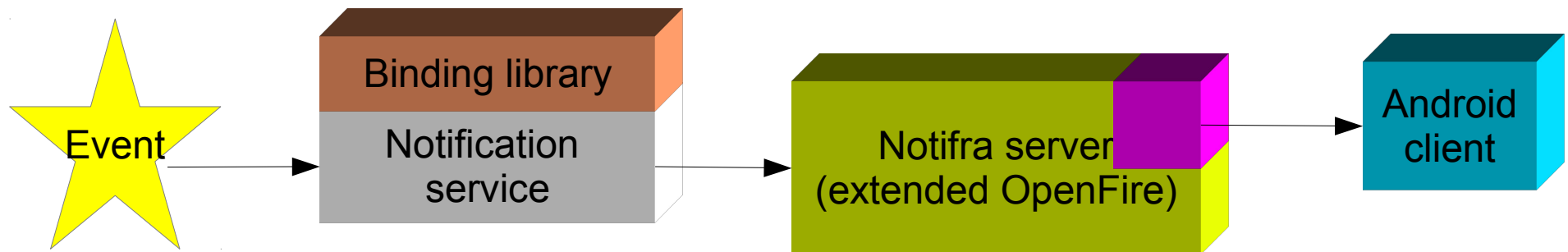
Middleware layer



Services layer  
(not part of the framework)

# How does it work?

- Android client connects to Notifra server
- Notification services do the same
- However, they do not have to be the same servers, therefore there is no bottleneck
- When a notification service detects an event:
  - It emits a notification and sends to Notifra server
  - Notifra server plugin forwards it to all subscribers
  - It is displayed on Android screens

