WebRACE: A Distributed WWW Retrieval, Annotation & Caching Engine

Marios D. Dikaiakos
Department of Computer Science
University of Cyprus
Nicosia, Cyprus
mdd@ucy.ac.cy

IBM Research, NY, July 25th, 2001

Research Group

eRACE Project

Joint work with:

Demetris Zeinalipour-Yazti

WinMob Technologies & University of Cyprus

- Other collaborators
  - Athena Stassopoulou
  - Melinos Kyriakou
  - Eleni Georgiou
  - Christiana Christofi
  - Alexandros Koutsimbelas
  - Andreas Liverdos

U. of Cyprus

WinMob Tech.

R.P.F.

Intercollege

Outline of the Presentation

- Context and Motivation
- eRACE Project: an Overview
- WebRACE: Design & Implementation
- User Interface
- Conclusions and Future Work

Getting Information on Internet

- Browsing and/or Searching
  - Know what we are looking for
  - Time consuming and unproductive
  - Not a continuous process
  - Large volume of information to go through
- Information Dissemination
  - Push information to the user
  - Selective dissemination
  - Continuous process

Information Dissemination Services

- Mailing lists:
  - Subscription-oriented, email-based;
  - Coarse granularity of interest matching
- USENET News:
  - Very popular, huge volume of traffic- information overloading;
  - Coarse granularity of interest matching
- Subscription-based systems:
  - BCIS, Tapestry, Pointcast, SIFT, ProxiWeb, IntelliSync

What Now?

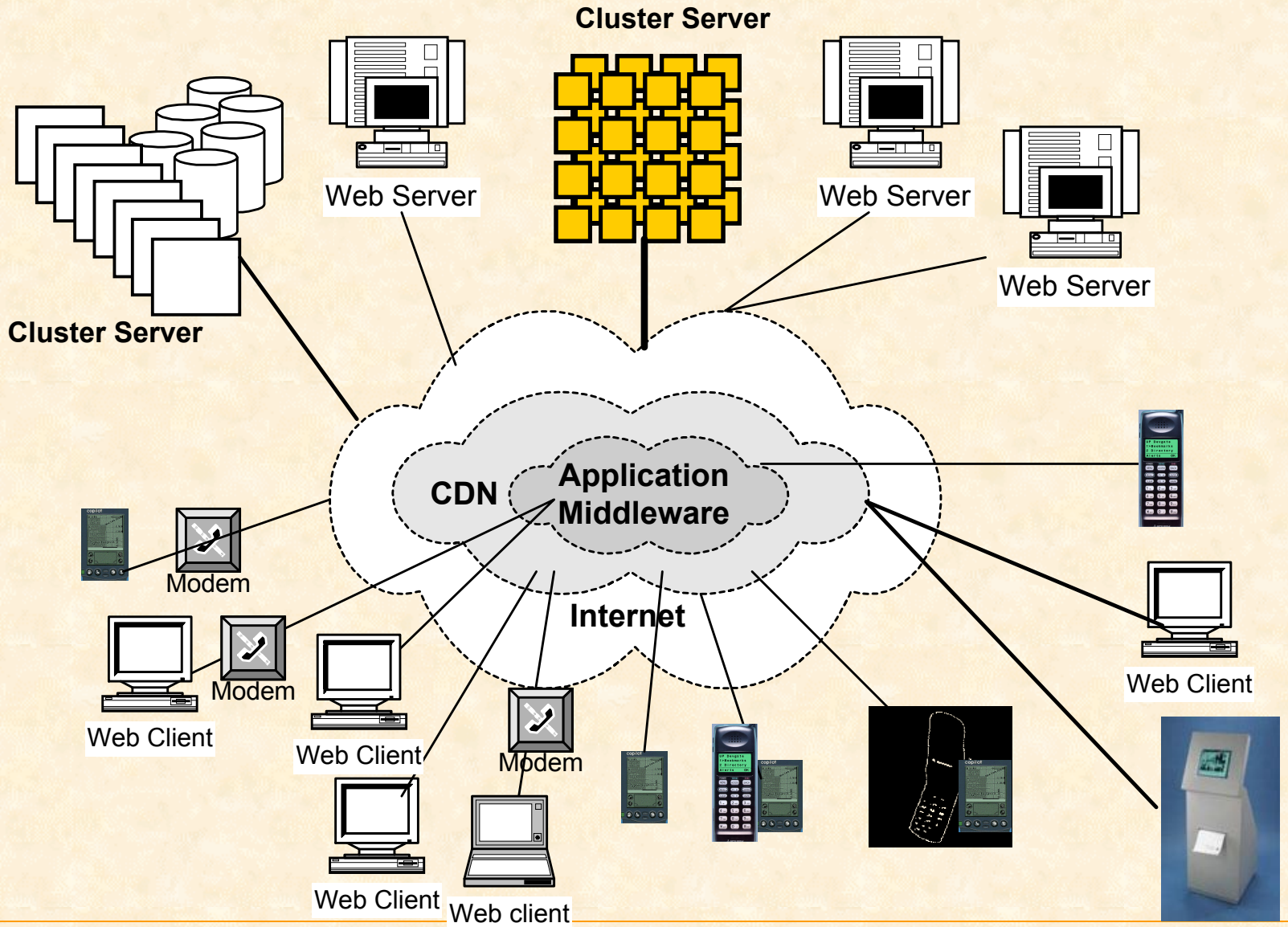
- Universality of client software (browsers):
  - Least-common-denominator output format (HTML, XML & friends).
  - Encourages a convergence of information provision paradigms–push and pull.
- Heterogeneous information sources:
  - Static and Dynamic Web
  - Email
  - USENET News
  - WML sites
- A large diversity of client devices:
  - *Thin, palm, mobile phones*
- A very large and increasing user-base.
- Moving towards decentralized, distributed and scalable infrastructures.

