Twitter as a Transport Layer Platform

Dmitry Namiot
Lomonosov Moscow State University
dnamiot@gmail.com

AINL-IMSW FRUCT 2016
About

- Information systems in post-SMS time
- How to replace SMS as a main information channel for mobile users
- Interactions between mobile users and applications
- Internet services as data delivery channels for mobile users
Contents

Introduction

SMS deployment

Non-SMS interactions

T411 model

Performance and Reliability
Introduction

• SMS is a traditional solution for information services on mobile networks
• Built-in client, always on
• Easy to use
• Built-in billing
• Growing price
• Telecom based solution: how to get a number?
SMS deployment

- SMS (MMS) request: service_number, key_word, additional parameters
- Could be integrated with QR-codes
- We can send links
- Programming: SMPP and service libraries on the top
- In the most cases: web programming
- Do it yourself tools: small penetration
Non-SMS channels

- Push-notifications: require application
- In the most cases, it is one-way interaction
- Social networks: easy to publish information
- Social circles are problems
- Internet Messengers: potentially is growing area, no clear winner at this moment
Why Twitter?

• The roots are in SMS
• There are no social circles
• Rich API
• Web browser as a client
• Could be integrated with QR-codes
• Used in SMM
Can we borrow Twitter’s delivery mechanism?

- Tweet a program (Wolfram Alpha)
- IFTTT
- Aperator
- Twitter-based automation in IoT
T411 service

- DIY information service: http://t411.linkstore.ru
- Programming processing for tweets like: key_word additional parameters
- Account in Twitter plays a role of a service number
- Request is a “mention” and/or direct message
T411 architecture

• The typical requests:
• t stock_ticket E.g.:
  t ORCL   t YNDX
• w city_name
  w msk   w spb
• Geo-coding support
• Could be linked with QR-code
• Google App Engine
T411 architecture

- Keyword: requested and reserved by the user (author of service)
- Author registers a web hook: CGI script for the response
Performance and Reliability

• Works on the top of Twitter API
• Response time for API (as per Twitter): 0.5 s
• Application server processing time: 0.4 – 0.5 s
• Reliability: 96% - 99%