

Swiss Sense Synergy

SUPSI

07 October 2015

Michela Papandrea

Silvia Giordano

Luca Luceri

Alan Ferrari

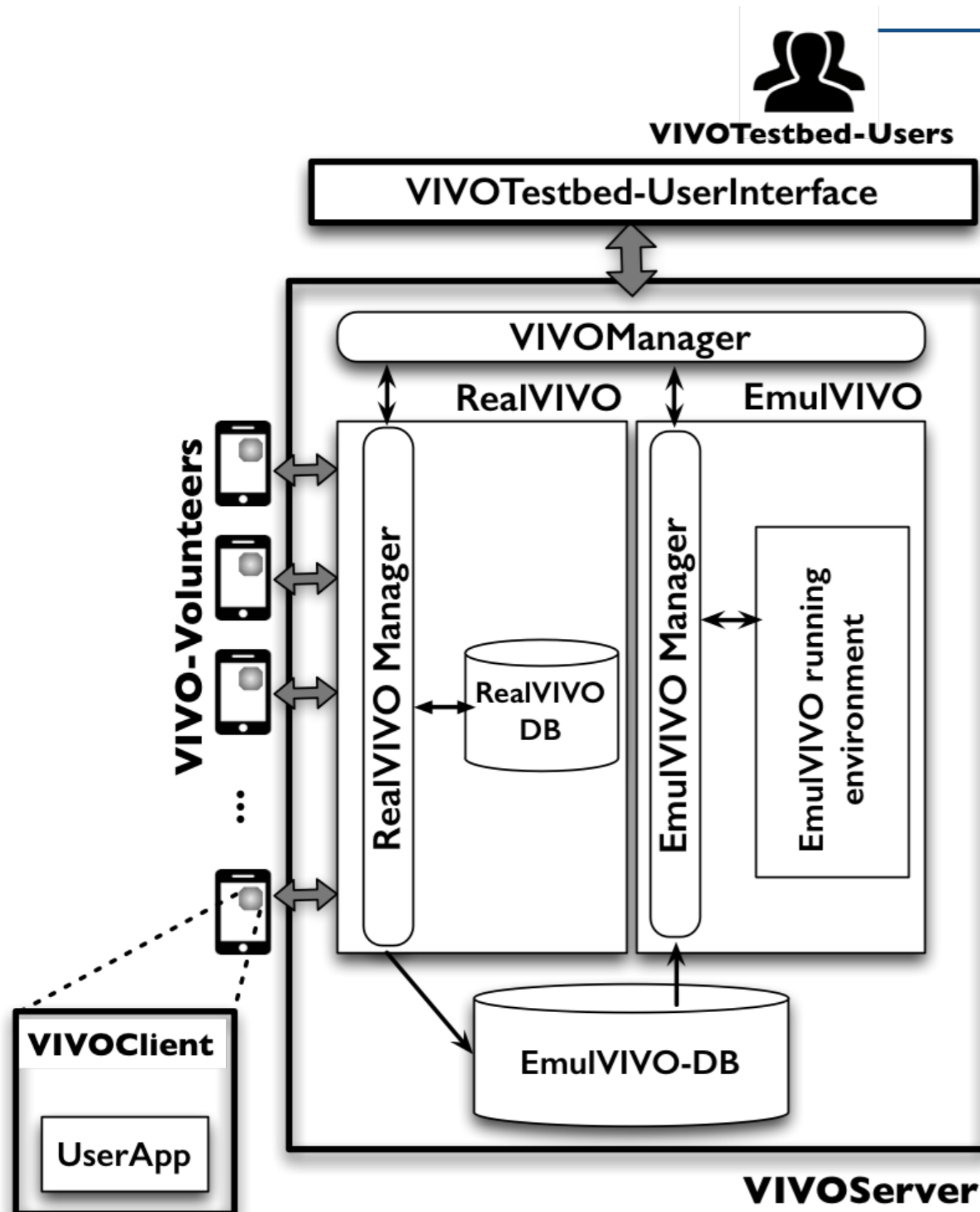
Dario Gallucci

Kamini Garg

Steven Mudda

VIVO TestBed Implementation

VIVO Architecture



Currently working on:

- VIVOTestbed Web UserInterface;
- a first version of the RealVIVO Server;
- RealVIVO DB
- VivoApp application to be run on Volunteer smartphones;
- 2 sample Experimental Applications.

VIVO Server

Specifications:

- Web Server: **Apache**
- Scripting: **Phyton** (*Flask framework*)

Server Tasks:

- User registration and login (Server web)
- Volunteer registration and login (JSON Web Interface)
- Experiment creation by the user and upload of the apk (Server Web)
- Push Experiments on Volunteers' devices (Server Web)
- Creation and Synchronisation of a couchDB on VIVO Server per each established pair Volunteer-Experiment.

VIVO Web Interface

User Registration

86.119.34.53/vivoapp/signup

Register to Vivo App

test2@supsi.ch

test2@supsi.ch

.....

.....

Register

Forgot your password, instead? [Click here to reset it.](#)

User Login

86.119.34.53/vivoapp/login

Login to Vivo App

test1@supsi.ch

.....