The New Context

The paradigm shift of the basic Web-services model:

- From that of a Web-server running on a well-defined host and providing content to clients over a specific communication protocol (HTTP)
- To a fully distributed and dynamic web of interacting servers and software entities, possibly mobile, deployed at a global scale, serving a variety of terminals with widely differing capabilities

Next-generation Internet Services



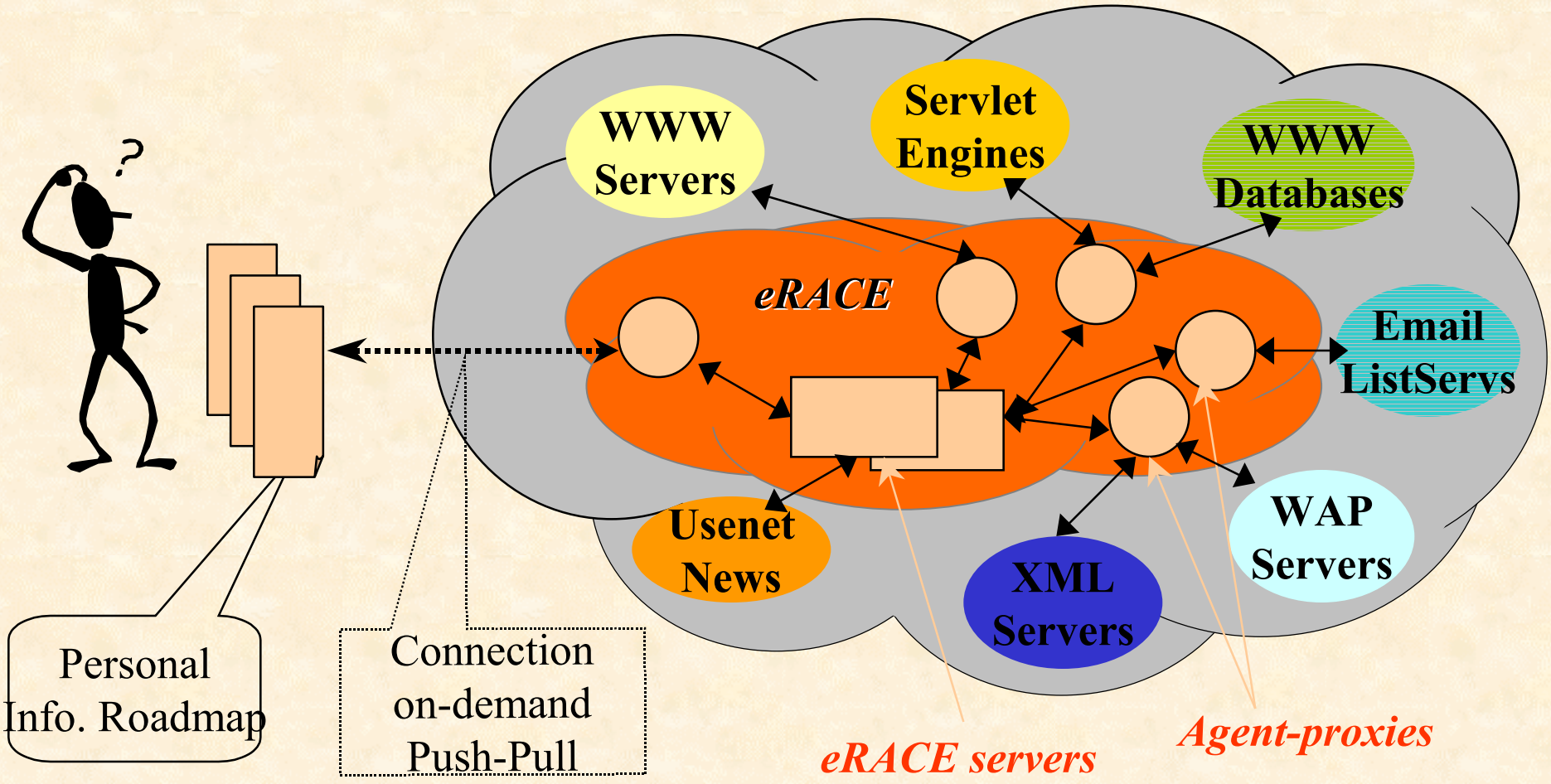
Motivation

- Develop systems and service-infrastructures that enable:
  - Info. Dissemination adaptable to user priorities & connection modalities: content adaptation, push-pull.
  - Easy and dynamic composition of new services: content aggregation
  - Composition of Web services, portals, mobile services.
  - Explicit Management of QoS and Pricing.
  - Incremental processing and communications scalability.
  - Robustness to peak loads.

Outline of the Presentation

- Context and Motivation
- **eRACE Project: an Overview**
- WebRACE: Design & Implementation
- User Interface
- Conclusions and Future Work

eRACE Infrastructure Overview



eRACE Project Goals

To develop an infrastructure that:

- Collects, transforms, customises and personalizes information from heterogeneous sources on a continuous basis, according to user interests.

