Cost-effective Web-based Media Synchronization Schemes for Real-time Distributed Groupware

Maarten Wijnants, Peter Quax and Wim Lamotte
Hasselt University / tUL / iMinds / Expertise Centre for Digital Media
Wetenschapspark 2, 3590 Diepenbeek, Belgium / {firstname.lastname}@uhasselt.be

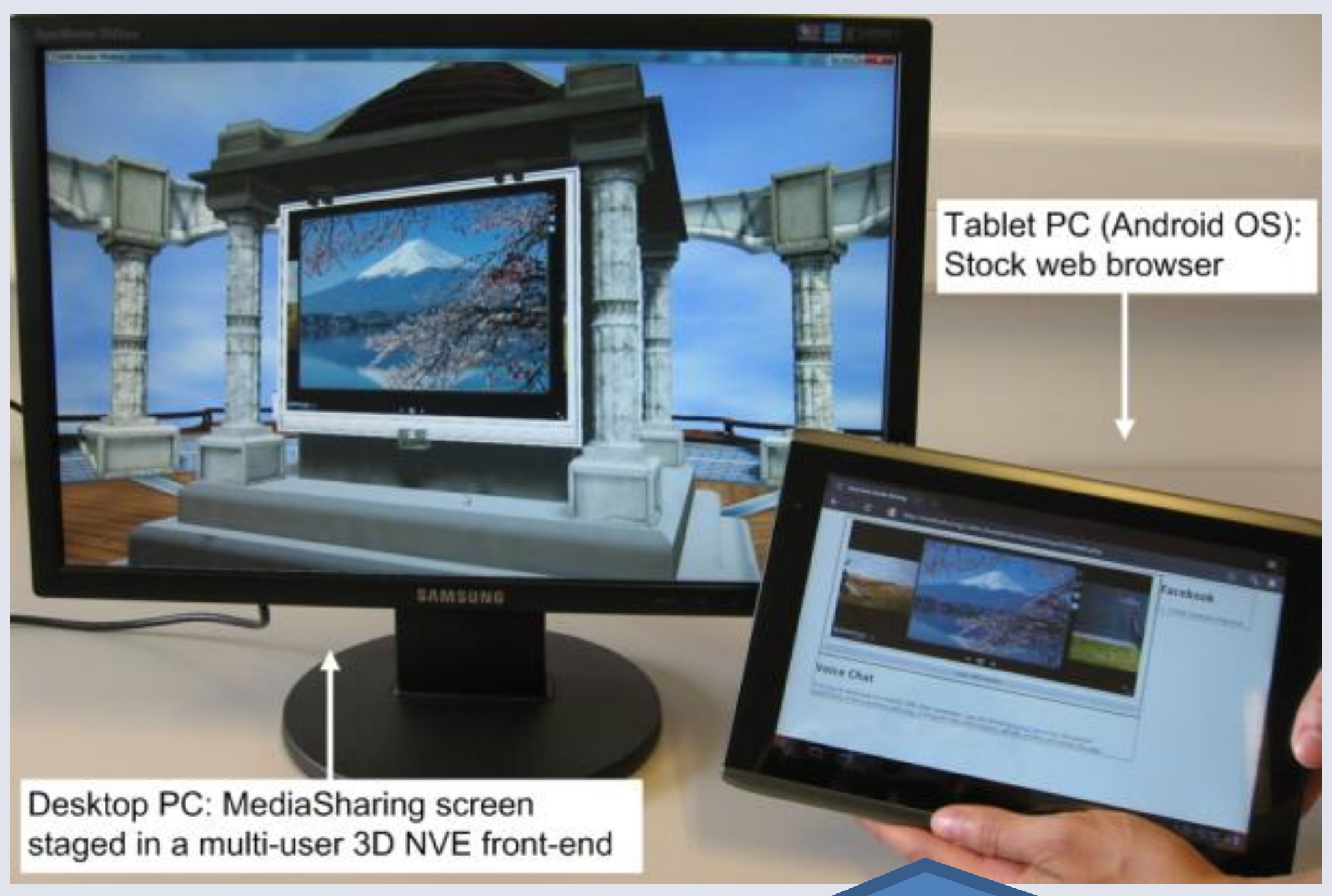
A multitude of application domains benefit from **synchronized multimedia content presentation** and playback **across spatially scattered client stations**. Example use cases situated in the **real-time distributed groupware** realm include online remote tutoring, telework, and the non-co-located synchronous browsing through digital photos.

