Application Development for Mobile and Ubiquitous Computing

Seminar Task
First Presentation

Group №: 20
Team: Anton Caceres
Andrii Chaichenko

Why mobile advertising application?

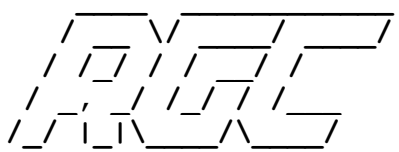
- In Germany people are generous
  - they give lots of good things 'zu verschenken'
  - usually just thowing them to the street container
- Because to post or get an offer you would usually need:
  - computer for web brosing
  - phone to call owner
  - maybe camera to take photos
- However, you can do it all-at-once with our mobile app:
  - no more need to search
  - inbox based on feeds
  - search terms, location-based search
  - mail, browsing, maps and calls all in one place

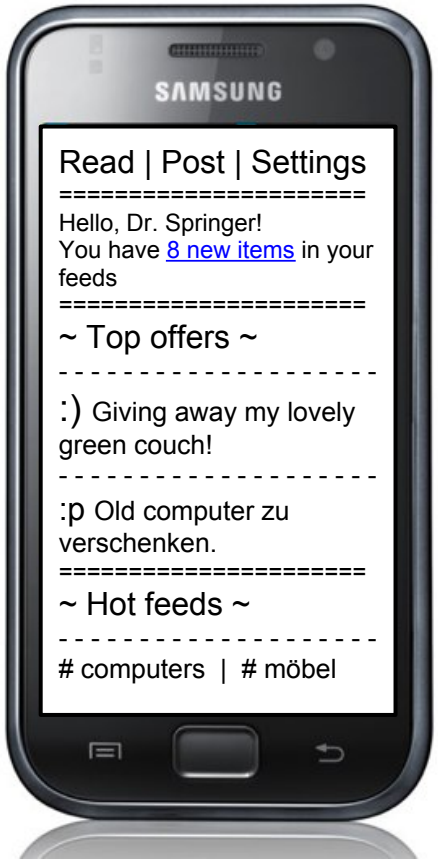
Application Scenario

- user can cpecify searh terms for products, as well as perform standard search
- based on given search terms user receives offers
- offers arrive in feed-like view
- direct contact with seller
- possibility to filter by location (i.e. in 5 km radius)
- post your own ad
- add a picture to your post from a phone camera

Basic features and interface

- Key features - users can read feeds and post new items
- Feeds are groupped into slidable lists of offers
- Everything else is in "settings" tab.
- Mockup inspired by ASCII Art