- Selectively feeds information to users adapting to:
  - User interests and priorities.
  - The urgency & relevance of collected information.
  - Available connection modalities, terminal devices and preferred information-access modes.

- Provides a user-centric view of the global information space by aggregating customised content and using a simple information provision paradigm.

eRACE Project Goals

To develop an infrastructure that:

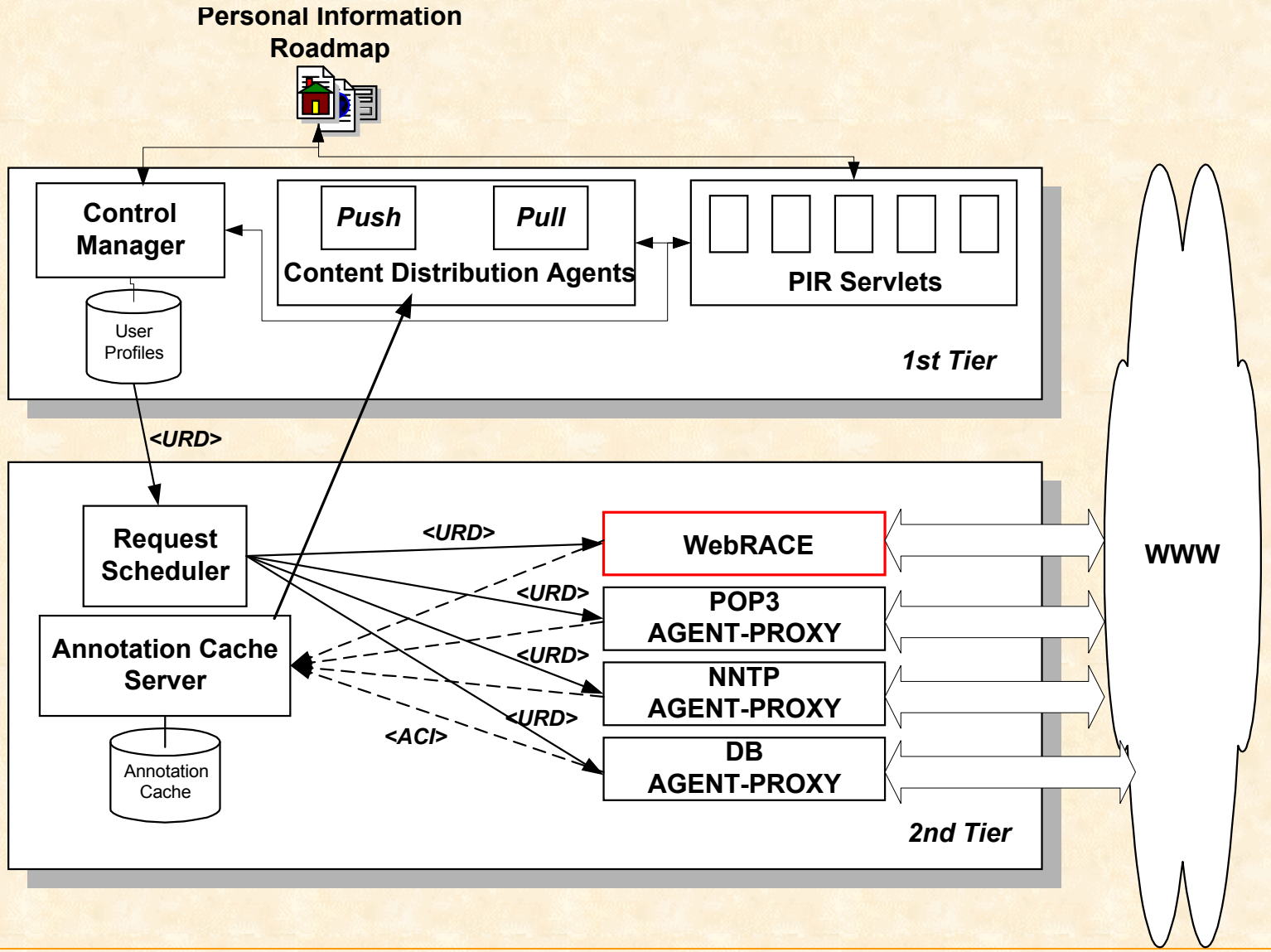
- Is incrementally scalable and can be distributed to different machines.
- Exposes policies of scheduling user-requests, QoS, garbage-collection.
- Enables the easy development of new services and re-targeting to new terminals.

eRACE System Architecture

Two-tiered Architecture:

- Tier 1:
  - Control Manager
  - Content Distribution Agents
  - Personal Information Roadmap Servlets
- Tier 2:
  - Request Scheduler
  - Distributed Agent-Proxies (WWW, NNTP, POP3, etc.)
  - Annotation Cache Server

eRACE System Architecture



eRACE Information Architecture

- Information Architecture: describes the data representations of *state information* and *information exchanges*, in terms of XML DTD's:
  - Control Manager DTD: account, authentication, connection-status.
  - User Profile DTD: personal data, notification addresses, resource information (URD).
  - Annotation Cache Interface DTD: meta-information for collected content.

- Data sharing between various components is done using pass-by-value semantics (messages and events).

- This choice enables us to decouple and physically separate components.

