TECHNISCHE UNIVERSITÄT DRESDEN

Department of Computer Science | Institute for System Architecture, Chair for Computer Networks

AroundTheCorner

Application Development for Mobile & Ubiquitous Computing - Final presentation

Group 16:

António Monteiro

Javid Abbasov

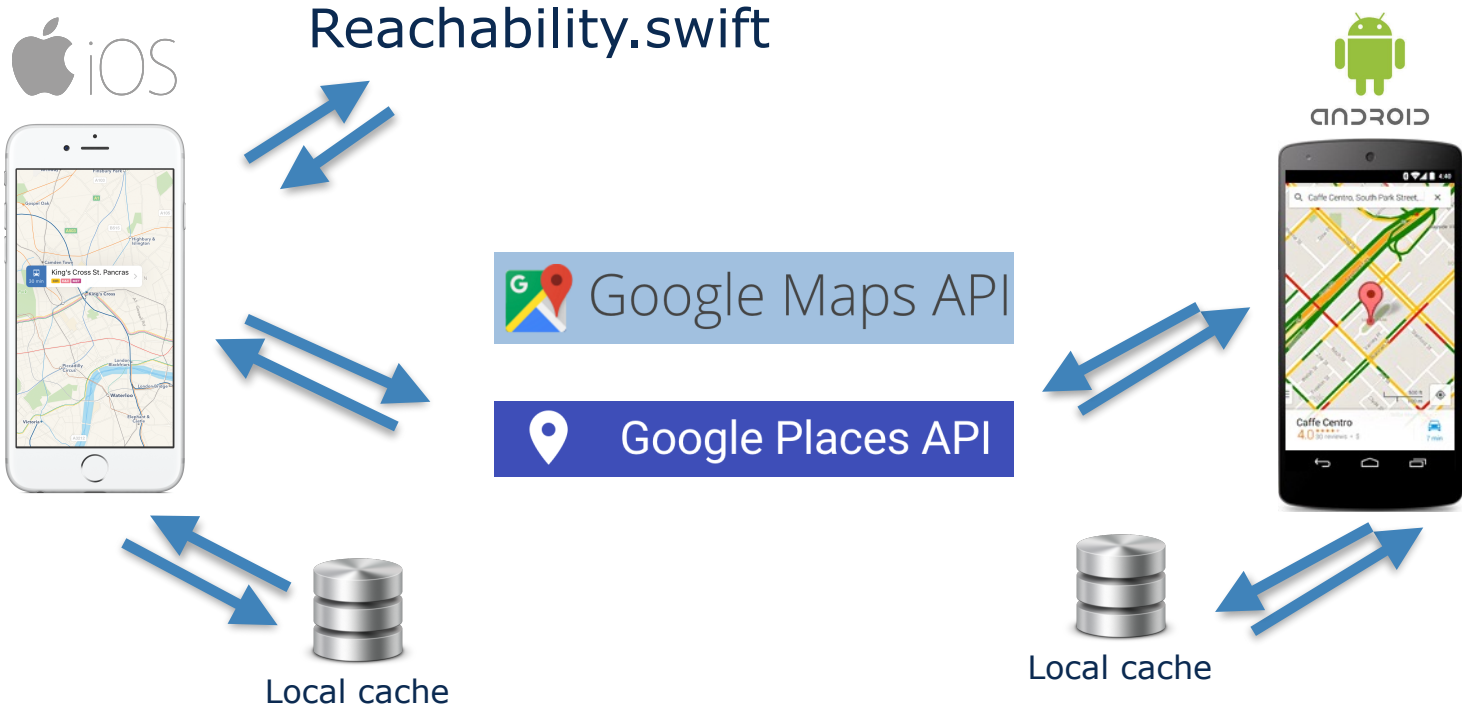
Dresden, 29.01.2016

Application Scenario

- Find place near user's location:
  - in a map
  - by type of place (restaurant, bar, nightclub, etc.):
- check information on given place (is it open?, phone for reservations, etc.)
- bookmark a favorite place;