At the core of any synchronous media sharing and consumption platform lies its real-time content synchronization procedure. The synchronicity requirement namely dictates that spatially distributed participants need to be presented **the same content at (approximately) the same time**. In the literature, this concept is sometimes denoted by the term **group synchronization** or **Inter-Destination Media Synchronization** (IDMS).

This poster presents five concrete hosting and deployment strategies that realize media synchronization over the Web. The proposed techniques have been implemented and experimentally validated as part of the **synchronous MediaSharing** (**sMS**) service, a Web-conform framework which grants geographically dispersed users the ability to share and synchronously consume digital pictures and video clips. All five schemes are completely **Web-compliant**, achieve **relatively loose synchronization** accuracy, and are non-distributed (i.e., they require **centralized coordination**). The proposed synchronization solutions have also been subjected to a **high-level economic cost-benefit analysis** by assessing their infrastructural requirements and the thereby induced operational expenditure.

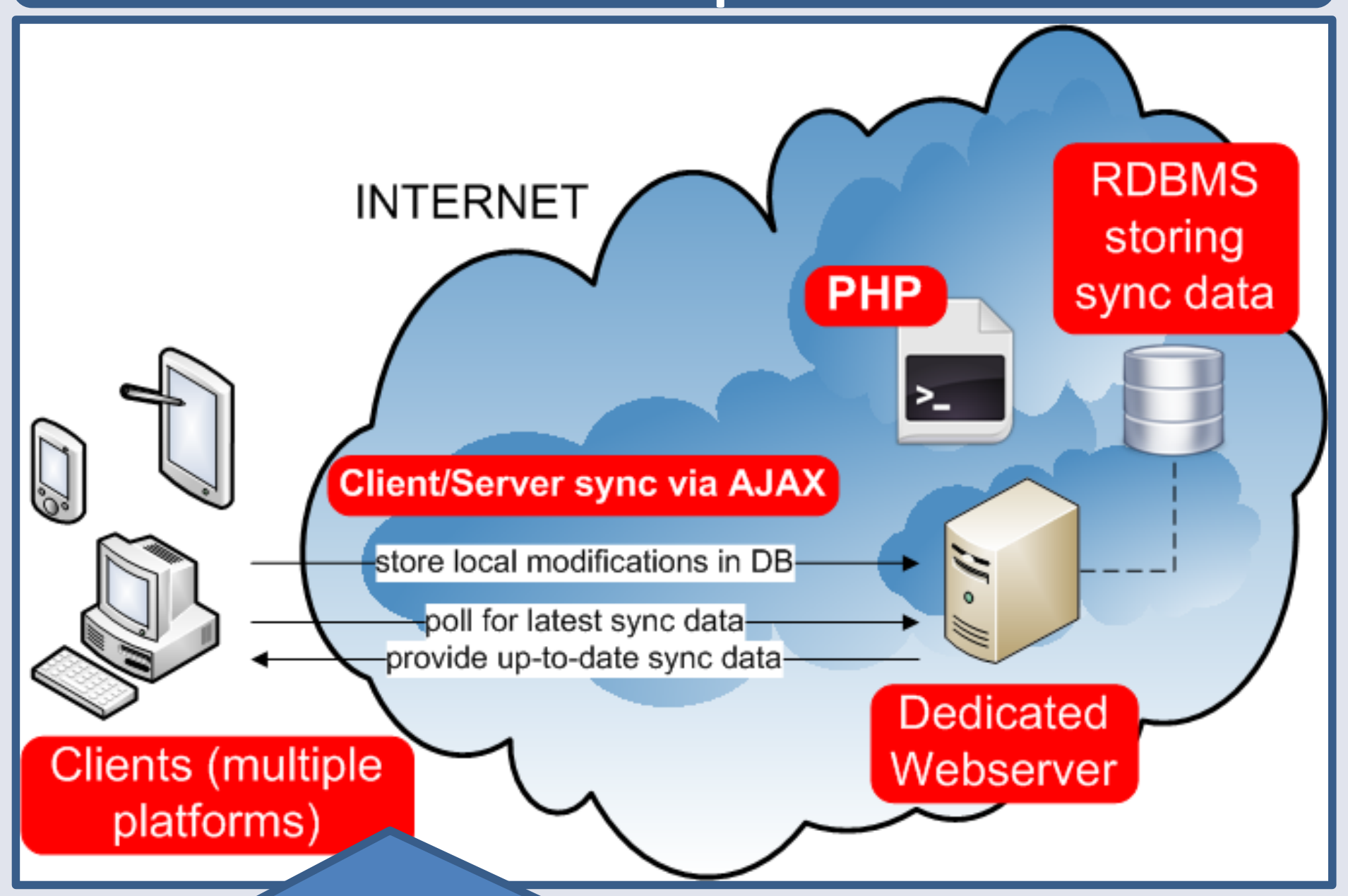
The presented results are not bound to the sMS framework but instead are **generalizable** to many Web-based services that are conceptually situated in the *same time, different place* category of Johansen's groupware typology.