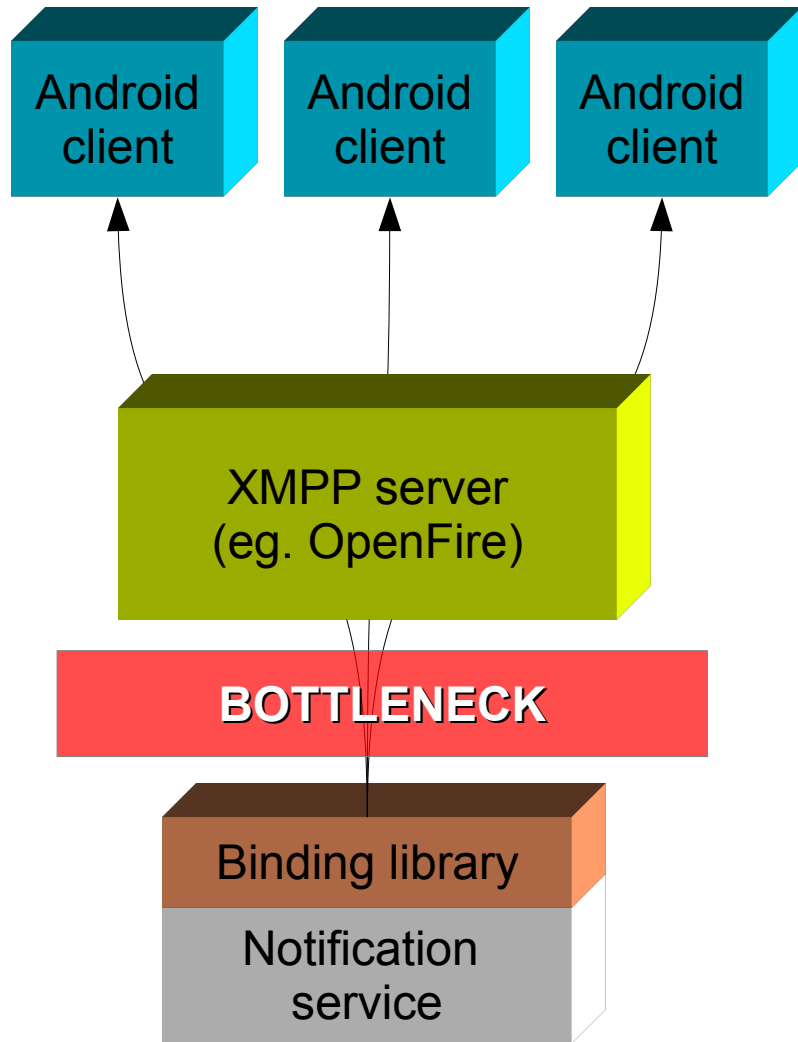




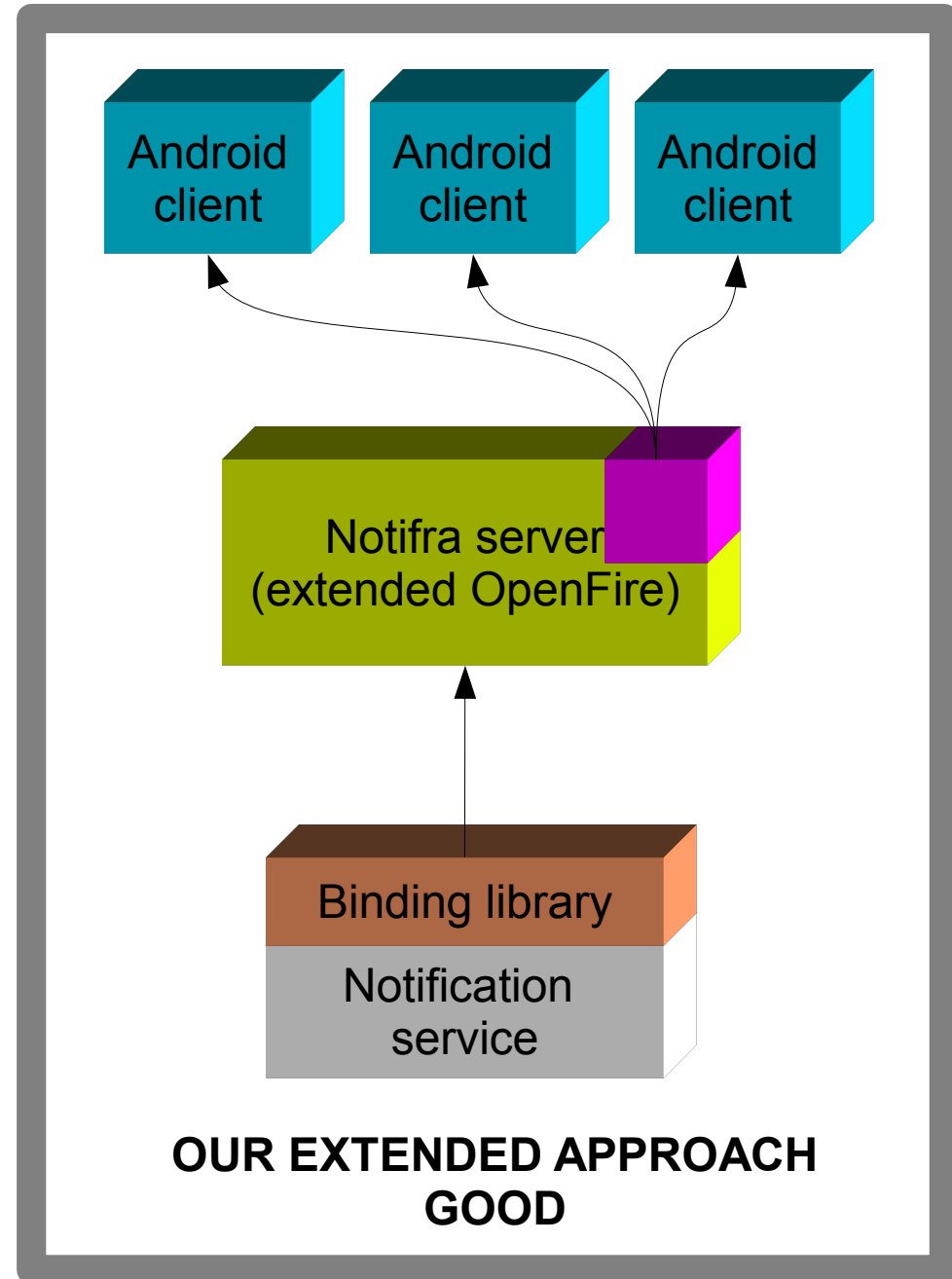
# OpenFire Notifra Extension

- It is an extension to open-source OpenFire XMPP server
- **Without it:**
  - Service has to send notification message to each subscriber (eg. for 100000 users). Bottleneck.
  - It is not scalable
- **With it:**
  - Service has to send only one message to server, which populates the message to all subscribed clients.
  - It is scalable