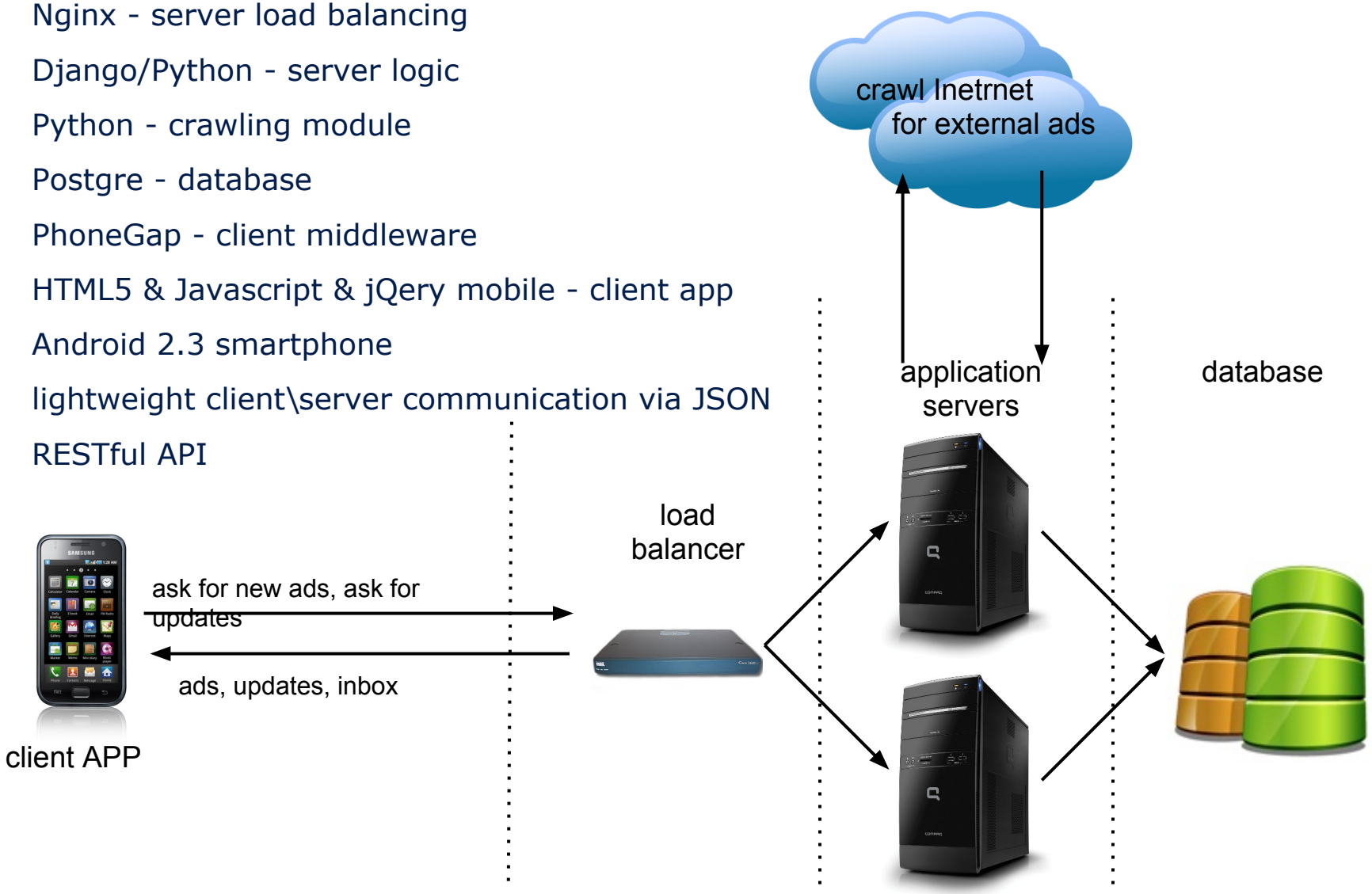


Basic features and interface

- Content is delivered to user (concept of RSS feed), no need to search in different places
- User needs just 1 device to arrange a meeting (search, email, call, lookup the map)
- In case we have enough time, user would also be able to post offers from the single device.
- Multi-platform!

Technologies

- Nginx - server load balancing
- Django/Python - server logic
- Python - crawling module
- Postgre - database
- PhoneGap - client middleware
- HTML5 & Javascript & jQery mobile - client app
- Android 2.3 smartphone
- lightweight client\server communication via JSON
- RESTful API



Challenges

- Crossplatform mobile application development
- Server load balancing
- Secure data transfer and authentication
- Work under unstable network connection
- Real-time updates
- Suitable for low bandwidth

Work plan

- Conceptional design
  - Define use cases
  - Decide on platforms and tools
  - Freeze feature-list
- Set up backend:
  - DB server
  - Django application server
  - front-end NginX server
- Create an external crawling daemon
  - Constanltly running Python daemon
  - gets new posts from ebay, cybersax etc
  - Pushes results via RESTful API to main server

Work plan

- Write application logic
  - Design models
  - Code controllers
  - Add middleware (e.g. authentication, http sessions)
  - Hook up RESTful API
- Create client application
  - Code controllers
  - Code html views
  - Debug damn Javascript
  - Test the whole thing
- Profit

The Most Important Slide

Questions?