Example cross-platform sMS session (picture sharing scenario involving PC and tablet)


Tablet PC (Android OS): Stock web browser
Desktop PC: MediaSharing screen staged in a multi-user 3D NVE front-end

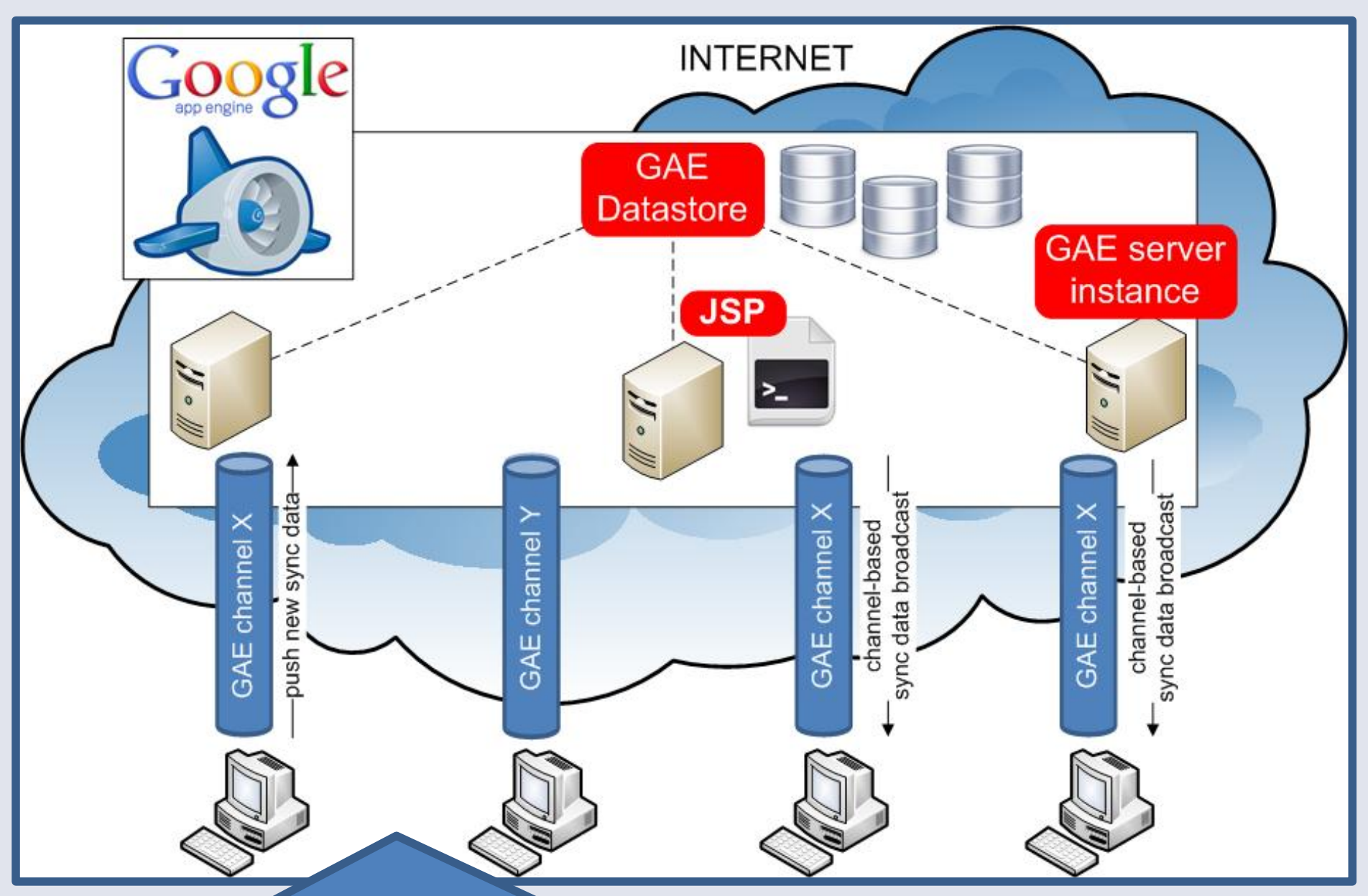
Converged access: Seamless content sync between physical devices and digital world