# OpenFire Notifra Extension

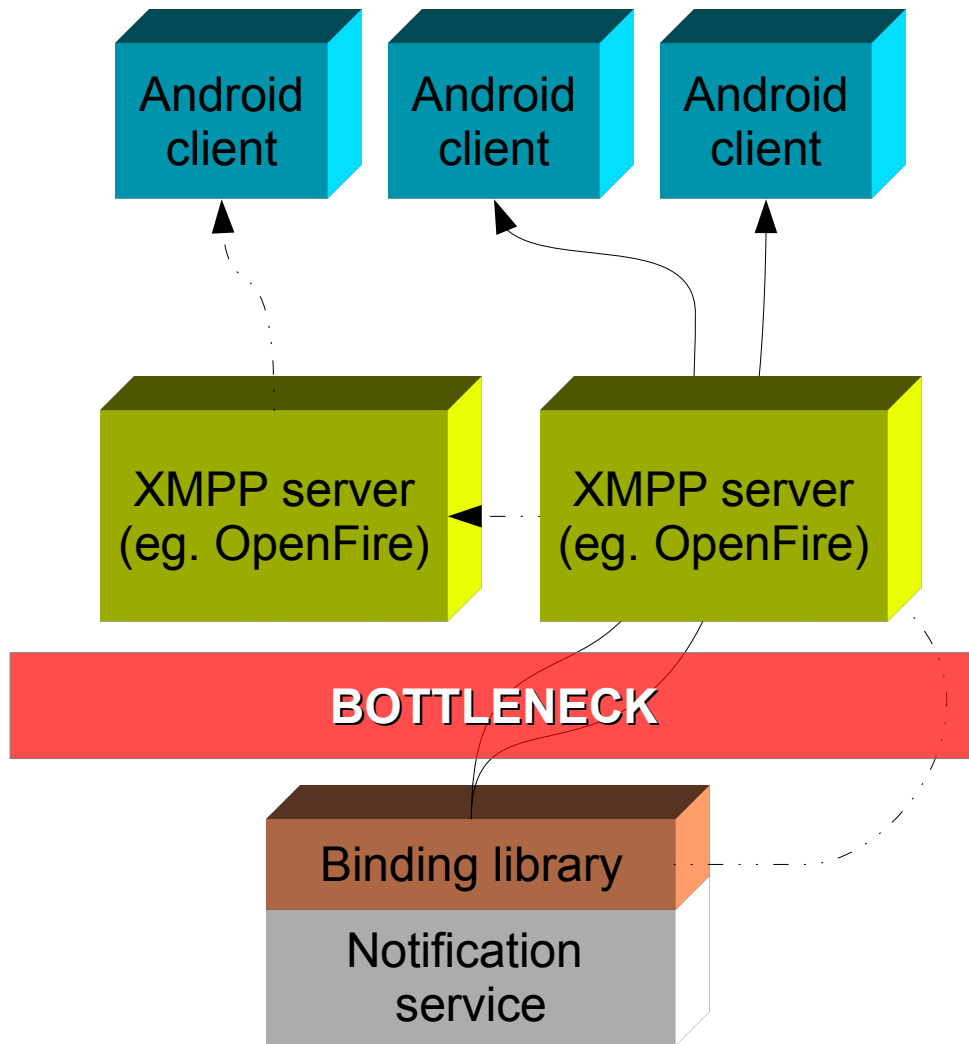


**TYPICAL XMPP APPROACH  
BAD**

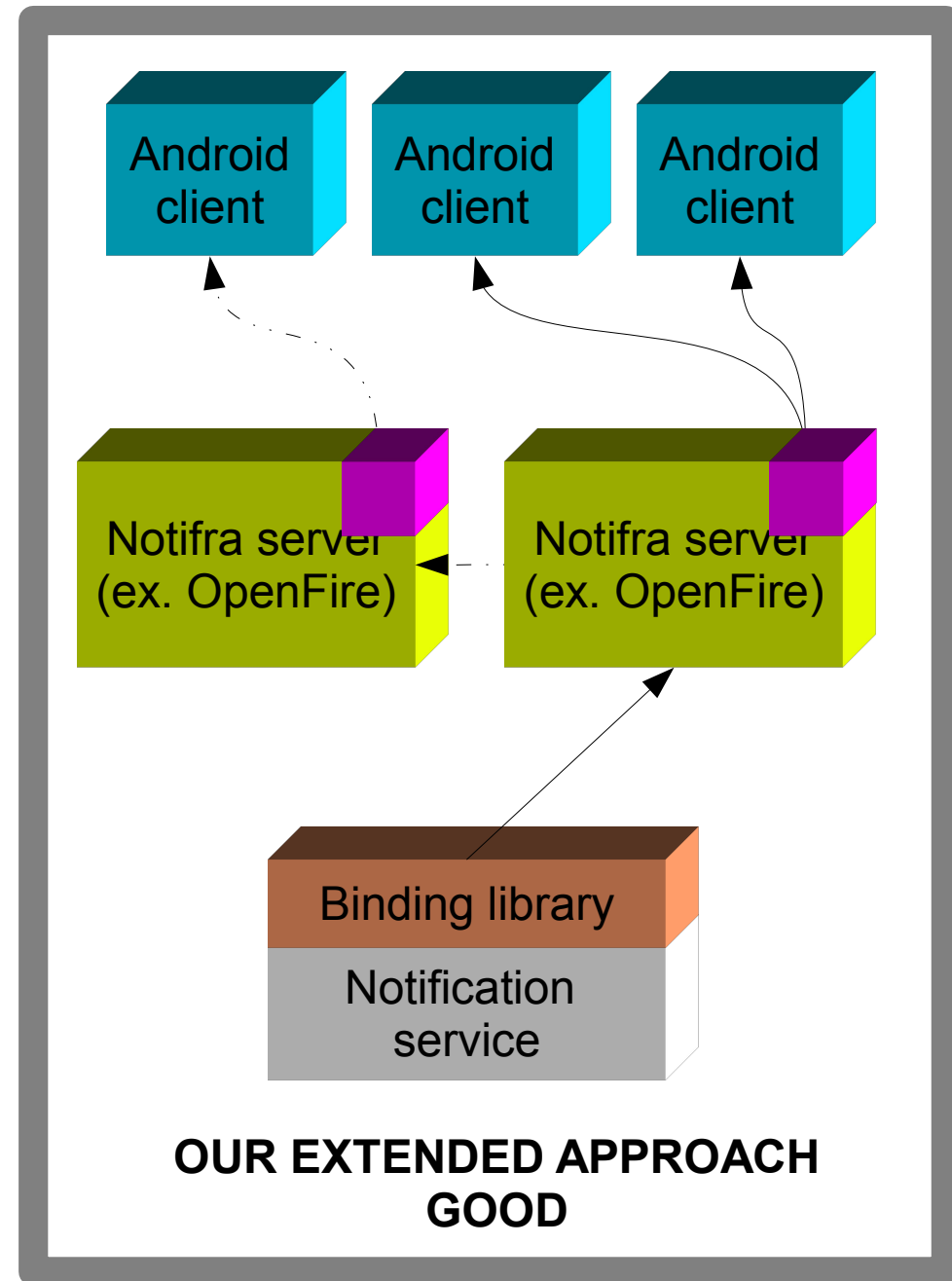


**OUR EXTENDED APPROACH  
GOOD**

# OpenFire Notifra Extension



**TYPICAL XMPP APPROACH  
BAD**



**OUR EXTENDED APPROACH  
GOOD**

# Comparison

Topic	Typical approach	Notifra approach
<b>Client for Android</b>	One for each service. Consumes memory and CPU. It is non consistent.	One for all services. Efficient, stable and consistent.
<b>Scalability</b>	Very difficult and inefficient.	Extremely simple and efficient.
<b>Easy to implement new services?</b>	No. There is no framework, so it has to be written from scratches every time.	Yes. With provided middle-ware bindings, requires just 2 lines of code.
<b>Traffic at Android site</b>	High for pooling. If binding available, then low. In both cases lots of active connections.	Always low. Only one connection.
<b>Traffic at service site</b>	Very high. For milion of subscribers, milion of messages have to be sent.	Very low. Only one message to server.
<b>Service requirements</b>	Strong hosting to handle traffic.	Any hosting, even free.
<b>Server requirements</b>	Not required.	Linux or Windows server. Can be shared between thousands of services.
<b>Lag</b>	For pooling it may be from seconds to minutes. For binding real-time.	Always real-time. Lag in milliseconds.

