



JavaOne™

java.sun.com/javaone

Building Interactive Mobile Messaging (Short Message Service) Applications

Michael Yuan, PhD

<http://webclip2go.com/>

TS-5584



Learn how to develop a SMS server application
that sends / reacts to SMS messages with the
mass



GOAL

Agenda

- **What is a SMS Mobile Application**
- Free Tools and Services
- Use Short Codes
- Scalability and Design Issues
- Conclusions

Real World Examples

➤ American Idol

- Short codes for ATT users
- Generates \$70m per show!

➤ Larry King quick vote

- Text in answers to questions in the show
- Get results before the show end
- Easy to prevent double voting

➤ Google and Yahoo

- Text in to get URL to download apps

➤ Amazon SMS service

- Comparison shopping
- buy via SMS

More examples

➤ Political campaigns

- Obama is followed by 20k+ people on twitter. And he follows almost all of them back.
- NGOs use SMS extensively in developing countries (e.g., election monitoring, activism)

➤ Extension to social networks

- Facebook SMS
- Dodge Ball
- Twitter and many copy cats

➤ Tech conference session feedback

- The “unofficial” twitter mob feedback in SXSW keynote was the talk of the conference
- Well, I am still waiting for JavaOneSM conference to do it!

PUSH-based applications

➤ Reminder from doctors

- Appointment reminders
- Reply to re-schedule

➤ Opt in to get promotions

- Red Box / McDonalds video vending machine
- Free meals on college campuses

➤ Alerts

- Airport / flight updates
- BA sends out 11,000 SMS apologies after its opening of Terminal 5 of Heathrow airport

➤ Location and context-sensitive PUSH?

➤ Privacy is a big issue in SMS Push

Add SMS to your existing apps

- Expose some of your data via SMS
 - Amazon SMS service
 - Google SMS search
- Deploy your mobile web applications
- Provision mobile rich applications
 - Google Maps / Yahoo Go
 - Mobile games
- Push alerts when things happen
- Manage user accounts
 - SMS-based account verification when gmail first came out

Agenda

- What is a SMS Mobile Application
- **Free Tools and Services**
- Use Short Codes
- Scalability and Design Issues
- Conclusions

Use the Operator's Email Gateway

➤ Send email to phones

- `phonenumber@tmomail.net`
- `phonenumber@vmobl.com`
- `phonenumber@messaging.sprintpcs.com`
- `phonenumber@vtext.com`
- `number@cingularme.com`
- `number@txt.att.net`

➤ Problems

- No way for the user to reply
- Cannot exceed 20-50 messages a day
- Message auto-truncated
- No delivery confirmation

Hack the Google SMS “API”