Webserver plus AJAX


INTERNET
PHP
RDBMS storing sync data
Client/Server sync via AJAX
store local modifications in DB
poll for latest sync data
provide up-to-date sync data
Clients (multiple platforms)
Dedicated Webserver

- **Dedicated back-end hardware** (Webserver)
- Back-end PHP software implementation
- Sync instructions persisted in back-end DB
- **AJAX** as client/server transport mechanism
- **Poll-based sync data dissemination** (clients poll server periodically to solicit up-to-date state)

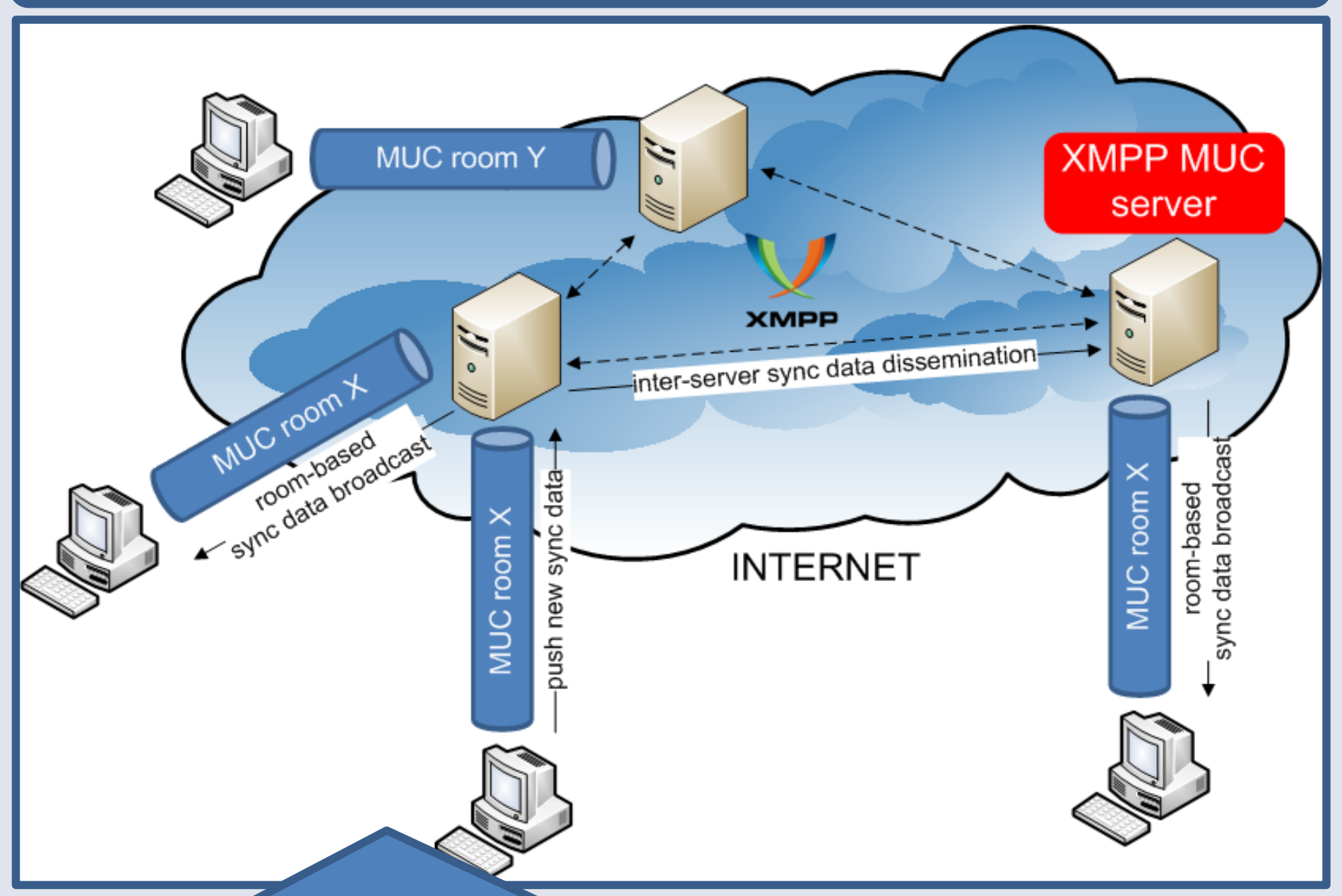
Synchronization in the Cloud



- **PaaS implementation** on Google App Engine
- JavaServer Pages back-end implementation
- Sync instructions persisted in GAE Datastore
- **Push-based sync data dissemination** via GAE Channel API (sMS session maps to channel)

+ Intrinsic scalability of back-end resources
+ Channel API expedites sync record distribution

