

Public Web Services Interfaces of Social Web

Tomi Saarinen

28.3.2008

Seminar on Multimedia
Spring 2008 Mashups

Contents

- Introduction
- Architectural styles
- Application programming interfaces
- Challenges
- Conclusions

Introduction

- Mashups are web applications that combine information from different sources.
- There are at least two distinct approaches to mashups
 - Data from your site to another web site
 - Data from another web site to your site
- In either case, creating a mashup requires structured data from the source web site.
 - This is where Application Programming Interfaces (APIs) come in.

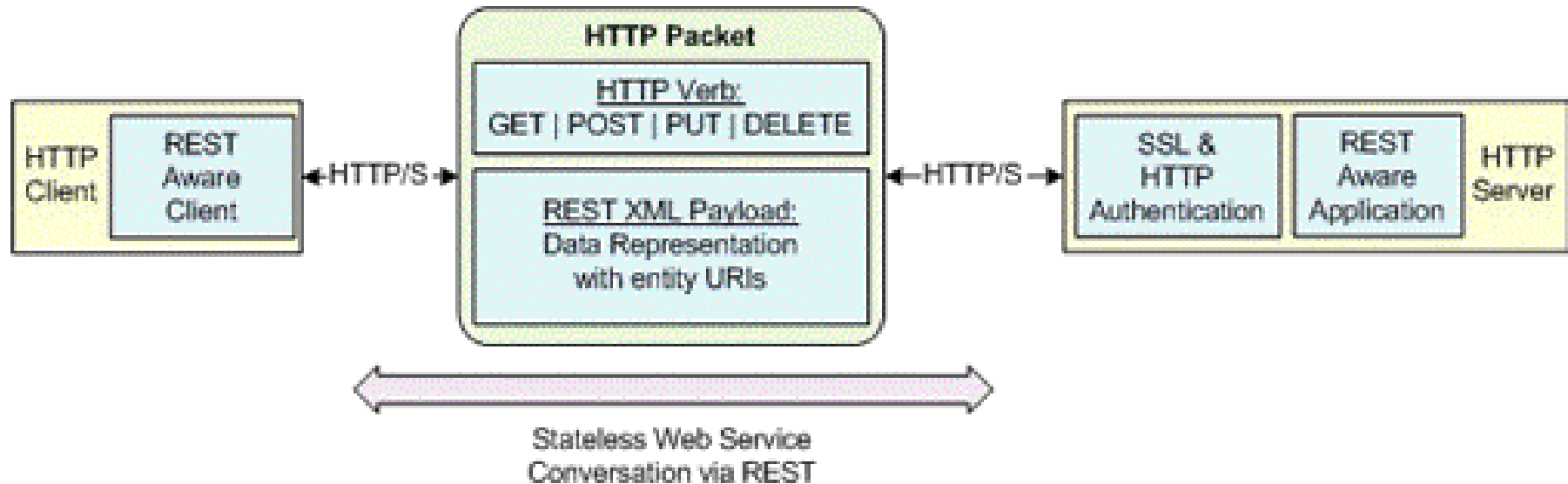
Architectural styles

- Service-oriented architectures (SOA)
- Resource-oriented architectures (ROA)
- Object-oriented architectures (OOA)
 - for enterprise level systems, e.g. Java RMI
 - Not discussed here
- Something else (e.g. Javascript interface)

REST

- Representational State Transfer (REST)
- Resource-oriented approach
- Introduced by Fielding, 2000
- Retrieving resources instead of calling functions
 - Resources are accessed using unique addresses
- Operations for resource lifecycle management
 - e.g. HTTP PUT, HTTP GET verbs
- Returns a representation of data
 - e.g. HTML or XML
- Stateless

REST (2)



Web Services

- Service-oriented approach
- Three stakeholders
 - (service) provider, consumer and broker
- Defined by W3C
 - SOAP and XML-RPC (Remote Procedure Call)
 - WSDL content descriptions
 - Data encapsulation in XML

ROA vs. SOA

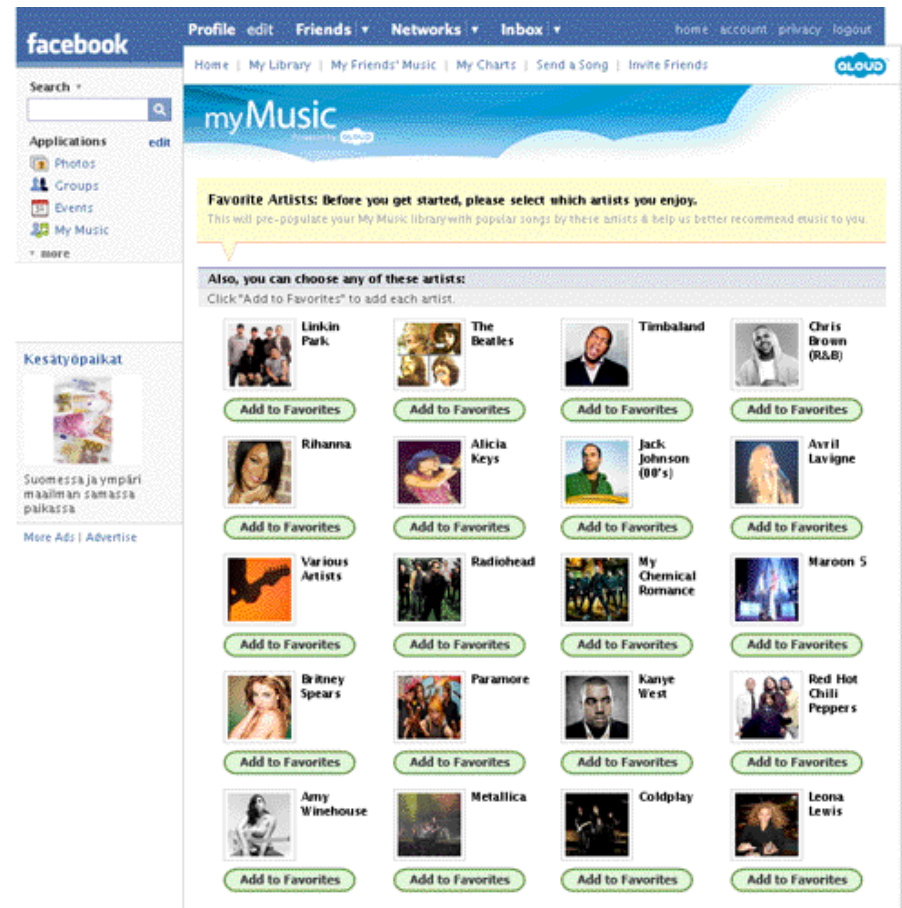
Attribute	Resource-oriented	Service-oriented
Granularity	Resource instances	Service instances
Main Focus	Request addressing (usually URLs)	Creation of request payloads
Addressing and request routing	Unique address per resource	One endpoint address per service
Are replies cacheable?	Yes	No
Application interface	Generic to the request mechanism (e.g. HTTP verbs)	Specific to service – description is protocol specific (e.g. WSDL)
Payload and data format	No – nothing directly linked to address or	Yes – part of service description (e.g. XML)

