Integrating iOS Applications with Backend REST Services

Monday, October 4th - JAOO - Århus, Denmark

Adrian
Kosmaczewski

akosma software

akosma.com
github.com/akosma
linkedin.com/in/akosma
formspring.me/akosma
twitter.com/akosma
slideshare.com/akosma

My Software Passion

People

Team members, users, clients, society in general

Software is a process

Software is a **social** process

One Problem

Several Solutions

Questions

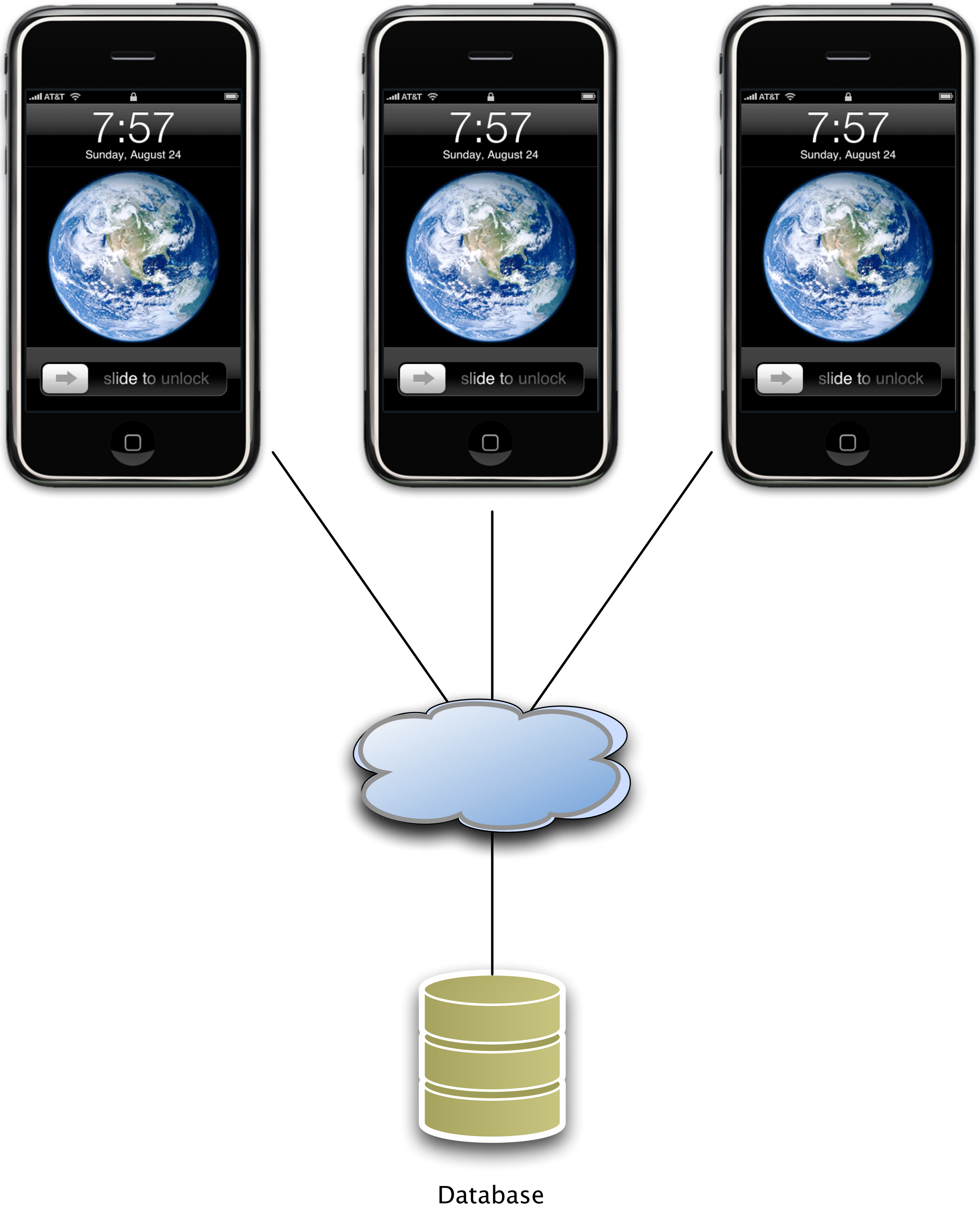
Web Developers?

iOS Developers?