Demo

Architecture



Challenges & Adaptations

Energy challenge:

- GPS location consumes a lot of energy
- Mobile devices have limited power sources

Challenges & Adaptations

Adaptation:

- Use location only while using the app
  - (no background location querying)
  - How to achieve this? e.g. iOS:
  - NSLocation**When**InUseUsageDescription *vs.*
  - NSLocation**Always**UsageDescription

Challenges & Adaptations

Usability & form factor challenge:

- Application needs to be usable in different devices

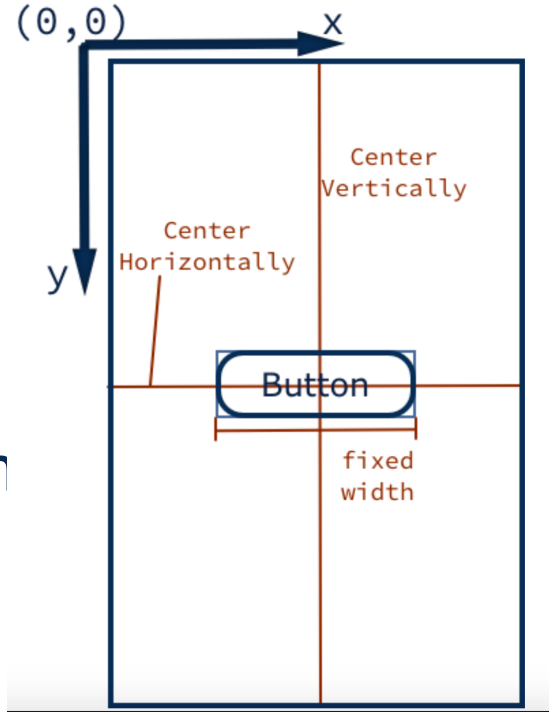
Adaptation:

- iOS & Android native apps (cross-platform usability)
- Adapt to different screen sizes

Challenges & Adaptations

Adapt to different screen sizes:

- e.g. for iOS:
  - take advantage of Auto Layout capabilities
  - specify constraints on how th interface should be laid out
  - constraints are just mathematical definitions



Challenges & Adaptations

Adapt to different screen sizes:

- e.g. for Android:
  - use Linear Layout
  - specify `layout-weight` and `layout-sum`
  - defines proportions between elements on the screen

Challenges & Adaptations

Offline challenge:

- Some application functionality needs to be available offline

- How to detect it? e.g. for iOS:
  - Reachability.swift library

Reachability.swift

```
let reachability : Reachability?
do {
  reachability = try Reachability.reachabilityForInternetConnection()
} catch {
  // some weird error creating the Reachability instance
  reachability = nil
}

if let reach = reachability {
  if reach.isReachableViaWiFi() {
    self.getPlaceDetails(placeID, alertView: nil, callback: {
      (place) -> Void in
      marker.userData = place
    })
  }
}
```

Instantiate the Object

Wi-Fi available?

Prefetching!

Challenges & Adaptations

Adaptation:

- Users' bookmarks stored both on- & offline
- Merge updated information when connectivity is restored
    - simple approach (correct because only bookmarks are saved)
        - perform set intersection between local & remote

Challenges & Adaptations

Connectivity challenge:

- Some application functionality needs to be available offline

Adaptation:

- **Prefetching** of place data while on Wi-Fi
- Server API filters and sends only requested types of places (**filtering**)
    - filters passed in the API request
        - (e.g.: show me only “restaurants”)

Pitfalls / Experiences

- Implementation across 2 native platforms
  - Not always easy to coordinate
    - Features, UI, etc.
- Initially planning to target Android 6.0
  - requesting storage & location permissions are different from previous Android versions
    - (user must approve manually)
  - solution: changed target Android version to 5.1 (Lollipop)

Future work

- Server integration
- Polishing UI

Questions?