WebdamPoor outlooks and Webdam system implementation

Emilien Antoine

Webdam workshop 04/03/2011

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche SACLAY – ÎLE-DE-FRANCE

WebdamExchange implementation

Work with

- principals: participant in the system
  - performers compute and store data
  - requesters request
  - owners own data or principals
- facts : piece of knowledge used to describe
  - data
  - localization
  - access right

WebdamExchange implementation

Statement:

alice-laptop states picture37@**alice** (….) requester **bob** at 12:30, 10/08/2009

- **alice-laptop** performer
- **bob** requester
- **alice** owner

Usually:

- performer are machines
- requester are
  - humans for original requests
  - machines after deduction
- owner could be everything even virtual (groups)

Communication

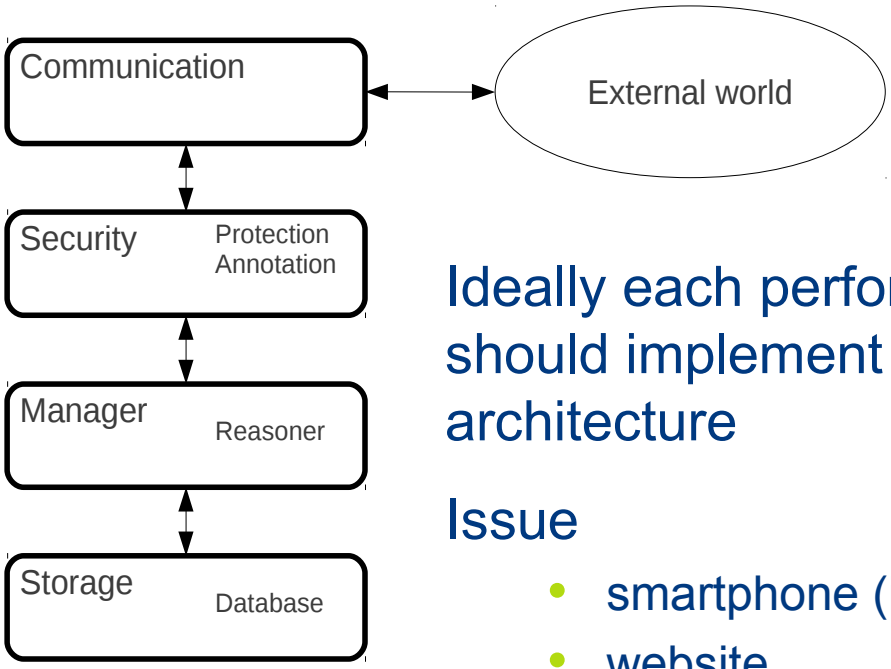
Bob wants alice's pictures:

bob requests pictures**@alice** (….) to **alice-laptop** at 12:30, 10/08/2009

needs

- localization
- authentication
  - bob signature
- access right
  - enforced by alice-laptop

WebdamExchange architecture



Ideally each performer should implement this architecture

Issue

- smartphone (iPhone)
- website
- email server

In practice two kind of peer

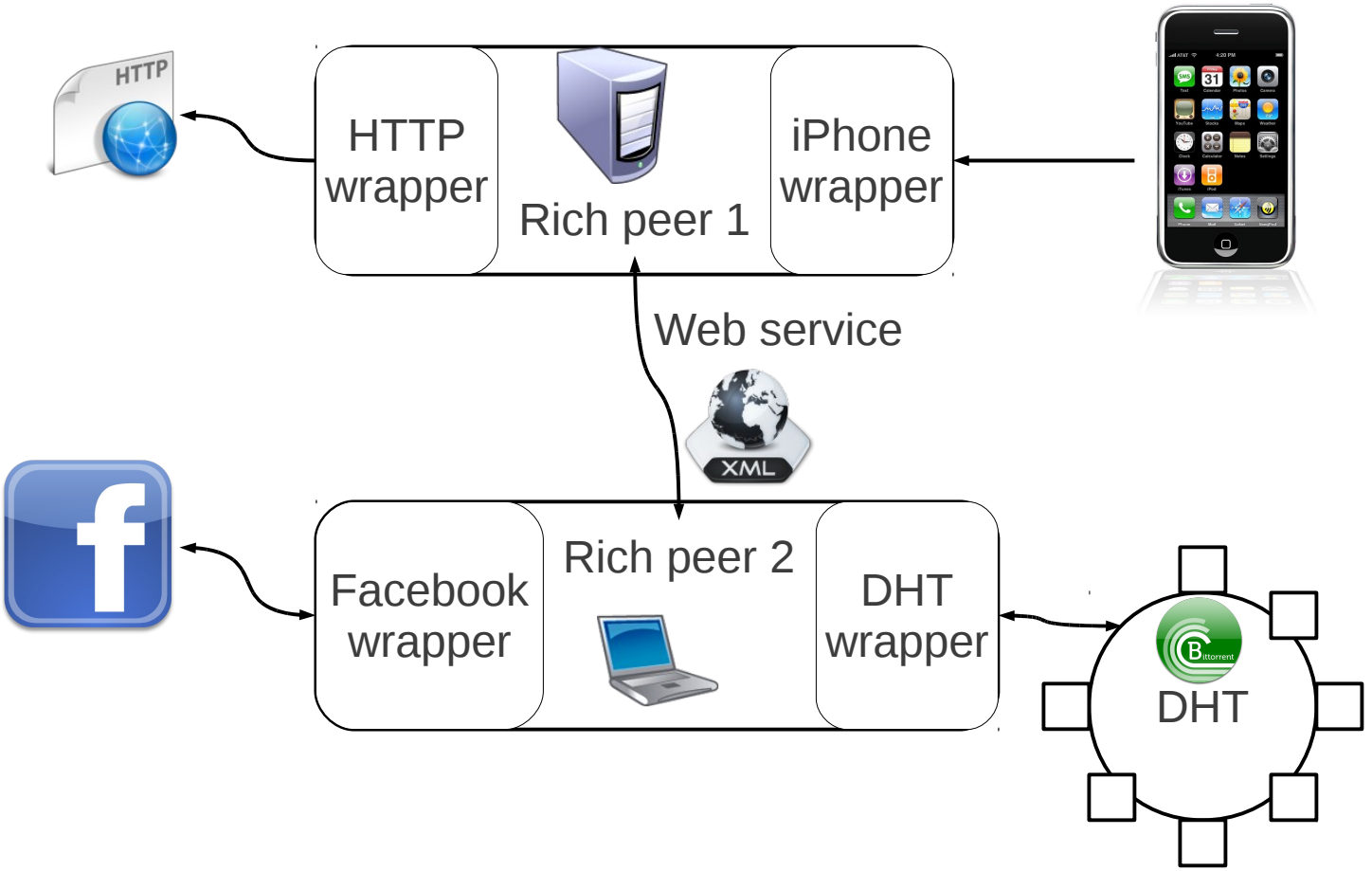
- Rich
- Poor
  - light peer
  - web peer