API Designers or Developers?

Server-side Technologies?

1
The Problem



AT&T
7:57
Sunday, August 24
slide to unlock
AT&T
7:57
Sunday, August 24
slide to unlock
AT&T
7:57
Sunday, August 24
slide to unlock

Database

Many Questions

Formats?

Libraries?

“Best” approaches?

2
The Solutions

Bad news

Far too many

REST vs SOAP

XML vs JSON

Synchronous vs. Asynchronous

Good News

Introducing iPhoneWebServicesClient

http://github.com/akosma/
iPhoneWebServicesClient

2 parts

1

PHP server app

2
iOS client

Many formats

XML

JSON

(duh)

- XML
    - 8 libraries
- JSON
    - 2 parsers
- YAML
- CSV

- SOAP
- Property List
    - XML
    - Binary
- Protocol Buffers

Extensible

(add more formats and libraries if you want)

Variable sized dataset

(from 1 to 5000 items per call)

Heterogenous Data Source

(the same data in different formats)

XML Libraries

(lots of them)

3
Demo

4
The Tests

- Local
- Wifi
- 3G
- EDGE

- All Combinations
- Different dataset sizes each time

5
Results

Easier to implement on the iOS side?

1. **JSON + Property Lists + CSV**

2. XML (DOM) + Protocol Buffers

3. XML (SAX)

4. SOAP + YAML

Easier to implement on the PHP side?

1. **JSON + YAML**

2. Property List + CSV + XML

3. Protocol Buffers

4. SOAP

Smaller Payload

1. **CSV + Protocol Buffers + Binary Plist**

2. JSON + YAML

3. XML

4. SOAP

Fastest Deserialization Speed

1. **Property Lists + TBXML**

2. SOAP + libxml (DOM) + Google XML

3. JSON

4. YAML + CSV + APXML

More Portable?

1. **XML + JSON**

2. Protocol Buffers + YAML + CSV

3. SOAP

4. Property Lists

More Readable?

1. **JSON + YAML**

2. XML + XML Property Lists

3. Protocol Buffers

4. Binary Property Lists

Less Memory Consumption?

1. **Binary Property List + Protocol Buffers**

2. CSV + JSON + TBXML

3. XML

4. SOAP + APXML

Some raw, deeply flawed comparisons?

- Binary Plists are 3 to 4 times faster to deserialize than JSON
- iPod touch 2nd Gen is ~25% faster than iPhone 3G
- iPhone 4 is ~300% faster than iPhone 3G
- JSON is 45% of its equivalent XML plist
- Binary plist is 35% of its equivalent XML plist

The “Best”?

1. **JSON + Property Lists**
2. TBXML + Protocol Buffers
3. Other XML parsers + CSV
4. YAML + SOAP + APXML

6
Next Steps

Test with other server-side technologies

(J2EE, ASP.NET, Ruby on Rails, Django...)

Test with other Cocoa networking libraries

- AsyncSocket
  http://akos.ma/0x37v
- IP*Works! for Mac OS X
  http://www.nsoftware.com/portal/macos/
- OmniNetworking
  http://akos.ma/0q
- ThoMoNetworking
  http://hci.rwth-aachen.de/thomonet
- ConnectionKit
  http://github.com/karelia/ConnectionKit/

Test with other serialization systems

- MessagePack
  http://msgpack.org/
- Apache Thrift
  http://incubator.apache.org/thrift/
- BERT
  http://bert-rpc.org/
- Apache Avro
  http://avro.apache.org/
- ONC RPC aka Sun RPC
  http://akos.ma/va

Open Network
Computing Remote
Procedure Call

Test with different data sets

(sport results, weather, financial data, hyerarchical data, binary data, etc...)

Thanks!

Questions?