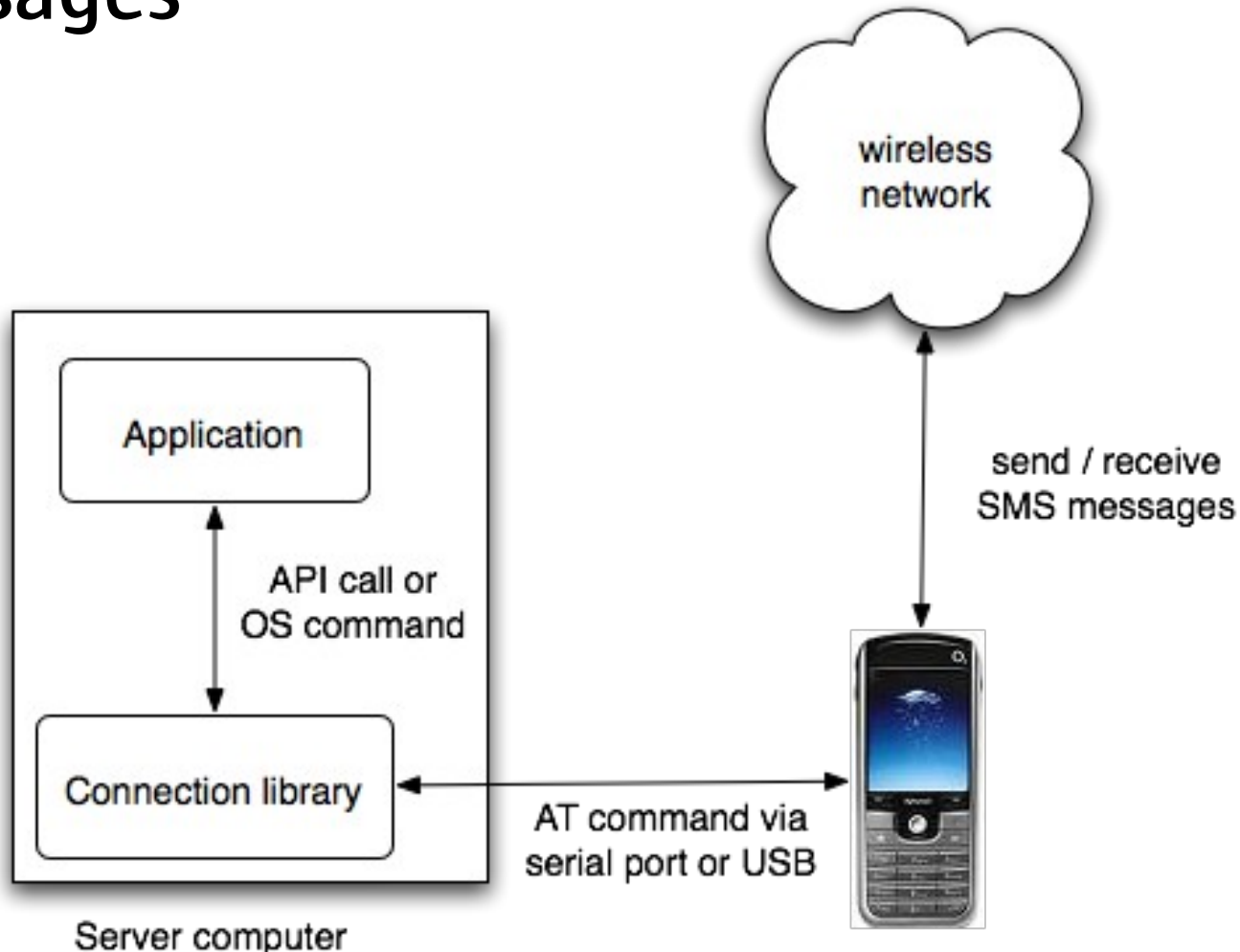
- HTTP Post to the Google “send to phone” web page
- Not a real API but a hack
- The same “API” used by Google IE / FX plugins
- No need to configure email in your app but the same issues with the email gateway approach remain

<http://toolbar.google.com/send/sms/index.php>

Inside the Google SMS “API” Hack

DEMO

Use your own phone to send / receive SMS messages



Communicate from application to phone

➤ Consumer software

- Nokia PC Suite
- Microsoft SMS Sender

➤ Libraries and APIs

- <http://www.gnokii.org/>
- <http://www.gammu.org/>
- <http://www.pxh.de/fs/gsmlib/>
- <http://www.kannel.org/>
- <http://smslib.org/> (Java™ and .NET APIs!)

➤ Problems

- The libraries typically do not support CDMA phones
- Must hose your app at your place with good cell phone signal
- Cannot do large volume messaging

Code example to send messages via SMSLib Java API



DEMO

Code example to receive messages with the SMSLib Java API



DEMO

Use Free Services

➤ Textmarks 41411

- Keyword-based service
- Send SMS to all subscribers of a keyword
- Respond to keyword requests from your own app

➤ Twitter

- Send bulk messages to your followers
- Comprehensive API to manage followers and send messages from your own application