Outline of the Presentation

- Context and Motivation
- eRACE Infrastructure: An Overview
- **WebRACE: Design & Implementation**
- User Interface
- Current Status and Future Work

WebRACE

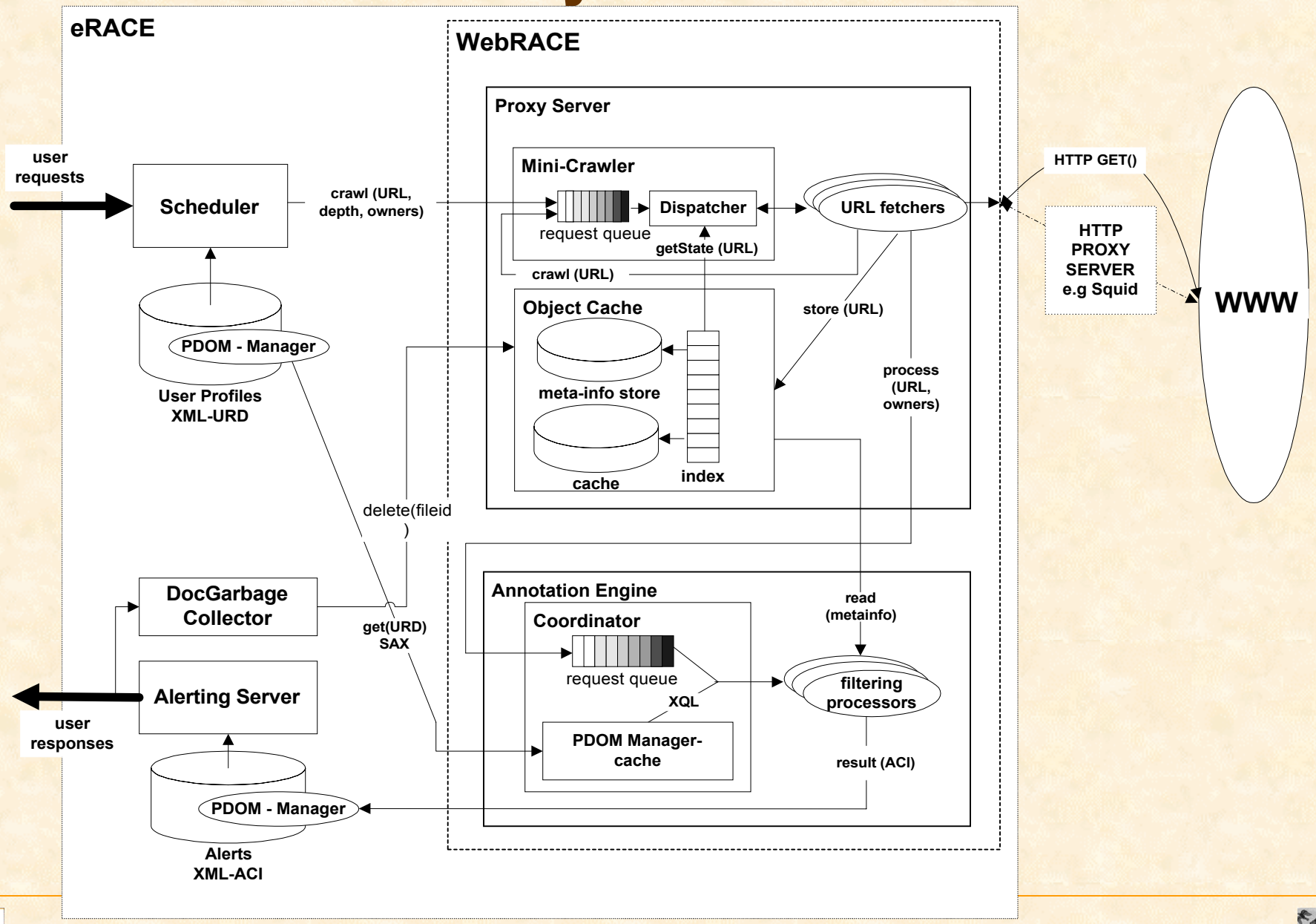
- **WebRACE:** an agent-proxy that collects, processes and caches content from information sources on the WWW, accessible through HTTP/1.0 and HTTP/1.1