# How to start?

- **For an end-user:**

- Install Notifra Android application
- Add service you are interested (type-in identifier)

- **For a service developer:**

- Create account at free Notifra server, eg. [www.jovislab.com/notifra](http://www.jovislab.com/notifra)
- Download middle-ware for PHP or Java
- Send notifications using 2 lines of code:

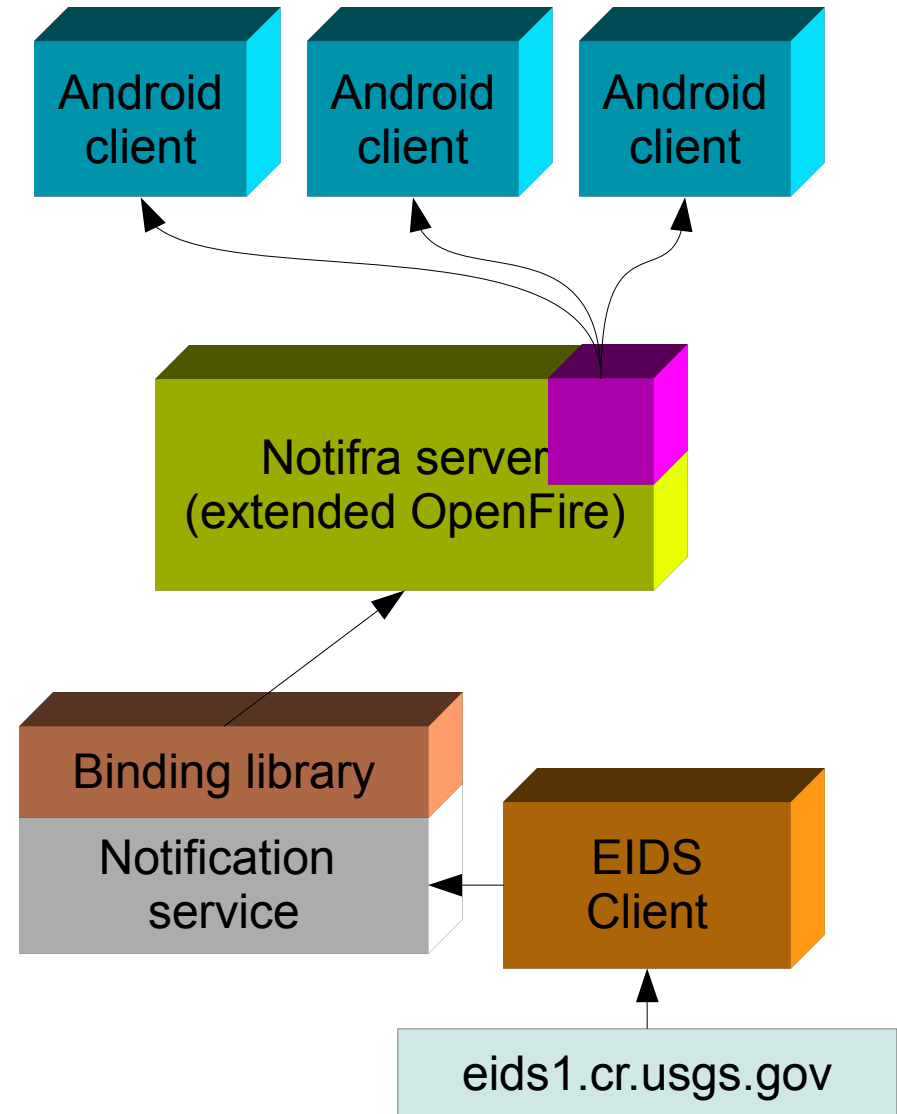
```
Notifra.connect(username, password);
```

```
Notifra.sendNotification([Your notification]);
```

