INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Project Assignment for Ubiquitous and Mobile Computing

MEIC/METI 2018-2019 – 2nd Semester

P2Photo

1. Introduction

A typical photo-sharing mobile application requires full trust in the application provider servers for maintaining users' photos and managing group membership. The downside of this web server architecture, however, is a loss of privacy, in the sense that the application provider can keep track not only of which photos are shared by the users, but also how they are accessed, by whom, and when. The goal of this project is to build P2Photo, a mobile application that allows users to share photos with their friends in a privacy-preserving way. This will be achieved by ensuring that the application provider will be involved only in maintaining group membership, i.e., allowing users to create new albums, finding new users, and adding or removing users from albums. In contrast, all the operations involving publishing photos in albums and reading photos from albums must be performed without the provider's awareness. Concretely, this means that the photos themselves must be stored and shared between users without the mediation of the provider. To this end, the project must support two architectures: cloud-backed (until the checkpoint) and wireless P2P (after the checkpoint).

2. Specification

2.1 Baseline Functionality

P2Photo is a mobile application that allows users to store photos on mobile devices without using centralized storage. Users can store photos, create albums, share albums with other users, provide photo storage for other users and use other users' devices for photo storage. The basic architecture of P2Photo relies on a central server and a client mobile application.

2.1.1 Mobile Application Functionality

The P2Photo client is a mobile Android application that users install and run on their devices and allow users to perform the following functions:

F1. Sign up.
F2. Log in/out.
F3. Create albums.
F4. Find users.
F5. Add photos to albums.
F6. Add users to albums.
F7. List user’s albums.
F8. View album.

The sign up operation (F1) allows users to create a new user account in the system; the user must enter a username and password. The client then contacts the P2Photo server, which must ensure that the new username is unique and adds the user to the user database. If the operation is successful, the user can then log in and start a new session on the client device.

To perform useful functions on P2Photo, the user must log into the system (F2) with his account credentials (username and password), which must be validated by the server. If they are valid, the server must generate a new session id, keep an internal record associating that session id to the username and return the new session id to the client. However, if the credentials are invalid, the server must return an error. Only users with valid session ids will be allowed to perform additional functions (F3-F8). The log out function (F2) ends the current session.

Once a session is active, users are able to manage photos and albums. Users can create an album (F4). This operation creates an album catalog file in the P2Photo server containing the URL of an album-slice catalog of the creating user. The album-slice is the part of an album that contains the photos contributed by a user to an album. An album-slice is stored in its owner’s cloud storage. It includes all the photos that user had contributed to an album and a text file (the catalog) grouping all the URLs pointing to those photos. For example, if Alice creates an album “Fun”, automatically there is a album-slice in Alice’s cloud storage (e.g. Dropbox account)

for that album “Fun” and a URL for that album-slice catalog. The album-slice URL points to an album-slice catalog which is a file in Alice’s cloud storage with a list of all the URLs of all the photos in Alice’s “Fun” album-slice.
P2Photo allows logged in users to find other users to create albums (F4). With the usernames returned by the server, a user can add those other users to the album membership (F6). Once a user is invited to an album, the application adds an album-slice to the album. If Alice invites Bob to join the “Fun” album, the catalog for that album (in the P2Photo server) will now include URLs for Alice’s album-slice catalog (in Alice’s cloud storage) and Bob’s album-slice catalog (in Bob’s cloud storage).

2.1.2 Cloud-backed architecture

In this version of the application the photos are maintained in storage space allocated on the cloud. When the user signs in to P2Photo for the first time, he associates the P2Photo account with an account on Dropbox, Google Drive, or similar cloud storage provider, which P2Photo will use for private storage. When a user publishes a photo on a given album, that photo will be stored on the user's private storage. In order to retrieve that photo, the other members of the group with access permissions to that album will be given a direct URL to the photo. This mechanism will allow members to publish and read photos without involving the P2Photo server. The P2Photo server only needs to maintain the metadata about group membership of each album as well as a list of all P2Photo users. That metadata includes a list of user ids, a list of each user’s albums and for each user’s album, a URL that points to a special file located in the user's private storage space. This special file, named catalog, contains the list of the URLs that point to the photos published by that user in that specific album. Thus, in order for a user to list all the photos available on a given album, the P2Photo mobile application needs simply to retrieve the album membership from the P2Photo server, and then, for each member, download its respective catalog, parse it, and download all the photos published by that member using the URL contained in the catalog. We assume that the application provider is not malicious and does not willingly retrieve the catalogs and respective photos from the users' private stores. It is assumed that photos cannot be removed, albums deleted or users removed from an album’s membership.

2.1.3 Wireless P2P architecture

In this version of the application, the photos are maintained in storage space allocated on album members devices and the photos are exchanged opportunistically in a P2P fashion between devices over WiFi Direct. The basic protocol is as follows. Each device manages local storage containing the photos published by the device owner and a catalog file which lists the set of photos associated with a given album. Whenever a set of co-located devices forms an ad-hoc wireless network, the devices broadcast their catalog files so as to inform all other members about the photos posted by their respective owners in each catalog. After assembling this information, each device has the necessary means to contact each device and download a copy of each photo and display it to the local user.