Demo on how TextMarks works in both Push and Pull apps

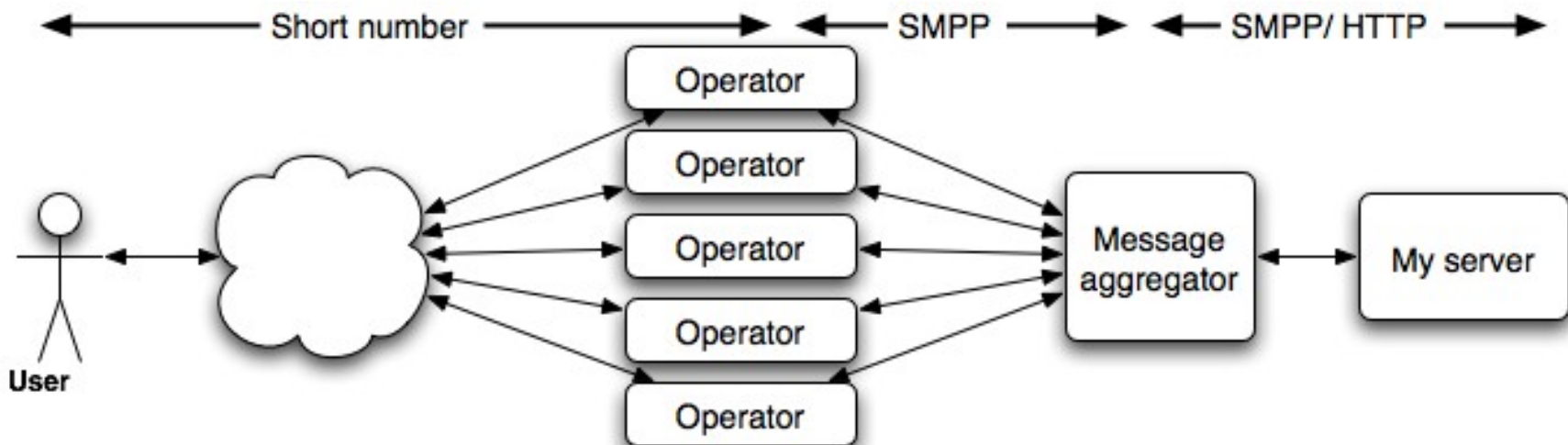


DEMO

Agenda

- What is a SMS Mobile Application
- Free Tools and Services
- **Use Short Codes**
- Scalability and Design Issues
- Conclusions

The Players



<http://en.wikipedia.org/wiki/SMPP>

Why the aggregator

- Buys and resells message packages from operators
- Provide access to all major operators
- No need to negotiate with operators directly
- Support “premium SMS” that would charge the user via the operator billing system
- Helps with certification
- Provides messaging tech infrastructure and tech support
- Many also provide MMS support

SMS Aggregators

- Mblox: <http://www.mblox.com>
- Mobile 365: <http://www.sybase.com/mobileservices>
- Open Market Server: <http://openmarket.com/>
- MX: <http://www.mxtelecom.com>
- Ericsson IPX: <http://www.ipx.com>

How Much Does It Cost?

- Setup fees for the short number across operators
- Maintenance fees for the short number
- Per message fees
- Cost for setting up the infrastructure

Deal with the Operators

- Must support double opt-in
- The entire process needs to be certified for all operators
- Must support banned words
- Must keep record for audits

Protocols and Communication

> SMPP

- Efficient protocol but difficult to code

> Some SMPP libraries

- JSMPP
- OpenSMPP
- SMPPSim

> HTTP

- Custom XML-based API
- Asynchronous notification for message delivery
- Slower performance but sufficient for most apps