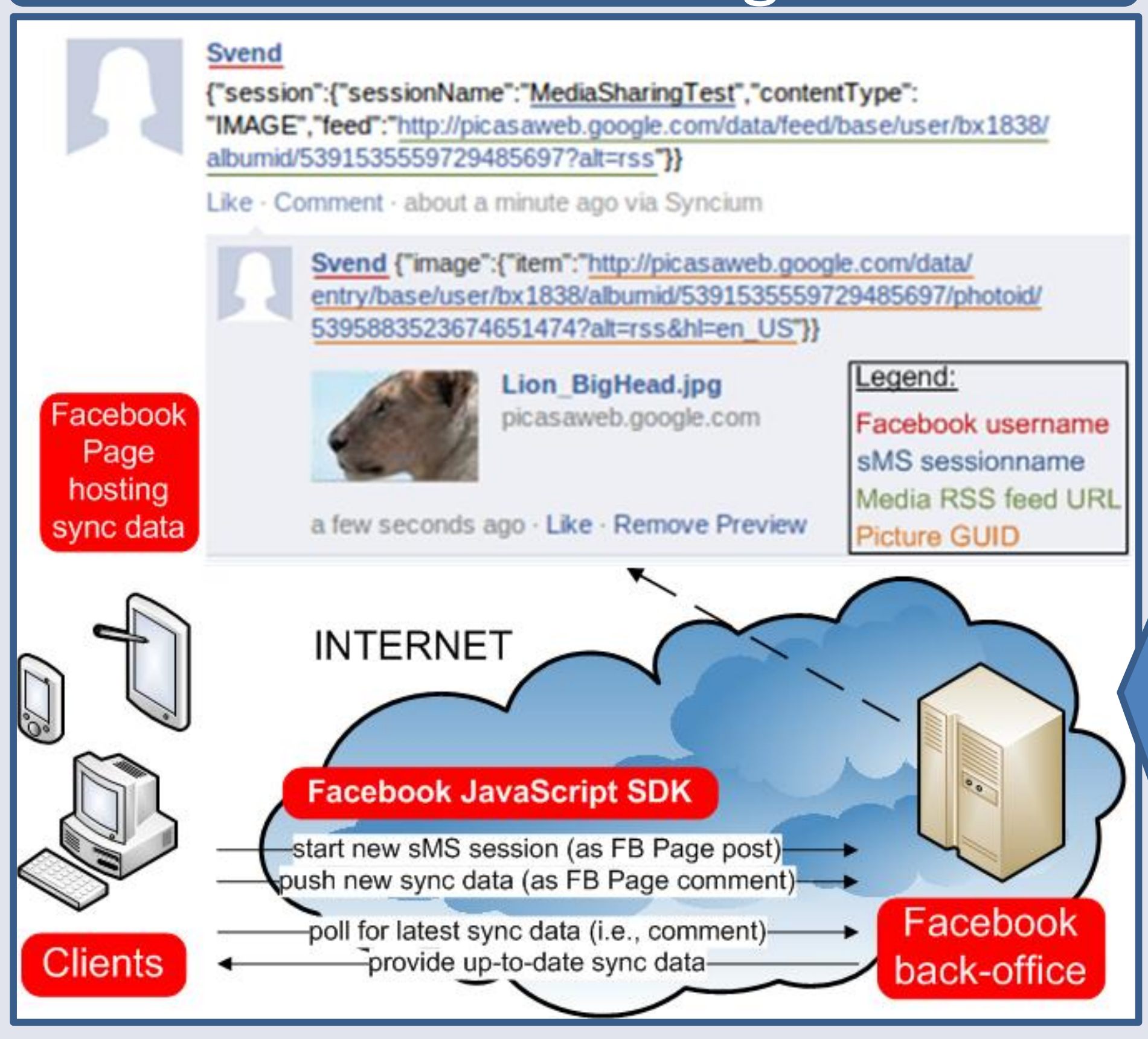
XMPP with Multi-User Chat


MUC room Y
XMPP MUC server
XMPP
inter-server sync data dissemination
MUC room X
room-based sync data broadcast
MUC room X
push new sync data
INTERNET
MUC room X
room-based sync data broadcast

- **Extensible Messaging and Presence Protocol**
- MUC extension introduces **textual group chat**
- sMS sessions are mapped to **rooms**, session **sync data broadcast** within room
- Publish-subscribe interaction paradigm

+ No back-end software implementation
+ Public XMPP MUC servers are freely available

Facebook Page


Svend
{"session":{"sessionName":"MediaSharingTest","contentType":"IMAGE","feed":"http://picasaweb.google.com/data/feed/base/user/bx1838/albumid/5391535559729485697?alt=rss"}}
Like · Comment · about a minute ago via Syncium