- WebRACE Components:
  - Mini-Crawler
  - Annotation Engine
  - Object Cache

Design and Implementation Challenges

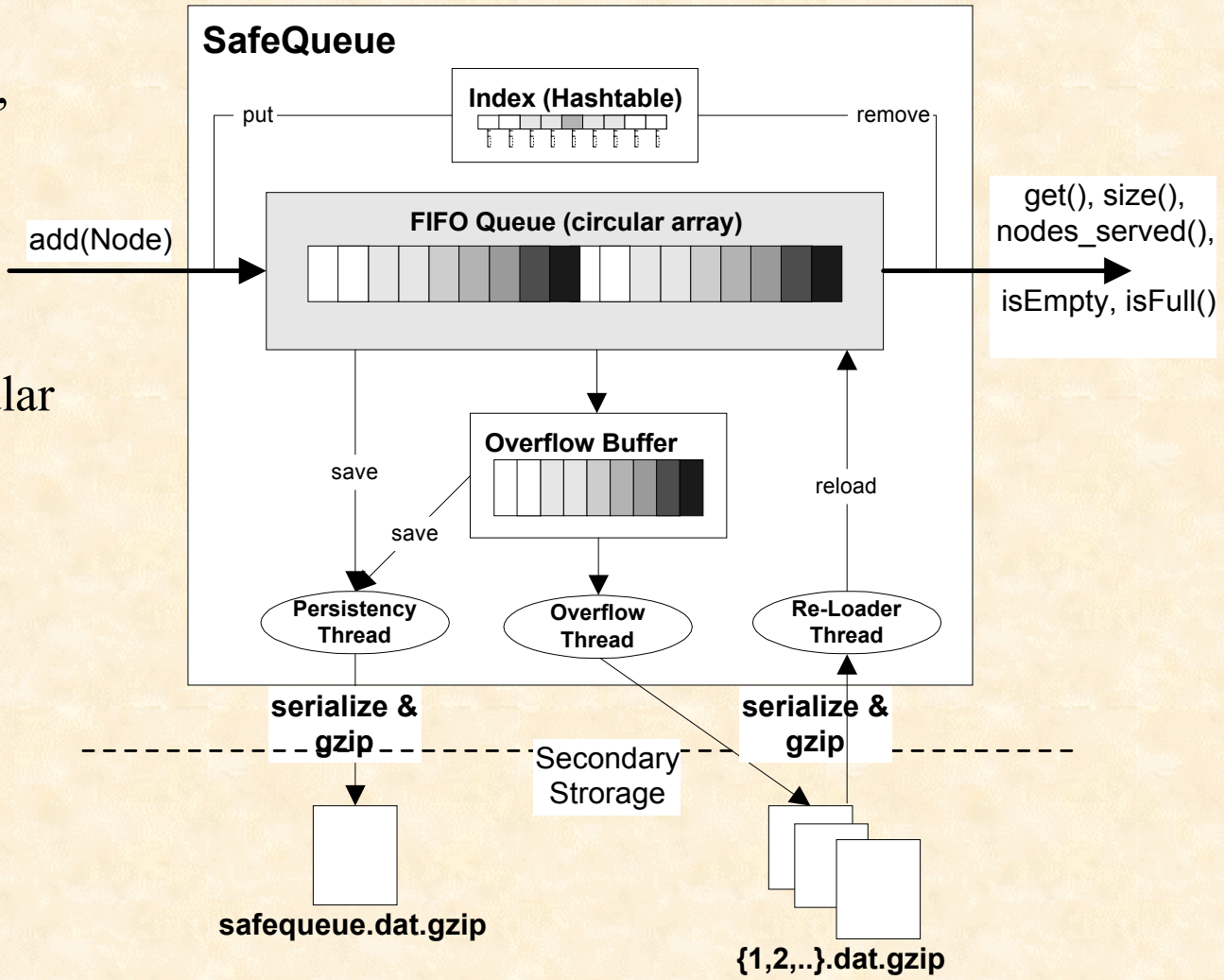
- User-driven crawler – no fixed "seed" list.
- Crawling to capture frequently updated sites:
  - Short-term time constraints.
  - Multiple versions indexed and kept in store.
- Massively personal and site-specific crawling:
  - Coalescing personalized Web-tracking for many users.
  - Performance scalability w.r.t. increasing user-base.
  - Built-in support for explicit QoS management.
- Java:
  - OO, Multithreaded, support for Network Programming, Code Mobility.
  - High-performance, robustness.
  - Memory bounded w.r.t. crawl size.

WebRACE System Architecture



WebRACE Data Structures: SafeQueue

- ***SafeQueue*** is a "*thread-safe*" and "*persistent*" FIFO queue implemented in JAVA.
- **SQ** is implemented as a circular array of ***QueueNodes***.
- ***QueueNodes*** are any type of JAVA object.

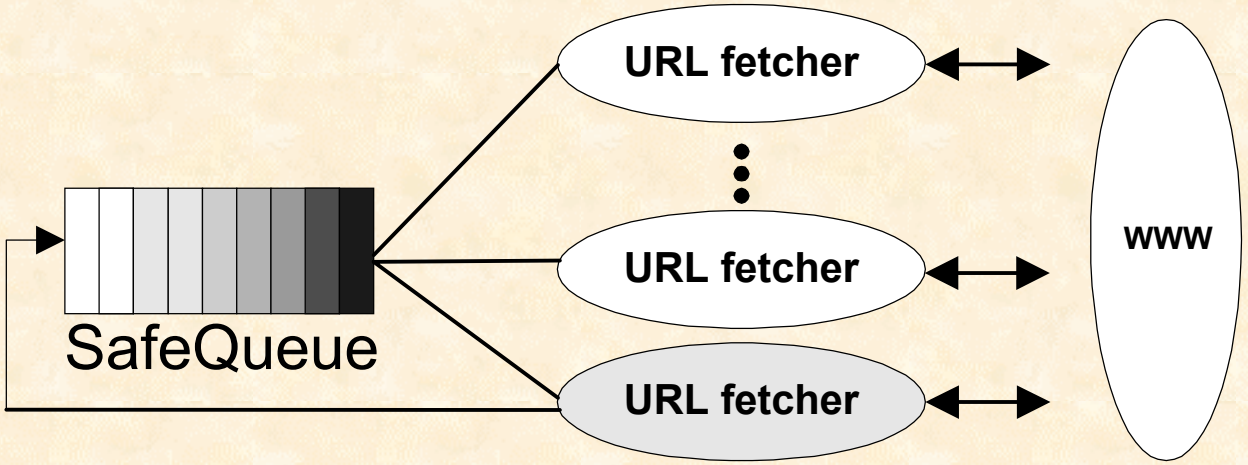


WebRACE Mini-Crawler

- Receives crawling instructions from the *eRACE Request Scheduler*.