2.1.4 Server

The server is a web server which is in charge of maintaining the user list, managing album membership and maintaining the album catalogs. Students should start by implementing all the project using only the web server and cloud storage. Later, the solution should transition to the more advanced solution where metadata is stored on the server but files are exchanged based on WiFi Direct.

2.2 Advanced Features

In addition to the baseline features, students must also implement two advanced features (each worth circa 3 points) in order to get the full 20 points as described below:
A) Security: Design and implementation of a security mechanism to encrypt the catalog files and prevent a malicious application provider from retrieving users' catalogs and photos from their private stores.
B) Availability: Design and implementation of a replication protocol for increasing photo availability in the presence of disconnections. A simple solution would be for each device to maintain a reserved space for caching, and use it for keeping replicas of photos published by other users. However, solutions that involve the partial or total replication of the albums can be considered.

2.3. Testing Mechanisms

In order to allow testing the system, the following additional features should be implemented. The mobile app should produce a timestamped log of all operations it performs (including all detections of nearby devices) and provide a GUI mechanism to display that log. The server should produce a timestamped log of all operations it performs and the mobile app should display the server log. If the availability advanced feature is implemented the mobile app GUI should include a mechanism to restrict the amount of storage to store other users' album-slices.

3. Implementation

The target platform for P2Photo is Android version >= 4.0. Communication between album group members should be based on WiFi Direct. The official course programming language for the project is Java but other language may be allowed (check with the course staff). Students may use Android APIs freely. However, third-party libraries are not allowed unless explicitly approved by the course staff.
Unfortunately, it is not possible to provide real Android devices to test the project. Therefore, the project must be tested on a software-based emulation testbed, which comprises the native Android emulator and the Termite WiFi Direct emulator. This testbed will allow you to emulate: 1) users devices executing the mobile application, 2) communication between devices, 3) beacon signaling, and 5) communication with P2Photo server. The testbed can be set up on a single computer. This computer can also be used to host the P2Photo server.
To implement the P2Photo scenario, a single WiFi network should be used for all communication. Whenever group members are reachable via WiFi Direct that should be the preferred communication method.
The Android emulator comes natively with Android SDK and provides support for GPS and Internet connectivity. However, it does not emulate WiFi Direct and Bluetooth APIs.
Termite is a WiFi Direct network emulator. Termite can emulate a wireless network of virtual nodes that move over time and are able to interact opportunistically whenever they are located within close range. Virtual nodes consist of Android emulator instances running the test application. The developer is responsible for specifying the topology and dynamics of the virtual network.
In the context of P2Photo, Termite must be used to emulate WiFi Direct communication between user devices. Termite will be made available on the course website.
The interface to be provided by the P2Photo app should be simple and, at the same time, complete, i.e. show in a clear way all the supported features.

4. Development Stages

We recommend that you develop the project in five stages:

1. **GUI design:** study the requirements of the project and design the graphical user interface of your application. Create an activity wireframe of the application.
2. **GUI implementation:** implement all graphical components of your application including the navigation between screens. At this point, do not worry about networking. Use hard-coded data to simulate interaction with the server. Make sure to design your application in a modular fashion.
3. **Implement cloud-based architecture:** implement the central server and extend the mobile application in order to communicate with the server. Implement the interaction with the cloud storage service.
4. **Implement peer to peer architecture:** complete the baseline functionality of the project by implementing WiFi Direct communication. You only need to use Termite at this point.
5. **Advanced features:** implement advanced features regarding security and availability.

5. Grading Process and Milestones

The projects will be evaluated based on several dimensions. The most important points are: implemented baseline and advanced features, modularity of the implementation, technical quality of algorithms and protocols, and resource efficiency decisions. We will also assess the responsiveness and intuitiveness of the application interface. However, the aesthetics of the GUI will **<u>not</u>** be graded. Therefore, as said above, students should keep the GUI as simple as possible while providing the required information.
There are two important project milestones:
1. April 22nd: Project Checkpoint – Students should submit their current prototype of P2Photo so that they can receive feedback on the architecture and status of the prototype.
2. May 17th: Project Submission – a fully functional prototype of P2Photo and a final report. The prototype sources and the report must be submitted through the course website. A template of the report will be published on the website. The report must describe what was and was not implemented, and it is limited to 5 pages (excluding cover). The cover must indicate the group number, and name and number of each of the group's elements.

6. Collaboration, Fraud and Group Grading

Student groups are allowed and encouraged to discuss their project’s technical solutions without showing, sharing or copying code with other groups or any other person from within or without the course. Fraud detection tools will be used to detect code similarities. All instances of fraud will disqualify all groups involved and will be reported to the M.Sc. coordination and to IST authorities. Furthermore, within each group all students are expected to have a working knowledge of the entire project’s code.