Remember me on this computer

Login

Forgot your password? [Click here to reset it.](#)

New user? [Click here to register.](#)

86.119.34.53/vivoapp/experiments

Experiments list for test1@supsi.ch

[Create New Experiment](#)

Experiment Name	Application Id	Category Use Environment
Acceleration Sensor Test	d50288b7-72be-49a3-9de1-f7c3a28cc2bd	False

← → ↻ 86.119.34.53/vivoapp/experiments/add

Apps Android Overview - AutoPlay

Create New Experiment

AccelerometerApp_test

All Users ▾

Use environmental data

Submit

Experiment creation

- *optional selection of the category of Volunteers preferred for running the experiments (sedentary users, always travelling users, routine users, ...)
- selection of the interest in **environmental data**

← → ↻ 86.119.34.53/vivoapp/experiments/

Apps Android Overview - AutoPlay

Experiments list for test2@supsi.ch

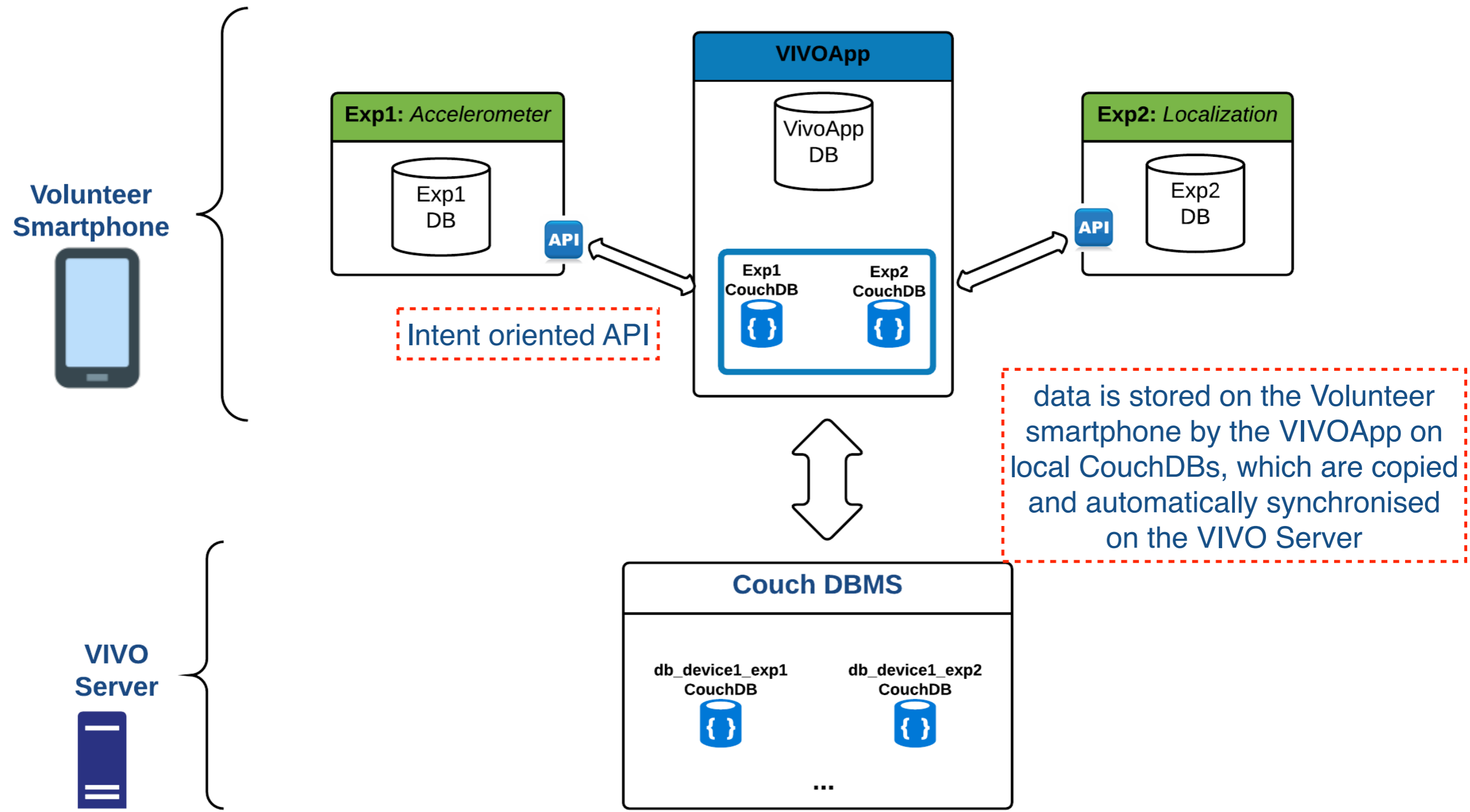
[Create New Experiment](#)

Experiment Name	Application Id	Category	Use Environment	Commands
AccelerometerApp_test1	5f72d954-9418-488d-910f-35c91a0d5d6d		False	Edit Upload APK Push