Code example to work with the HTTP XML API

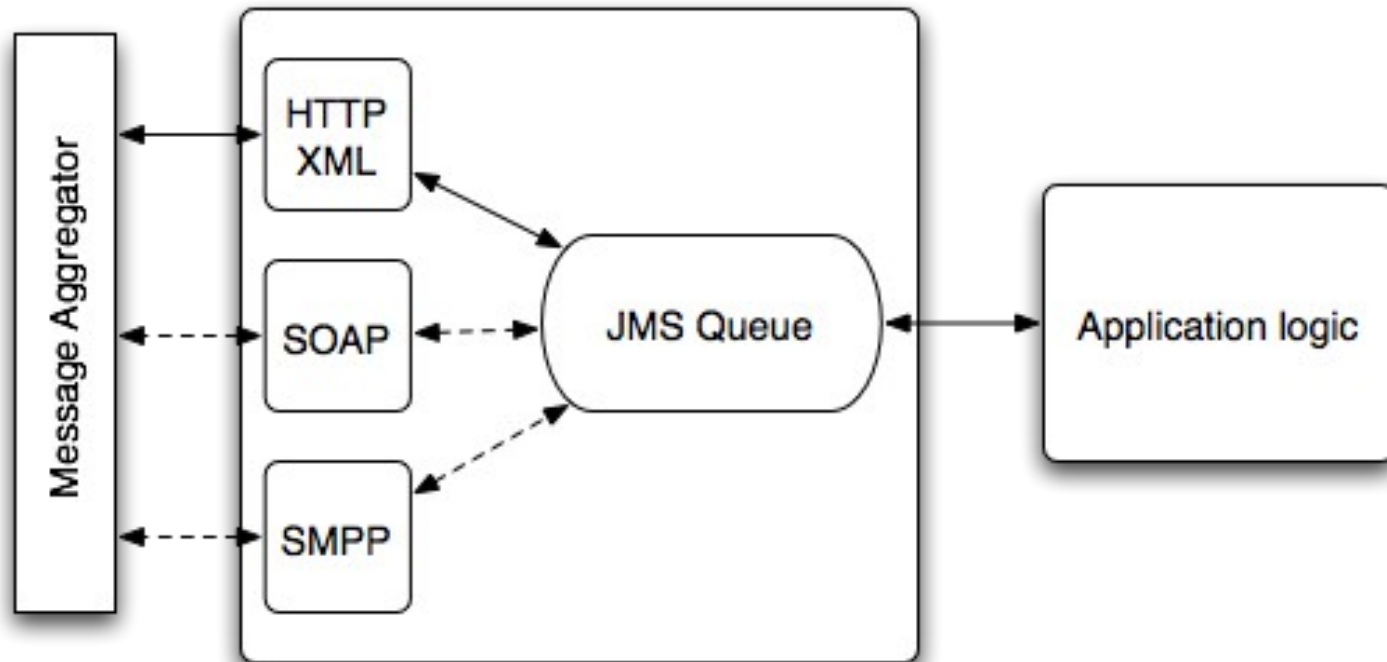


DEMO

Agenda

- What is a SMS Mobile Application
- Free Tools and Services
- Use Short Codes
- Scalability and Design Issues
- Conclusions

Overall design



Synchronous vs. Asynchronous

- **Some HTTP APIs require synchronous response**
 - Receive request and send out response in one round trip
 - Not scalable for high volume applications
- **Most production ready HTTP APIs are asynchronous**
 - Receive message in one round trip
 - Send out message in another round trip
 - Receive delivery notification in another round trip
- **Java Message Service (JMS) is a natural fit for this type of apps**
 - Put received messages in a queue
 - A listener thread reads the queue and send out responses with a “wait notification” in another queue
 - When a notification message is received, dequeue the notification queue

Pluggable Connectors

- Pluggable connectors allow you to work with multiple APIs and multiple message aggregators
- Design a common connector interface for your application
- Plug in multiple implementations
- Use configuration to select which implementation to use

Agenda

- What is a SMS Mobile Application
- Free Tools and Services
- Use Short Codes
- Scalability and Design Issues
- **Conclusions**

Conclusions

- SMS applications represent big opportunities
- There are multiple ways to support SMS in your applications
- Free methods are good for trials and experiments
- For production ready applications, you need to deal with message aggregators
- There are ready-made APIs and libraries in Java platform

THANK YOU



Michael Yuan

<http://webclip2go.com/>

TS-5584