Svend {"image":{"item":"http://picasaweb.google.com/data/entry/base/user/bx1838/albumid/5391535559729485697/photoid/5395883523674651474?alt=rss&hl=en_US"}}
Lion_BigHead.jpg
picasaweb.google.com
a few seconds ago · Like · Remove Preview

Legend:
Facebook username
sMS sessionname
Media RSS feed URL
Picture GUID
Facebook Page hosting sync data
INTERNET
Facebook JavaScript SDK
start new sMS session (as FB Page post)
push new sync data (as FB Page comment)
poll for latest sync data (i.e., comment)
provide up-to-date sync data
Clients
Facebook back-office

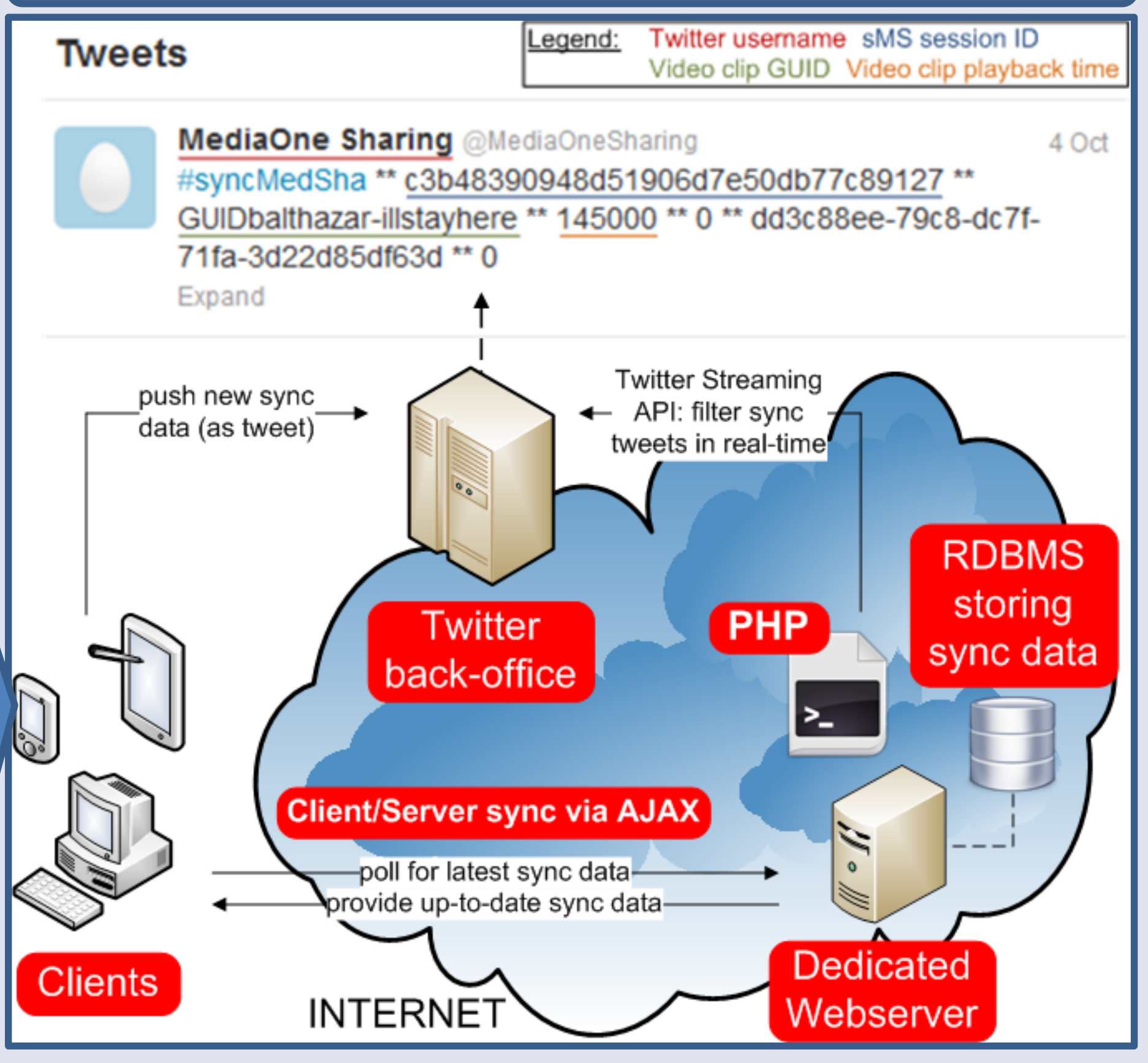
- Sync mediation via specialized, public **Facebook Page**
- sMS session mapped to **post** on Page (holds session config data)
- Session sync data communicated as **comments** on corresponding post
- Clients **periodically query** Page for newest comment
- Auto cleanup of obsolete comments