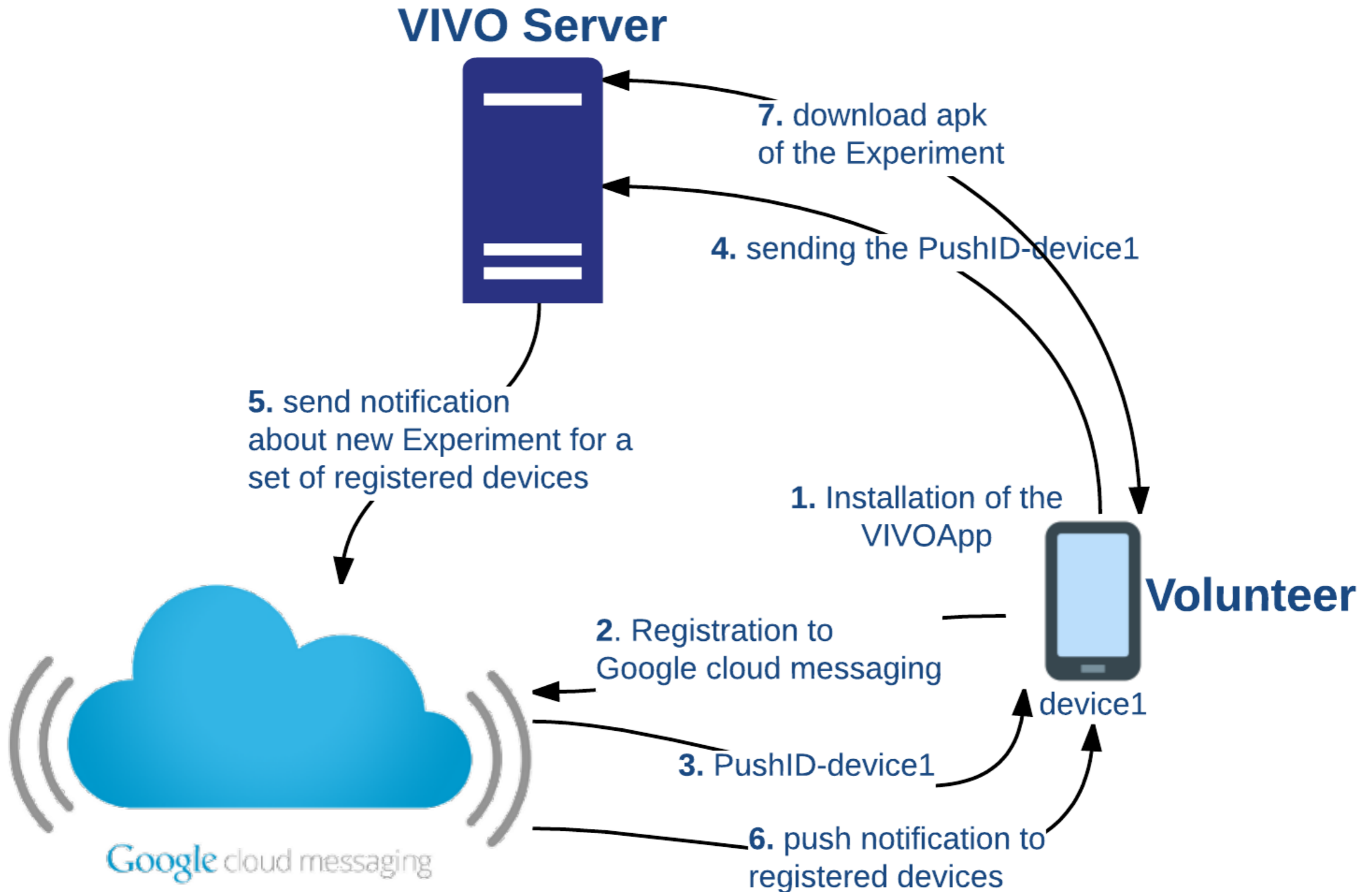
Experiment development requirements

- When a User registers an Experiment, he gets an ApplicationID (`expid`) from the VIVO Server. This ApplicationID has to be hard-coded into the application code. Successively the apk of the Experiment can be uploaded.
- The Experiment has to include the VivoApp API and to use it to store the data which the User need to get back. The data can be stored in any format (value-pairs, hashmap, ...)