APIs

- Facebook API
- Flickr API
- Google OpenSocial

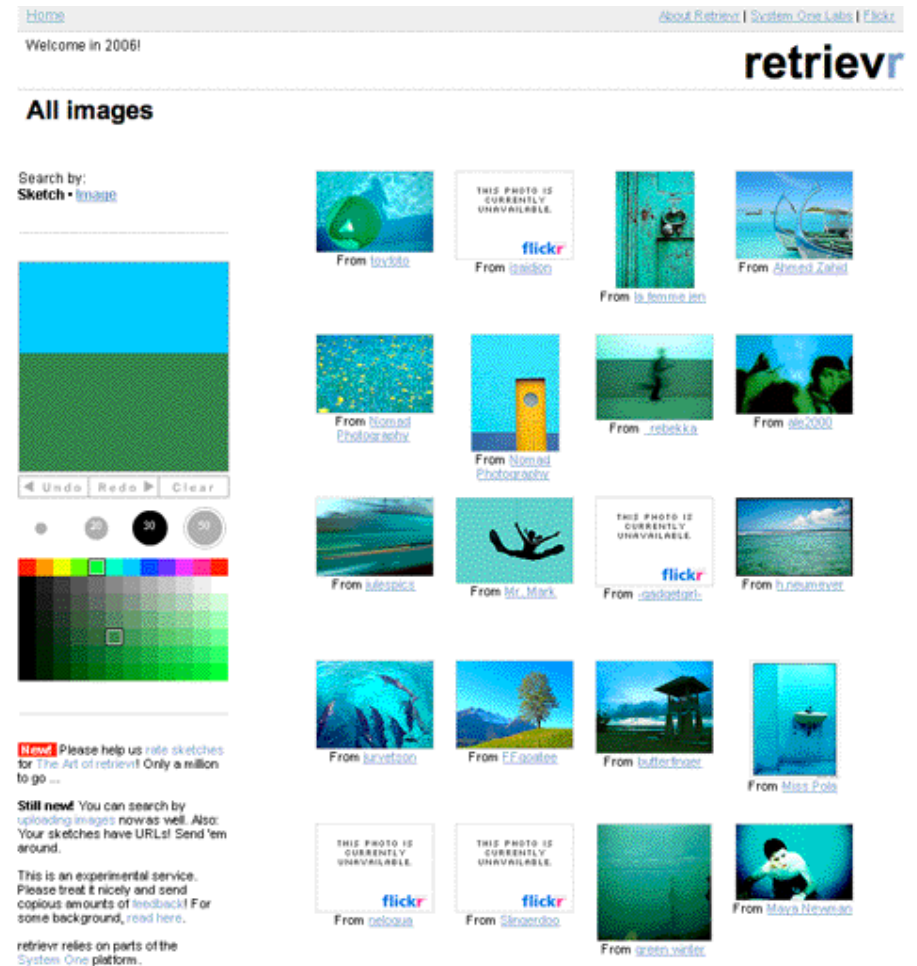
Facebook API

- Based on REST
- Your content inside Facebook UI
- Appears in
 - Canvas page
 - Profile box
 - Left navi...
- FBML or iFrame
- API library for PHP and Java available



Flickr API

- Flickr content on external website
- Many API technologies (e.g. REST, SOAP)
- 109 methods: photos, users, groups, tags...
- Example: Retrievr



Google OpenSocial

- Not tied to any single web site
- Provides abstraction for *social graph*
- Functions for:
 - People
 - Persistence
 - Activities
- Only Javascript interface available



Challenges

- Mashup data existence in the future cannot be guaranteed by anyone.
- Licencing problems and licence inheritance of mashups.
- Privacy issues (e.g. profile spamming).
- User identification is painful for the user.

Conclusions

- Three different APIs for different uses
 - Facebook
 - Your content inside Facebook layout
 - Solid REST api with many features
 - Flickr
 - Flickr content on your site
 - Several API technologies
 - OpenSocial
 - Great idea but incomplete in many ways
- Interfaces are evolving rapidly and there will be many interesting mashups in near future.

That's all folks.