- Components:
  - URLQueue
  - URLFetcher, Extractor & Normalizer
  - Object Cache

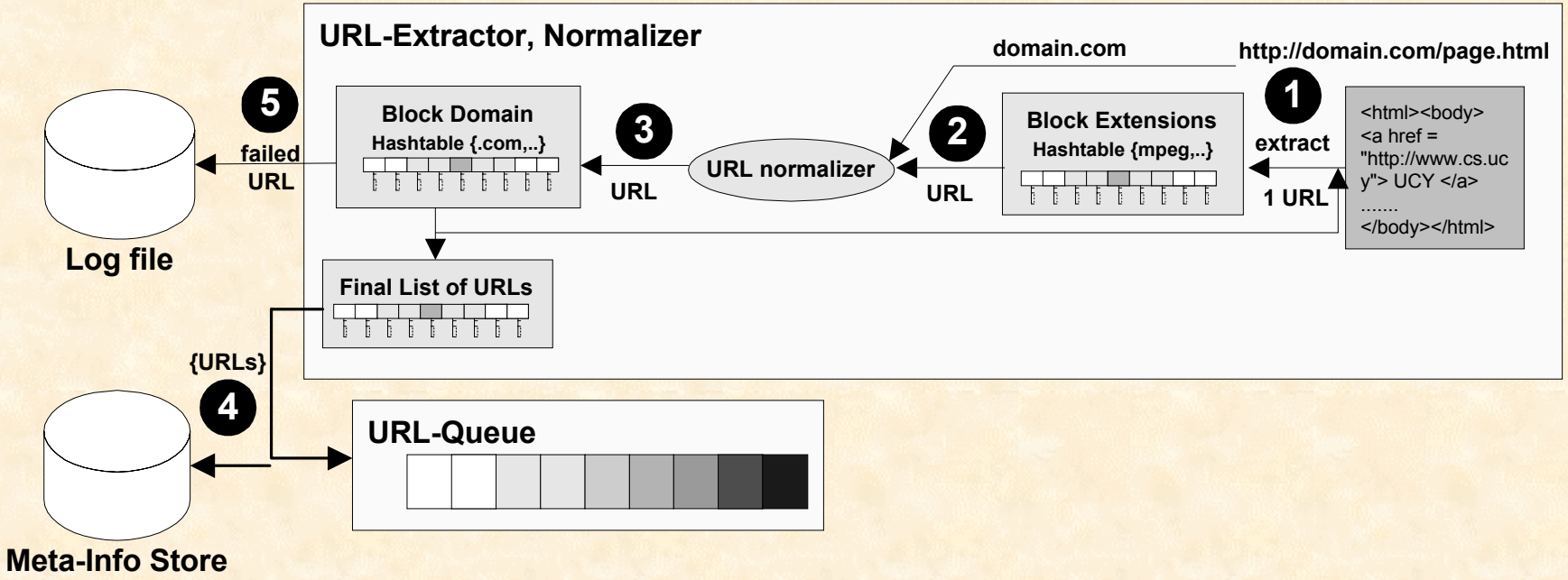
URL Fetchers



URLFetcher:

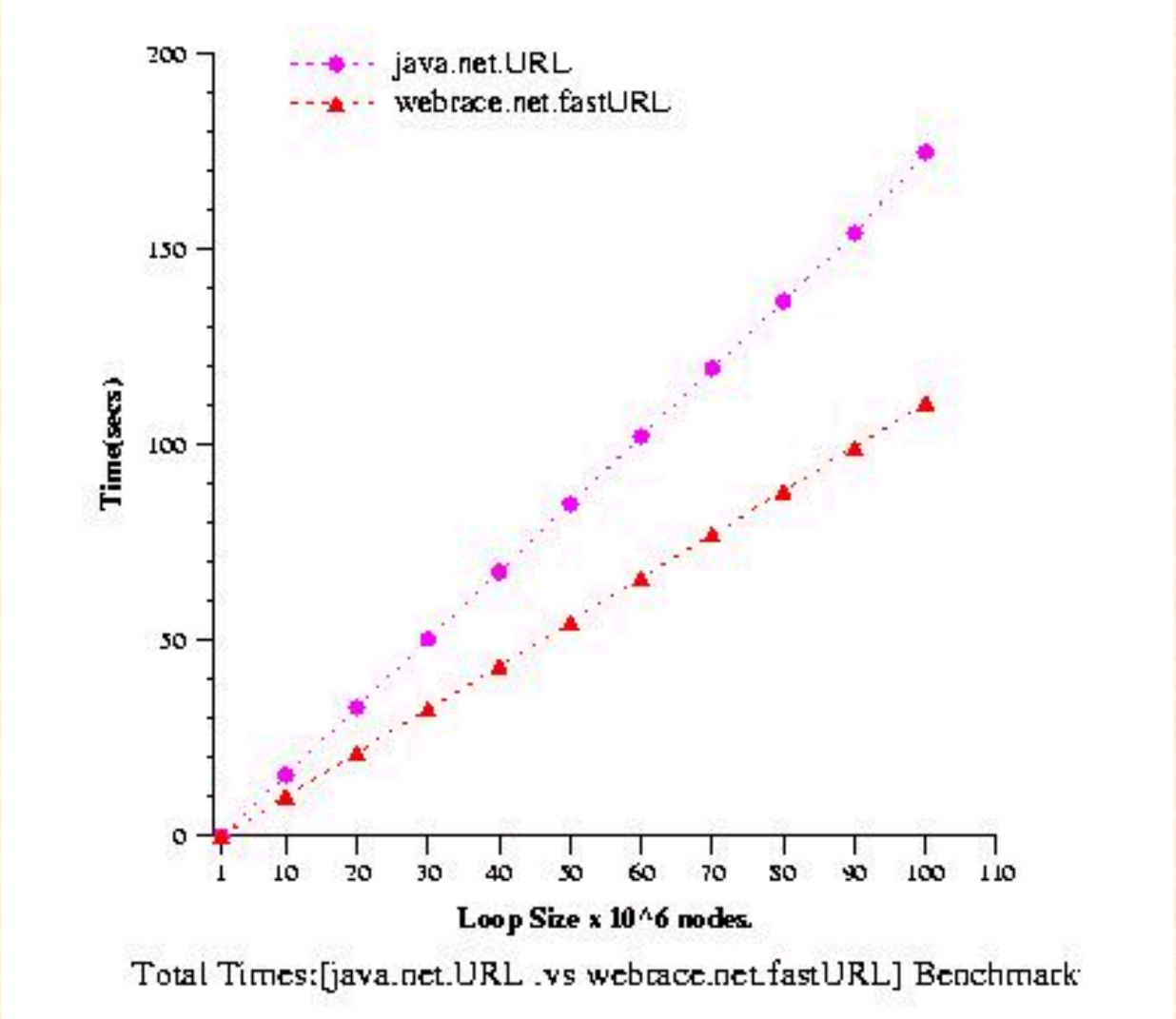
- Handles HTTP connections and URL extraction.
- Support for multiple URLFetcher threads; concurrency is configurable.
- Support for the Robots Exclusion Protocol.
- Support for blocking the crawling of particular domains or URL's.
- 6-step pipe for URL extraction.