VIVOApp API: data synchronisation

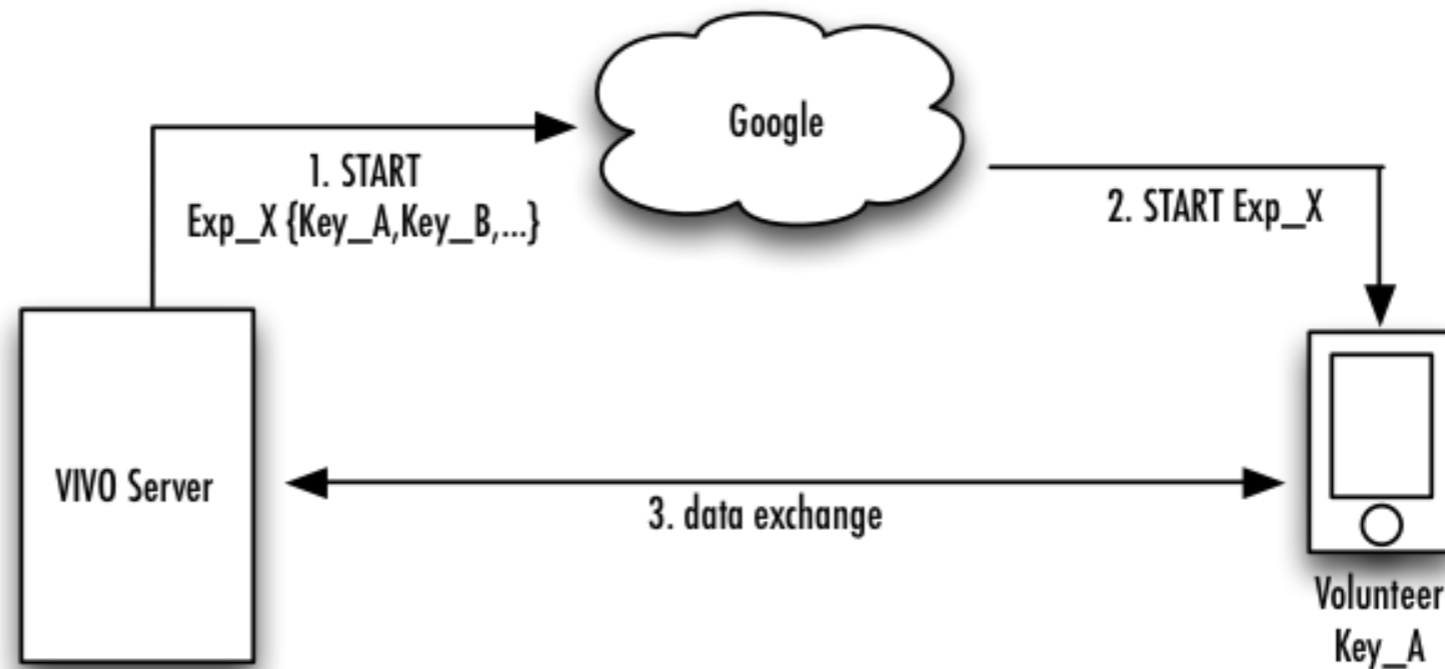


Push Notification for new Experiments



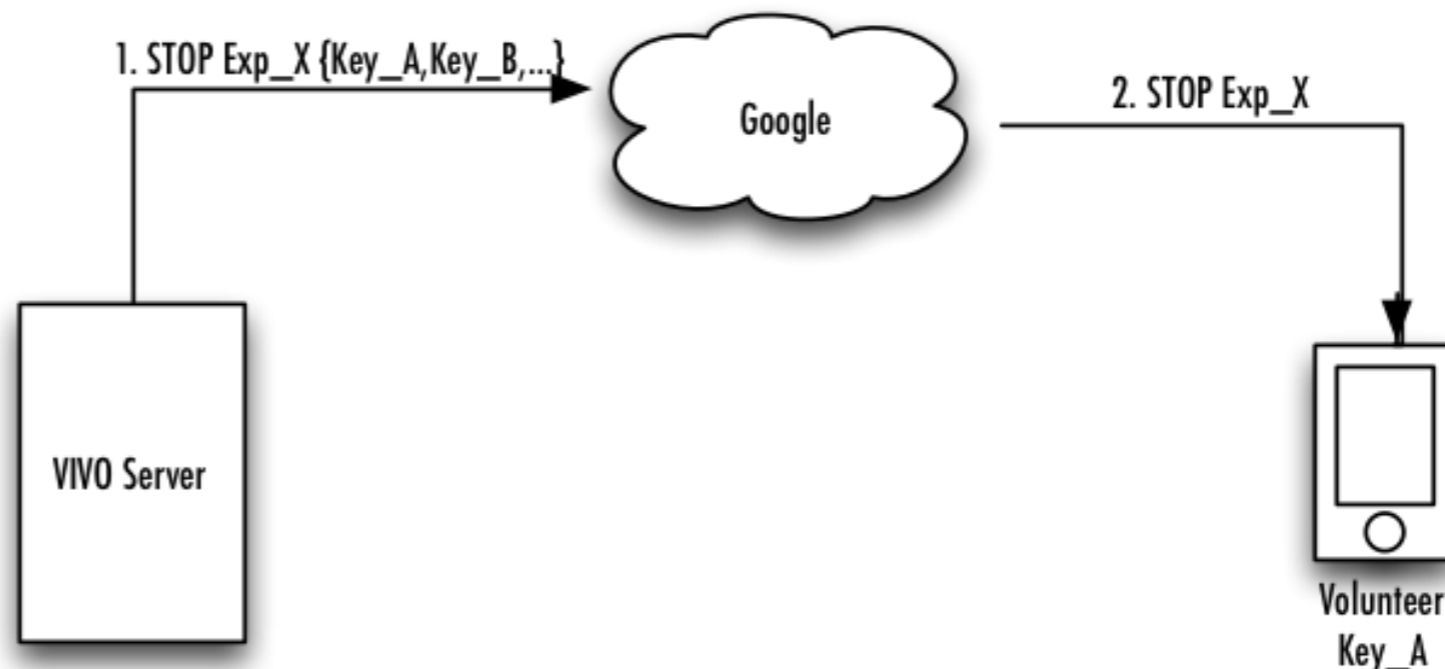
PUSH notification for an experiment

START notification

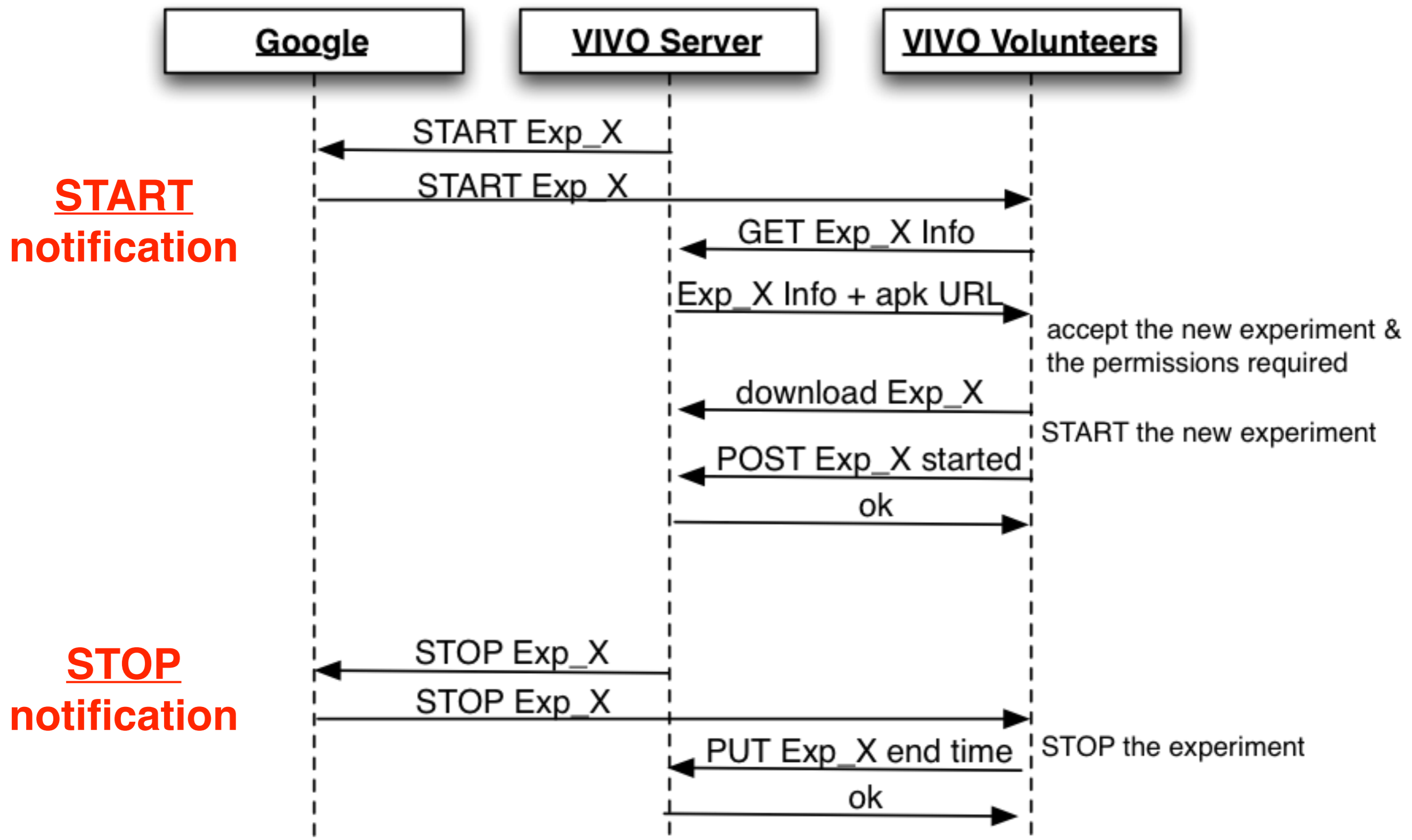


- 1.2. —> pushing a notification to the Volunteer for a new experiment available
- 3. —> VIVOApp downloads the apk of the new experiment from the VIVO Server (the link of the apk's is sent through notifications 1.2.)

STOP notification



PUSH notification for an experiment



- **PosgresDB:** DB used by the *VIVOWebApplication* to store information about *RegisteredUsers*, and by the *JSonInterface* to store information about *RegisteredVolunteers*.
- **Couch DBMS:** there will be a CouchDB database for each established pair *Volunteer-Experiment* (volunteer accepting to run an experiment).