Rich peer

Implement the complete WebdamExchange architecture

- Communication
  - Web service
- Security
  - Protection for confidentiality: RSA cryptography
  - Annotation for integrity and authenticity: RSA signature
- Manager
  - Reasoner for rule inference system(Datalog engine): IRIS
- Storage
  - Persistent storage system: eXist XML database

Webdam for real life ...



… is Webdam for the poor

Definitively impossible to implement a full Webdam system on all peers

- Communication
  - only receive from iPhone
  - only read on website
- Security
  - no RSA crypto for website
- Manager
  - only for rich peer
- Storage
  - not always obvious, dedicated solution

Degraded peer

- implement Webdam peer specification when it is possible
- rich peer use wrapper to communicate with poor peer
- work in progress focus on
  - iPhone
  - Facebook
  - website (HTTP protocol)

Light peer – iPhone

Use a rich peer as proxy to delegate most of the work:

- communication
  - iPhone sends message to rich peer proxy
  - rich peer store answer in buffer until next iPhone communiation
- security
  - either iPhone is able to deal with RSA crypto or it delegate to the peer
- manager
  - delegate to the peer (having a reasoner seems too ambitious)
- storage
  - either iPhone provide full storage for its facts or it relies on proxy (since it use facts for reasoning)

Web peer – HTTP

HTTP server reachable by its URL

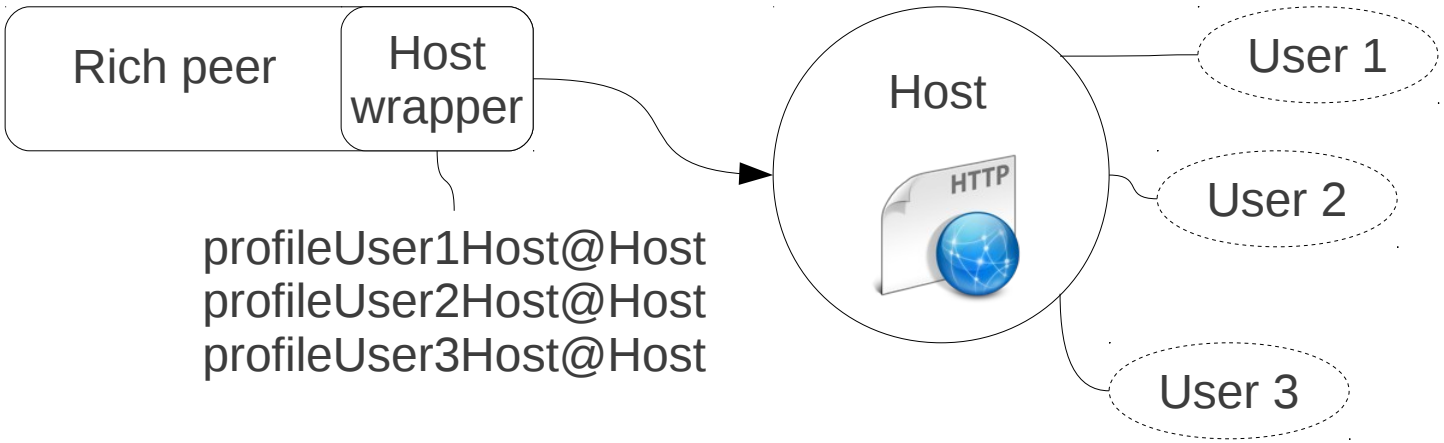
- communication
  - HTTP GET for read enforced by .htaccess or http.conf
  - HTTP POST PUT DELETE resp. for append remove and write but usually irrelevant for website (only read access)
  - own right means having access to configuration files (http.conf)
- security, manager, storage irrelevant