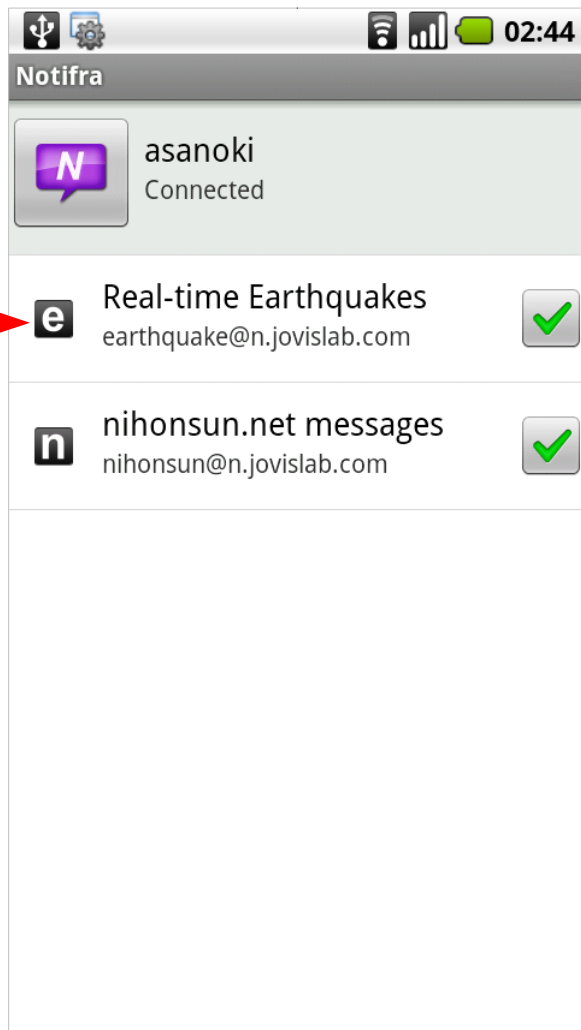
**DEMONSTRATION**

# Example 1 - Earthquake notifier

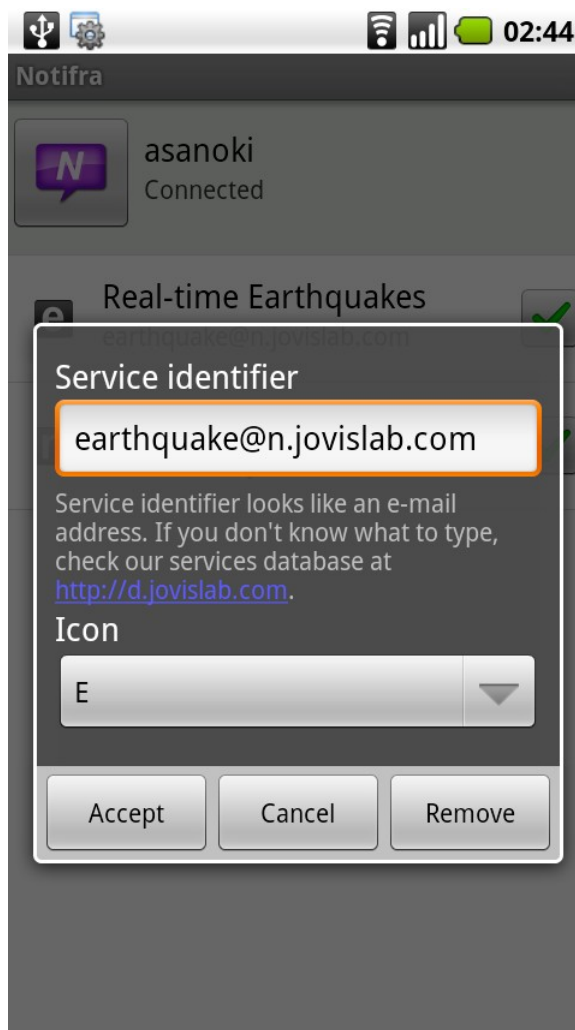
- Receives messages from USGS (U.S. Geological Survey)
- Uses EIDS Client by USGS
- Notification service (in Java) looks for Earthquake-related messages and forwards them to Notifra Server