Experiments (running on Volunteers smartphones) will never connect to the Couch DBMS. It will be the VIVOApp which will perform the synchronisation with the CouchDBs (containing the Experiments' collected data) on the VIVOServer.

VIVO CouchDBMS

Overview

+ Create Database ...

Name	Size	Number of Documents	Update Seq
_replicator	4.1 KB	1	1
_users	12.1 KB	3	3
db_f41d06ad91713a6e_d50288b7-72be-49a3-9de1-f7c3a28cc2bd	8.1 KB	1	1
vivotestautosync	272.1 KB	1	64

Showing 1-4 of 4 databases

← Previous Page | Rows per page: 10 | Next Page →

db_<devID>_<expID>

devID = Android secure ID (128bit) generated automatically on each Android device

expID = generated by VIVOserver when the User registers the Experiment through the VIVOwebInterface. It should be hard-coded into the Experiment Application, before it is uploaded on the VIVOserver

Tools

- Overview
- Configuration
- Replicator
- Status

Documentation

- Manual

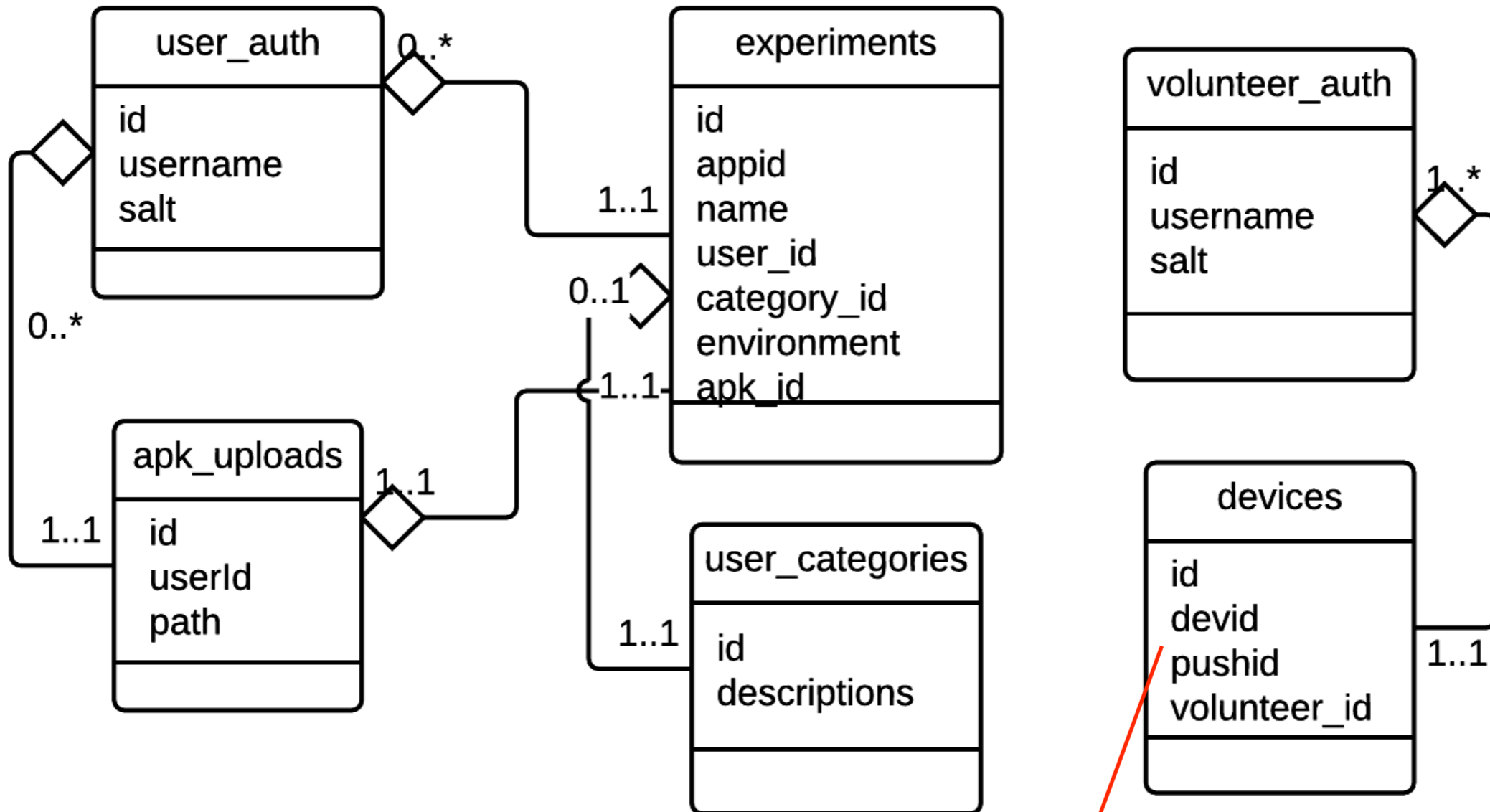
Diagnostics

- Verify Installation

Recent Databases

- db_351565052494546_...