URL Extraction & Normalization

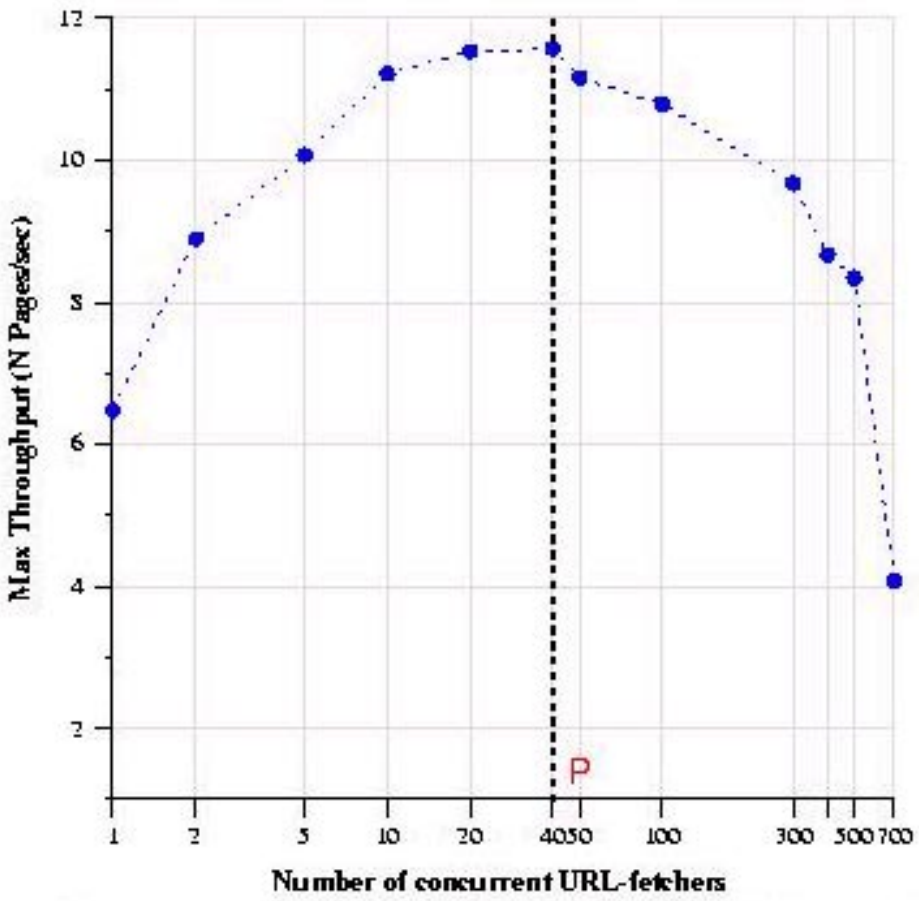


- URL extraction and normalization pipe requires approx. 300 ms for a 70KB HTML page, on a Sun E250 server.
- Implemented various optimizations in JAVA core libraries:
  - java.net.Socket
  - java.net.URL

webrace.net.URL Performance



URLFetcher Throughput Degradation



Number of Concurrent URL-fetchers executing in WebRACE (normal-log scale)

Object Cache

- Stores collected Web content for further processing.
- Caches crawling information to accelerate re-crawls.
- Components:
  - Index
  - Meta-Info Store
  - Object Store
- Meta-Info Store contents (encoded in XML)
  - URL address of the corresponding document
  - IP address of the origin Web server
  - Document size
  - Last-Modified field returned by the HTTP protocol during download
  - HTTP response header
  - Extracted and normalized links contained in the document