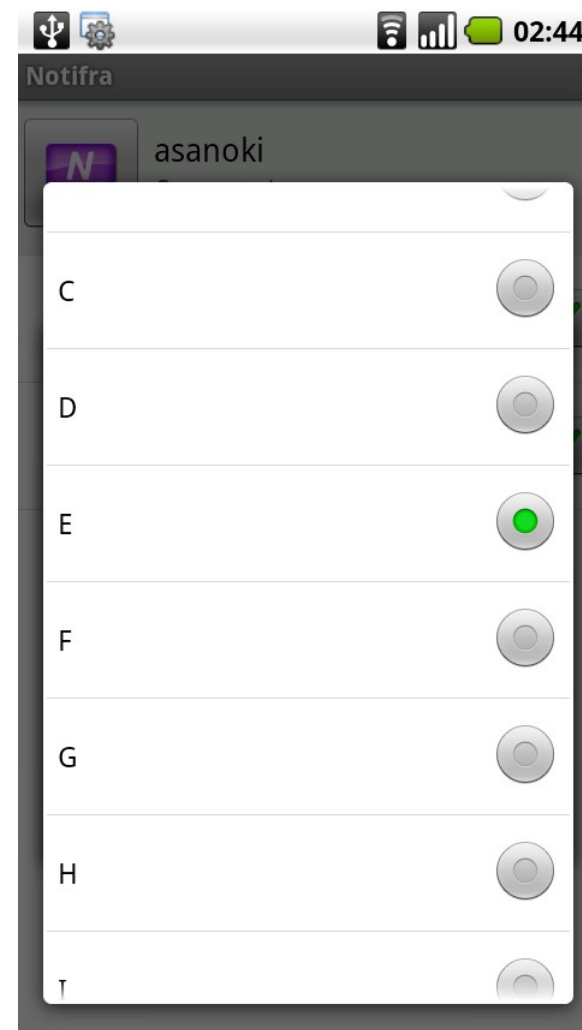
# Example 1 - Earthquake notifier



List of all subscribed services



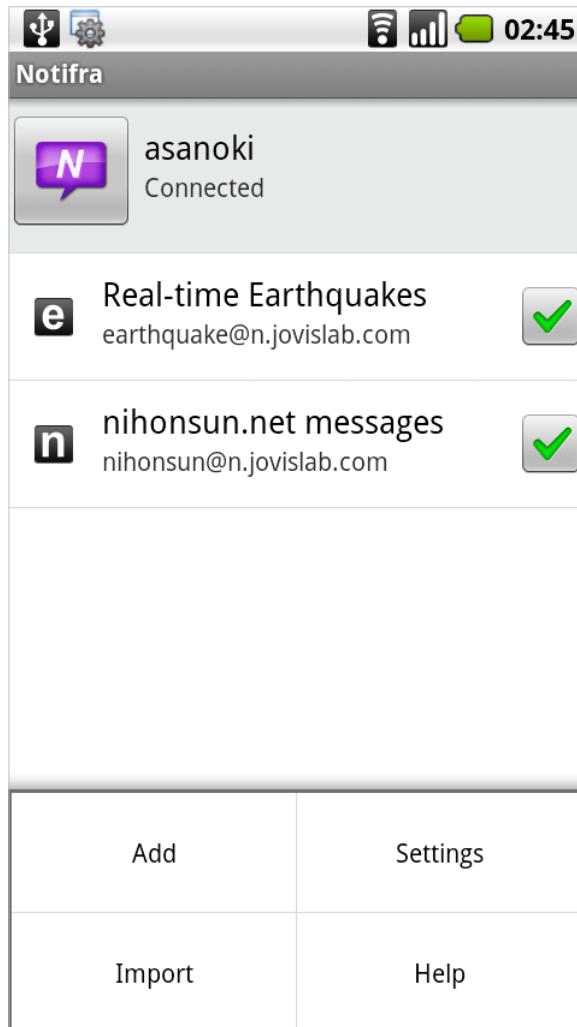
Service configuration



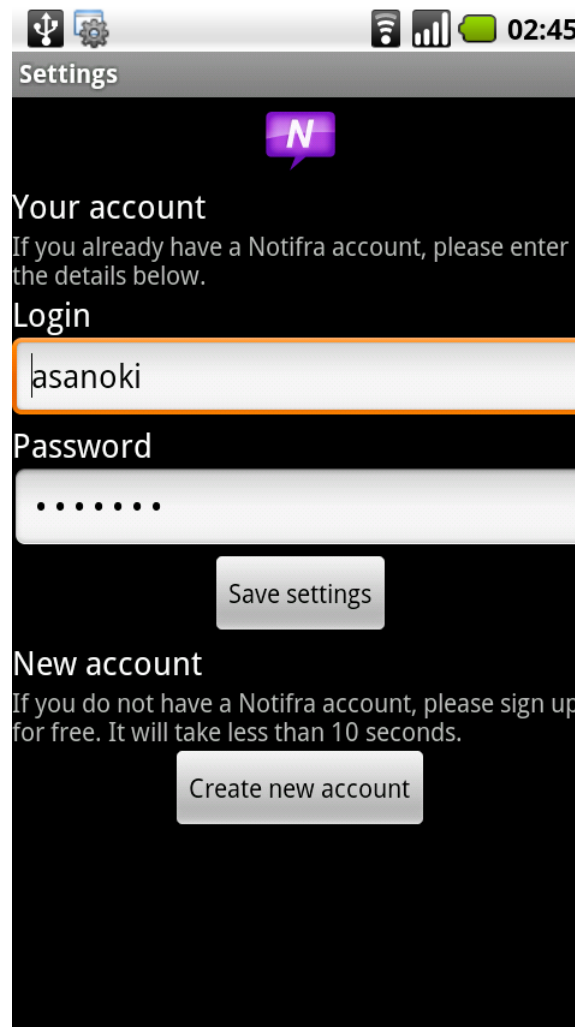
Notification icon choice



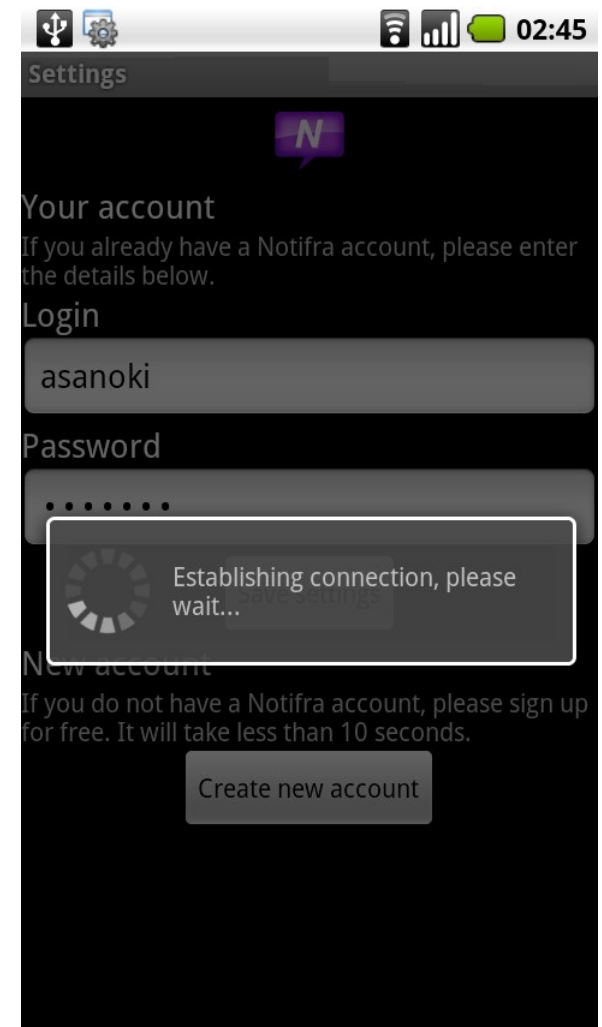
# Example 1 - Earthquake notifier



Menu

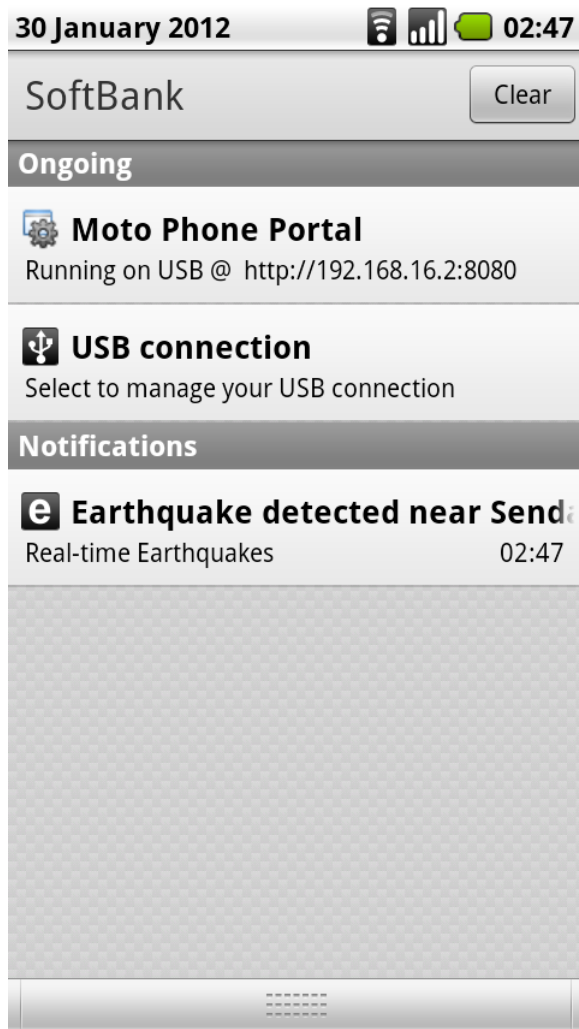


Account configuration