PostgresDB



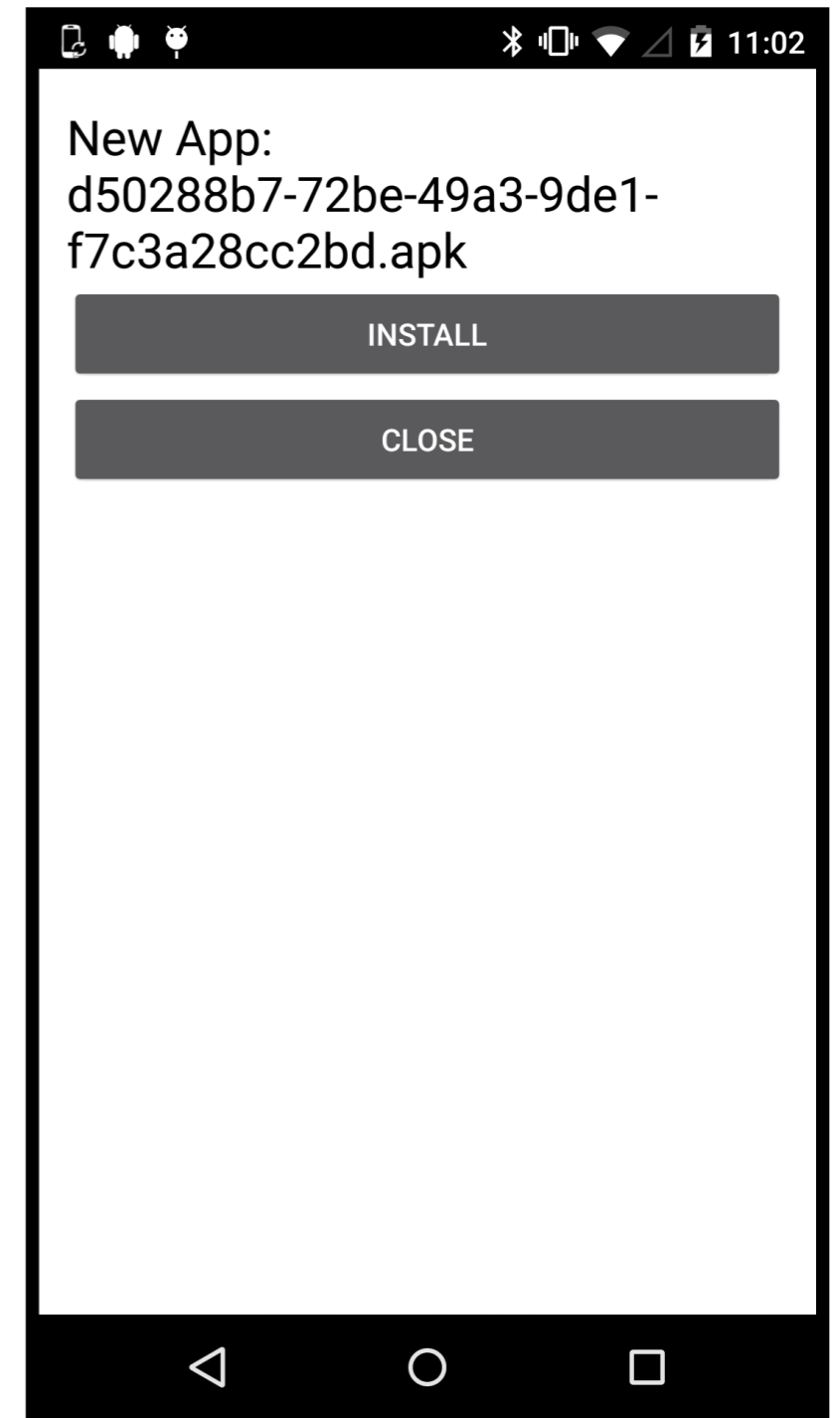
devid: Android secure ID of the *device*, generated automatically on each Android device (128 bit).
pushid: id associated to the *VivoApp* after the registration to the Google cloud messaging, for the *push notification* of new Experiments.
volunteer_id: email of the user, for the identification of the volunteer at the registration/login.

VIVO App

Volunteer registration

The Register screen features a title 'Register' in blue. Below it is the instruction 'Here you can create a new volunteer account'. There are three input fields: the first contains 'demotest@supsi.ch', the second also contains 'demotest@supsi.ch', and the third contains six dots. At the bottom, there are two buttons: 'Cancel' on the left and 'Register' on the right.

Notification about a new Experiment



Volunteer login

The Login screen features a title 'Login' in blue. Below it is the instruction 'To use this application you need to login. Please register if you do not have an account.' There are two input fields: the first contains 'testv@supsi.ch' and the second contains six dots. At the bottom, there are three buttons: 'Cancel' on the left, 'Register' in the middle, and 'Login' on the right.