Host peer

Peer with their own authentication system

- own users (virtual principals) which can be Webdam requester
- store facts on which they are performer

For example: social network, Internet forum, … see Facebook, twiter



Secure web peer

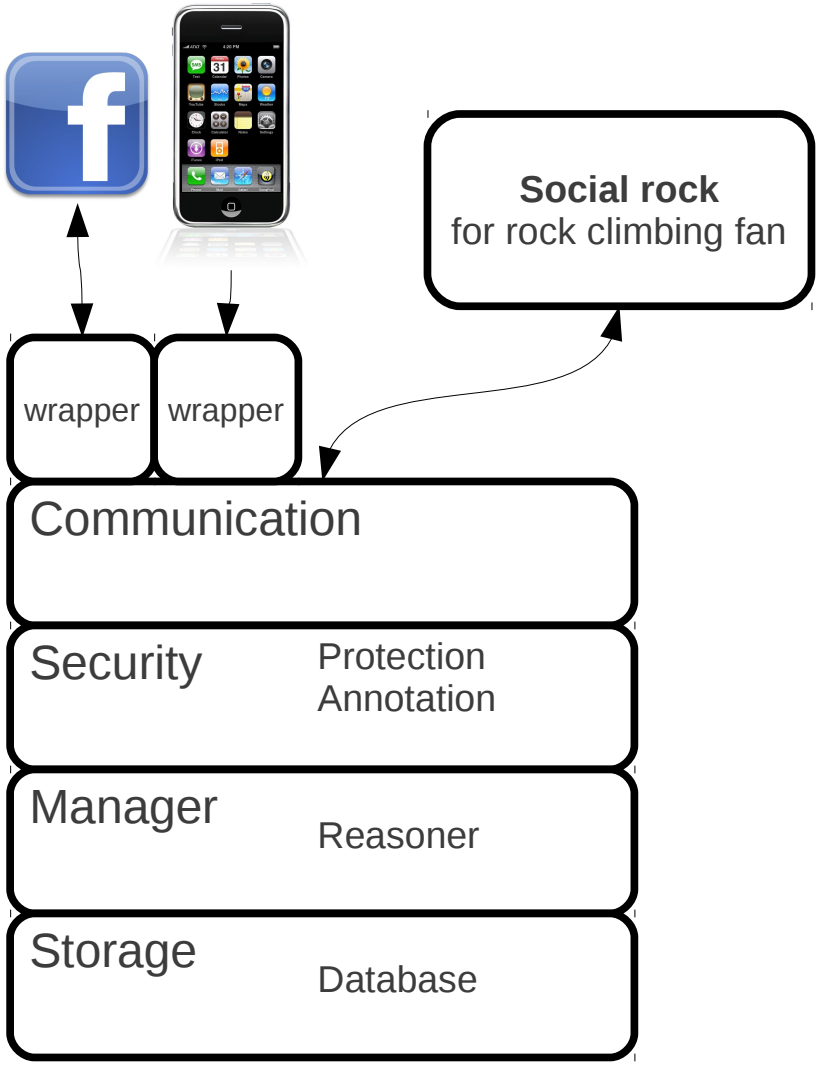
authentication refinement for Web peer and Host peer

- Public key authentication HTTPS / SSL client certicates
- Kerberos or SPNEGO Authentication Server (AS) and a Ticket Granting Server (TGS)
- Secure Remote Password protocol strong password authentication HTTPS / TLS

providing security feature

Implementation

- Social rock
  - demonstration software using web service communication
- Manager
  - use inference engine in the futur (IRIS)
- Storage
  - XML database (eXist)



Extensions

XML Document contains WebdamExchange understandable data

- Share information by spreading an ID card : allow to spread information from outside Webdam system
  - Localization of data: URL
  - Way to access them: protocol such as HTTP, web service, …
    - Communication module work to deal with it
  - Way to authenticate access right: login/password, crypto keys, …
- Store WebdamExchange manifest XML files on HTTP server directory
  - User can sign the directory to express content is trusted

Clues for futur work

- SSH
- VPN
- compounded principals
  - emilien[at]facebook vs emilien[at]laptop-home

Conclusion

- WebdamExchange and WebdamLog models capture some nice problems of web data management: distribution, access control…
  - Their good semantics allow us to prove theorems
  - We are implementing the corresponding system
- Many issues are still open
  - **Concurrency, optimization, *data availability***
    - **data replication: persistence, accessibility, …**
  - Defining and verifying protocols (access control is not violated, one gets all the information one has access to)
  - Looking for a killer application