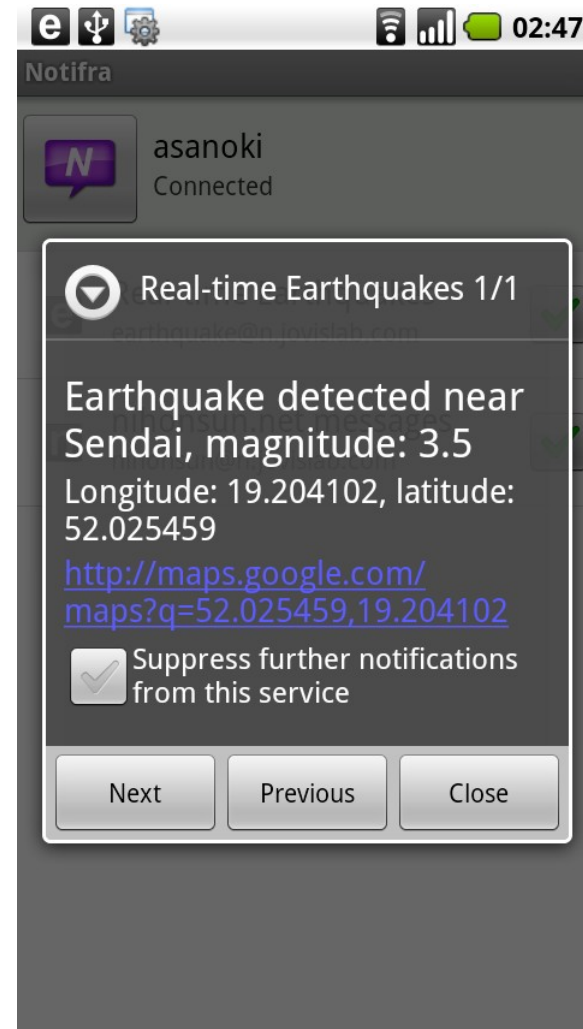


Auth data validation

# Example 1 - Earthquake notifier



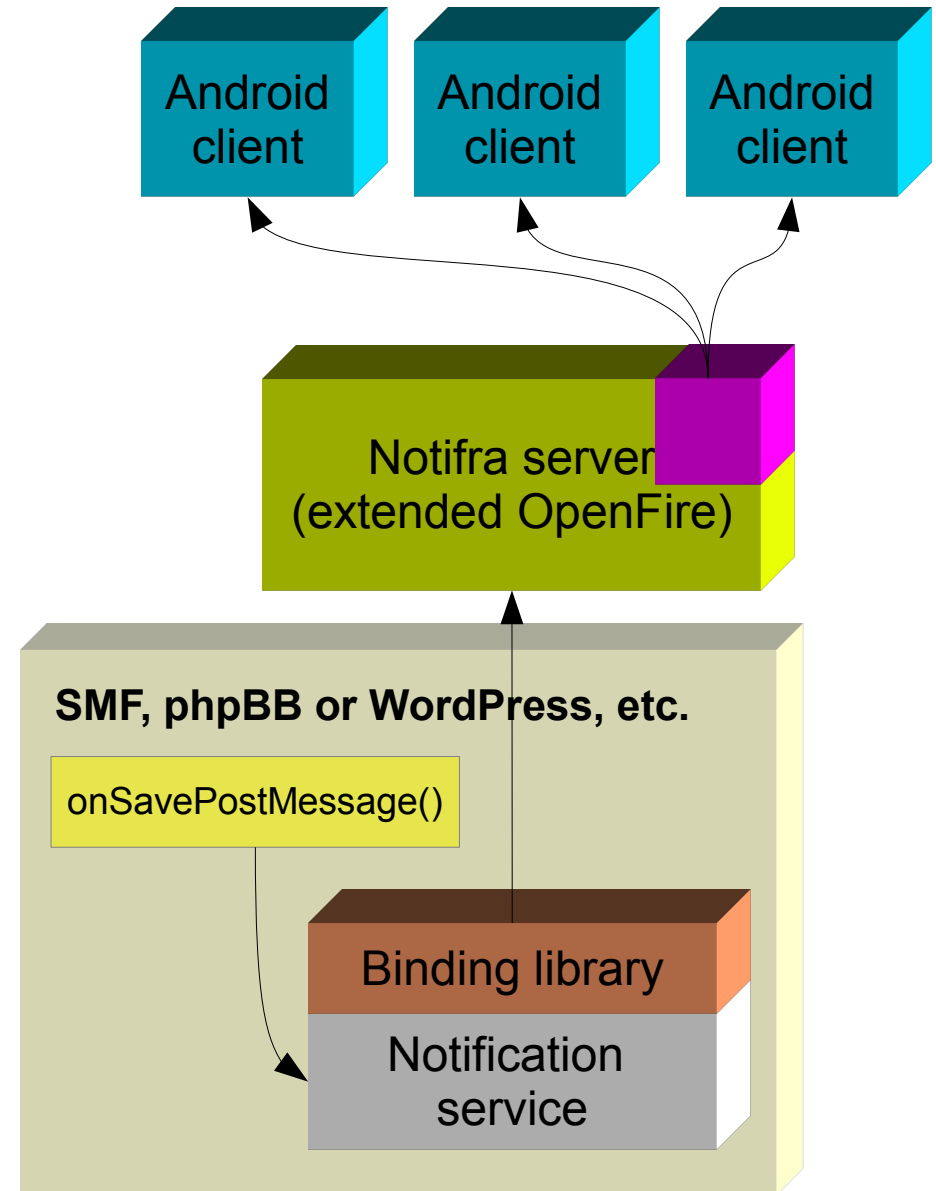
Earthquake has been detected



Notification details

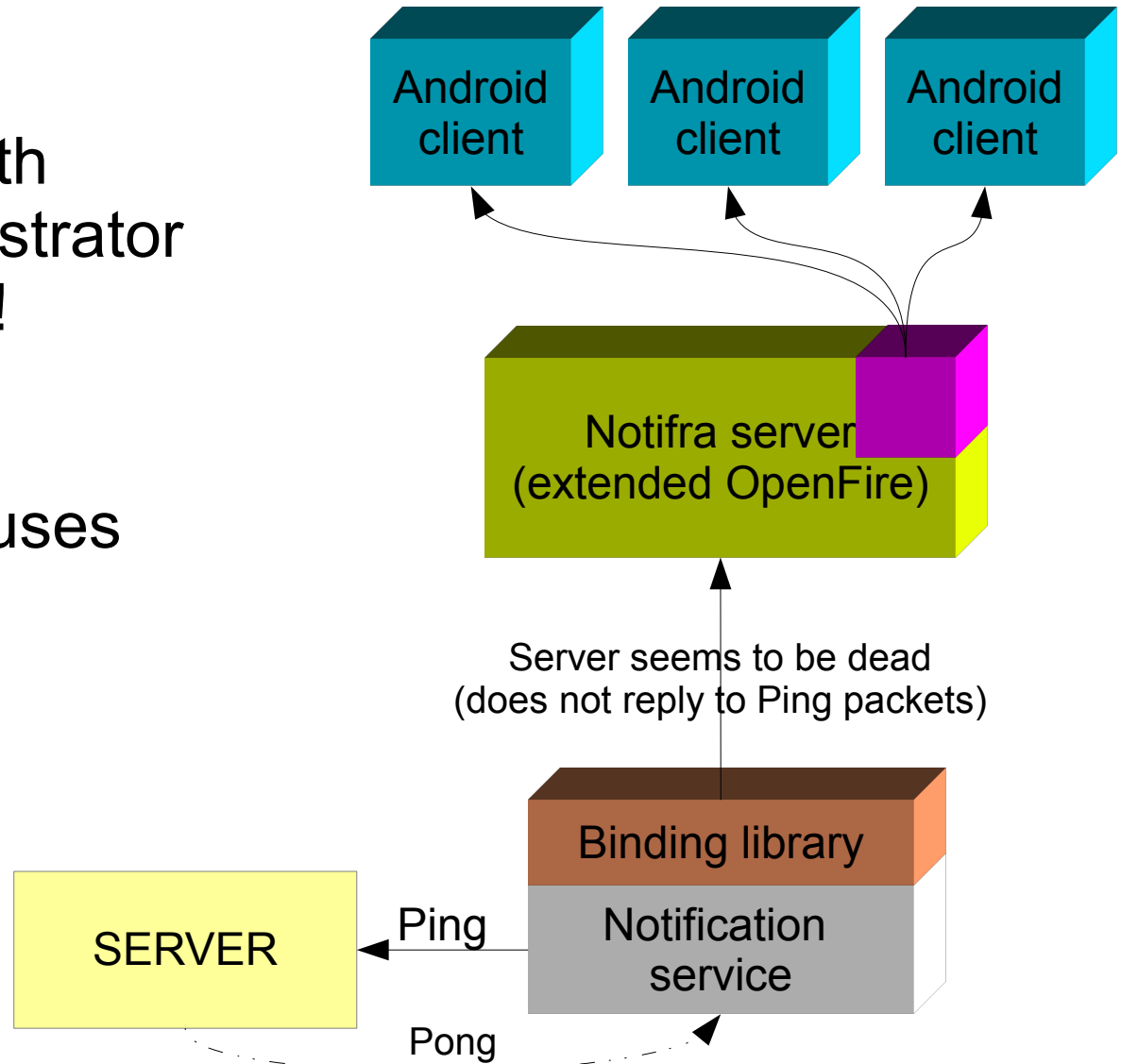
# Example 2 - New message notifier

- Idea: When someone posts a new message or comment, we get a notification in real-time!
- Requires integration with website scripts
- It is extremely easy
- Notification service uses binding in PHP



# Example 3 - Server monitor

- Idea: When there is something wrong with server, notify administrator as soon as possible!
- It is quite easy
- Notification service uses binding in Java



# Any questions?

Tomasz Mikolajewski

asanoki@gmail.com

asanoki@sc.isc.tohoku.ac.jp



jovislab