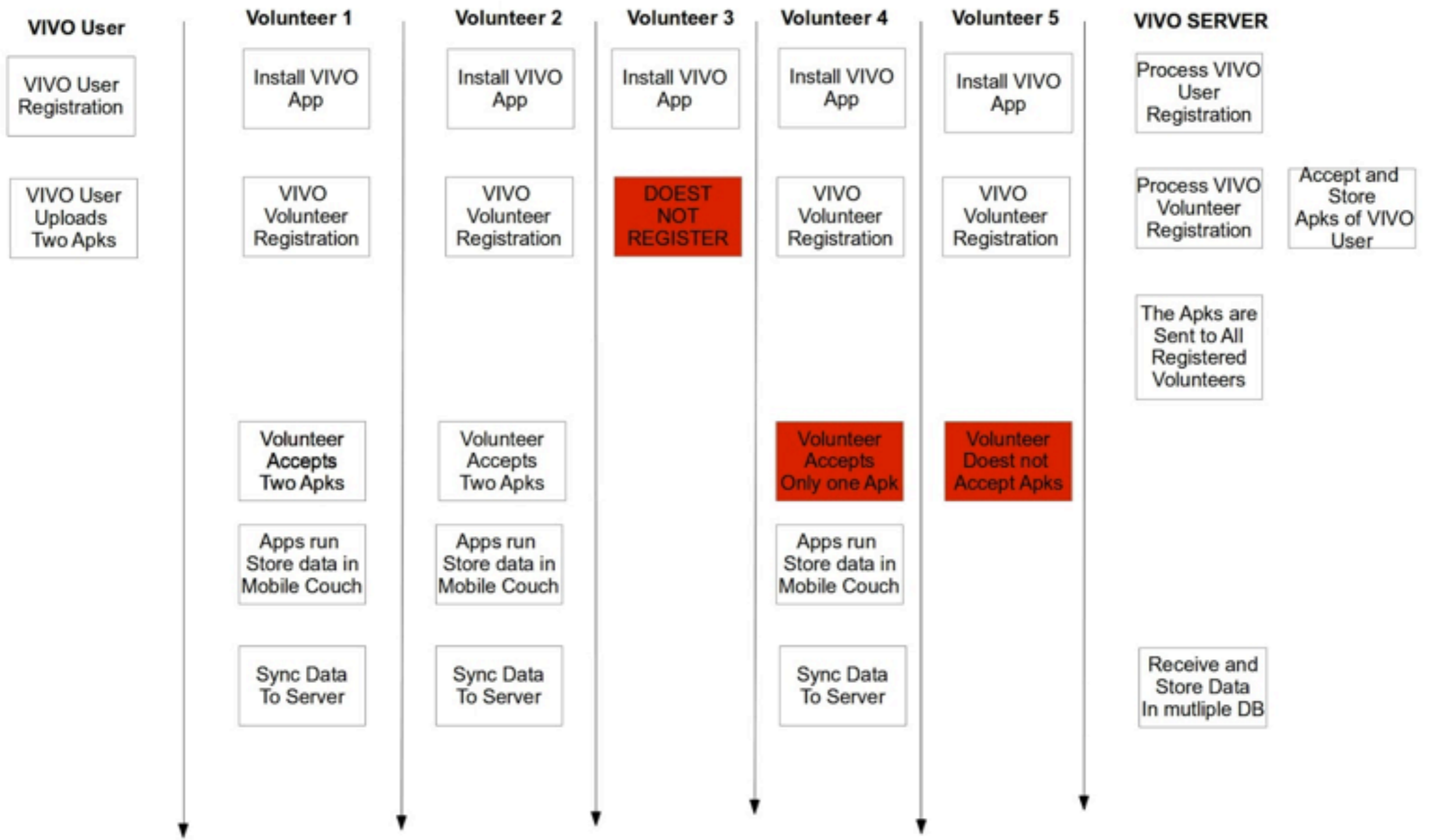
VIVO Experimental Applications

We created two sample Experimental applications for testing purposes:

- **Accelerometer Reader Application**: application which reads data from the smartphone embedded accelerometer (frequency SENSOR_DELAY_NORMAL) and stores it locally.
- **Location Finder Application**: application which reads localisation updates (maximum allowed frequency) from the Android GPS and NETWORK Location Provider and stores the data locally.

All experimental applications have to use a **VIVO API** in order to store data on the VIVO Server (data which has to go back to the User). The data is not sent to the Server directly from the Experiment, but the *VivoApp* intercedes between the *Experimental App* and the *Vivo Server* by means of the VIVO API.

VIVO Demo



Open Topics

Data Security

- Data collected by volunteers are stored in **JSON format**, and are sent in this format to VIVO Server. How do we secure data? (collaboration Chalmers-SUPSI, Differential Privacy Algorithm proposed by Chalmers)
- Data connection security (TLS?)
- Data Anonymisation

Optional Volunteer Information

When the Volunteer registers to the VIVOApp we might ask some personal information, or retrieve them automatically, in order to build some statistics and to perform a smart filtering of the volunteers when pushing new available Experiments:

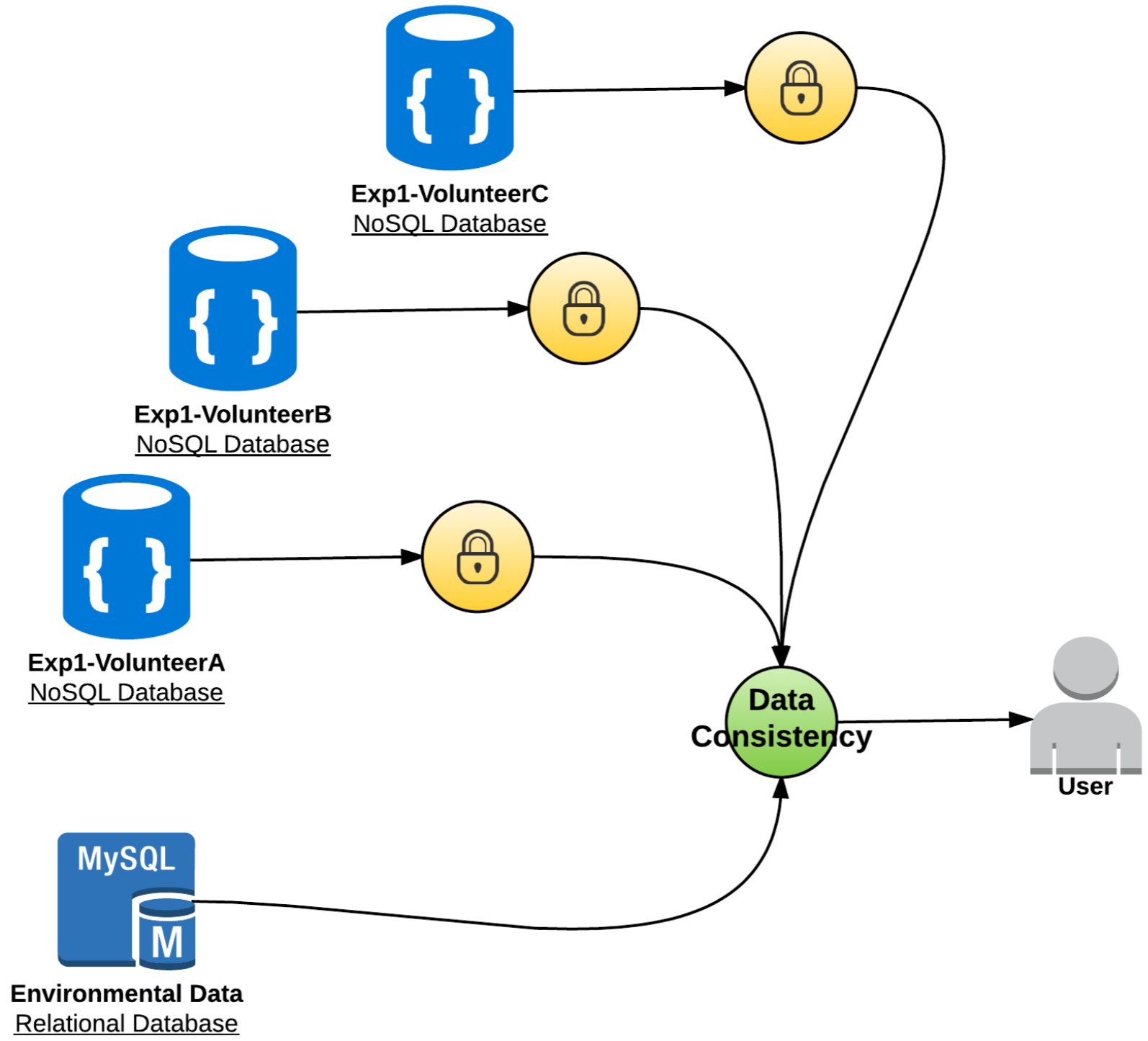
- (?) country of the SIM card or location at login (for environmental sensors proximity)
- (?) gender
- (?) year of birth
- (?) working during day or night
- (?) usual kind of mobility (none, by foot, bike/motorbike, public/private vehicle)
- (?) user mobility style (person-of-habit, globe-trotter)

Classes of Experiments

Do we need to define the class of experiments to be run on VIVO TestBed?

- sampling mobile device embedded sensor (i.e., Exp1 Accelerometer)
- sampling Location data (i.e., Exp2 LocationFinder)
- (?) Survey
- ...

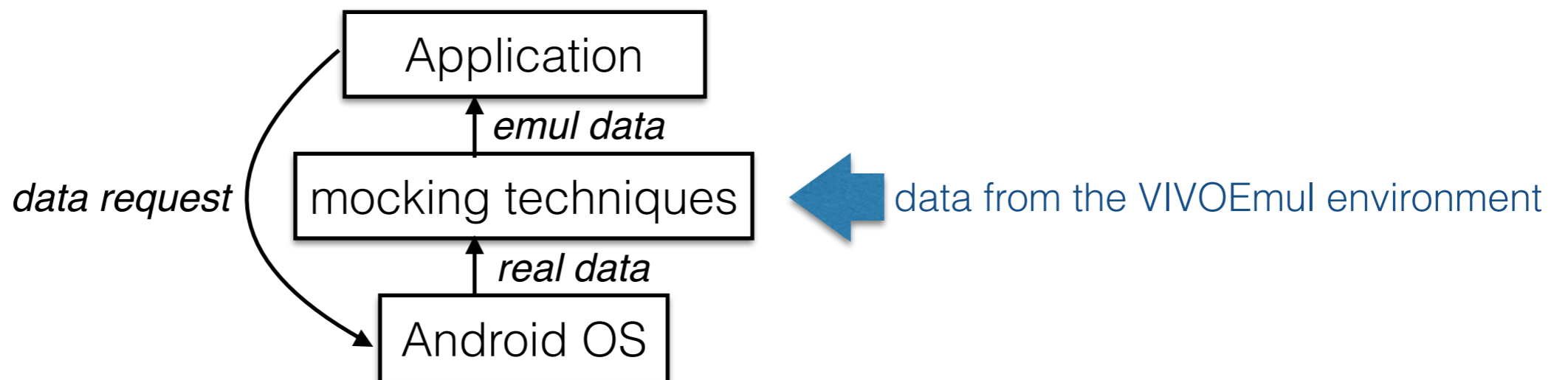
Integration with environmental sensors data.



VIVOEmulation

Some options for the creation of VIVOEmulation environment:

1. We create a dedicated VIVO Experimental App to collect data for the VIVOEmulation environment.
2. VIVOApp collects silently the data which will be used for the VIVOEmulation environment (i.e., location PASSIVE_PROVIDER).
3. We exploit the data collected by the various Experiments to create the VIVOEmulation Environment. In this case, we need to define a DATA FORMAT for the Users + the User has to specify if she/he wants to share the collected dataset.



Volunteers Motivation Campaign

- ~100 students in China
- Users have to provide Volunteers
- Classes (SUPSI, UniGe, UniBe, Chalmers)
- others