+ No FB user account pollution
- Facebook dynamically limits publication volume & rate
- Replication across Facebook servers is non-instantaneous → consistency issues

- Sync data as **public tweet**
- **Twitter Streaming API** on back-end Webserver to filter sync tweets
- RDBMS-based sync storage
- Iterative **poll-based sync data dissemination** (AJAX)

- User account pollution
- Tweet length restrictions limit sync message size
- No access to Streaming API from within Web browsers
- Rate limits on Twitter API usage and strict per-day upper bound on tweet publication volume
- API-powered tweet posting is non-instantaneous → consistency issues

Twitter Tweets


Tweets

Legend: Twitter username sMS session ID Video clip GUID Video clip playback time

MediaOne Sharing @MediaOneSharing 4 Oct
#syncMedSha ** c3b48390948d51906d7e50db77c89127 ** GUIDbalthazar-illstayhere ** 145000 ** 0 ** dd3c88ee-79c8-dc7f-71fa-3d22d85df63d ** 0
Expand
push new sync data (as tweet)
Twitter Streaming API: filter sync tweets in real-time
Twitter back-office
PHP
RDBMS storing sync data
Client/Server sync via AJAX
poll for latest sync data
provide up-to-date sync data
Clients
INTERNET
Dedicated Webserver

Cost-effectiveness and Infrastructural Requirements

Webserver plus AJAX	• Requires Webserver hosting (with PHP & RDBMS support) and management • Typically incurs a monolithic monthly or yearly fee
Synchronization in the Cloud	• All back-end hardware is hosted and managed by PaaS cloud provider • Actual cost depends on provider's pricing model; many vendors (e.g., GAE) apply a pay-per-use billing system and offer free policies for apps that satisfy predefined resource usage quota (CPU, …)
XMPP with Multi-User Chat	• Requires MUC-enabled XMPP server(s) • Complimentary XMPP servers are abundantly available online
Facebook Page	• No hosting expenses • Facebook's back-office takes care of persistence and dissemination of sync instructions
Twitter Tweets	• Twitter Streaming API imposes Webserver hosting (with PHP & RDBMS support) and management • Typically incurs a monolithic monthly or yearly fee

The work presented on this poster is co-funded by the iMinds ICON project 3DTV 2.0

universiteit hasselt · EDM · iMinds CONNECT.INNOVATE.CREATE