Meta-Info Store Example

```
< webrace:url>http://www.cs.ucy.ac.cy/~epl121/< /webrace:url>
< webrace:ip>194.42.7.2< /webrace:ip>
< webrace:kbytes>1< /webrace:kbytes>
< webrace:ifmodifiedsince>989814504121< /webrace:ifmodifiedsince>
<webrace:header>
  HTTP/1.0 200 OK
  Server:  Netscape-FastTrack/2.01
  Date:  Fri, 11 May 2001 13:50:10 GMT
  Accept-ranges:  bytes
  Last-modified:  Fri, 26 Jan 2001 21:46:08 GMT
  Content-length:  1800
  Content-type:  text/html
< /webrace:header>
<webrace:links>
  http://www.cs.ucy.ac.cy/Computing/labs.html
  http://www.cs.ucy.ac.cy/
  http://www.cs.ucy.ac.cy/helpdesk
< /webrace:links>
```

Meta-Info Store Functionality

URLFetcher Algorithm:

1. Retrieve a QueueNode from URLQueue; extract URL.
2. Fetch URL and analyze HTTP-header. If OK, proceed.
3. Download document body and store it in main memory.
4. Extract and normalize links.
5. Compress and save the document in the Object Cache.
6. Generate the meta-information and save it in the Meta-Info Store.
7. Update the Object Cache Index.
8. Notify Annotation Engine.
9. Add extracted URL's to the URL Queue.

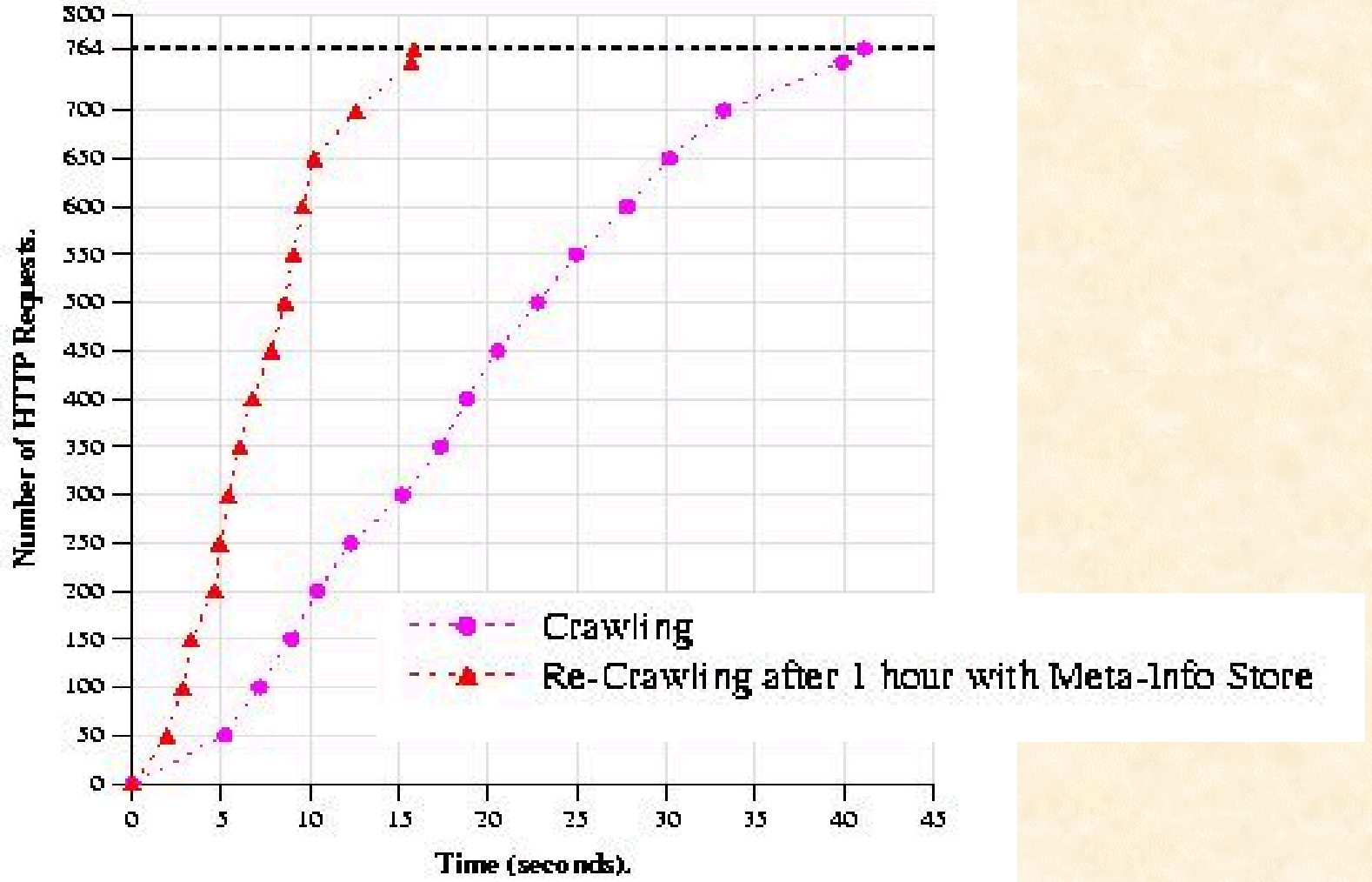
Meta-Info Store Functionality (ctd)

Using the Meta-Info Store to accelerate crawling

2. If URL is in the Object Cache, load its Meta-Info file.

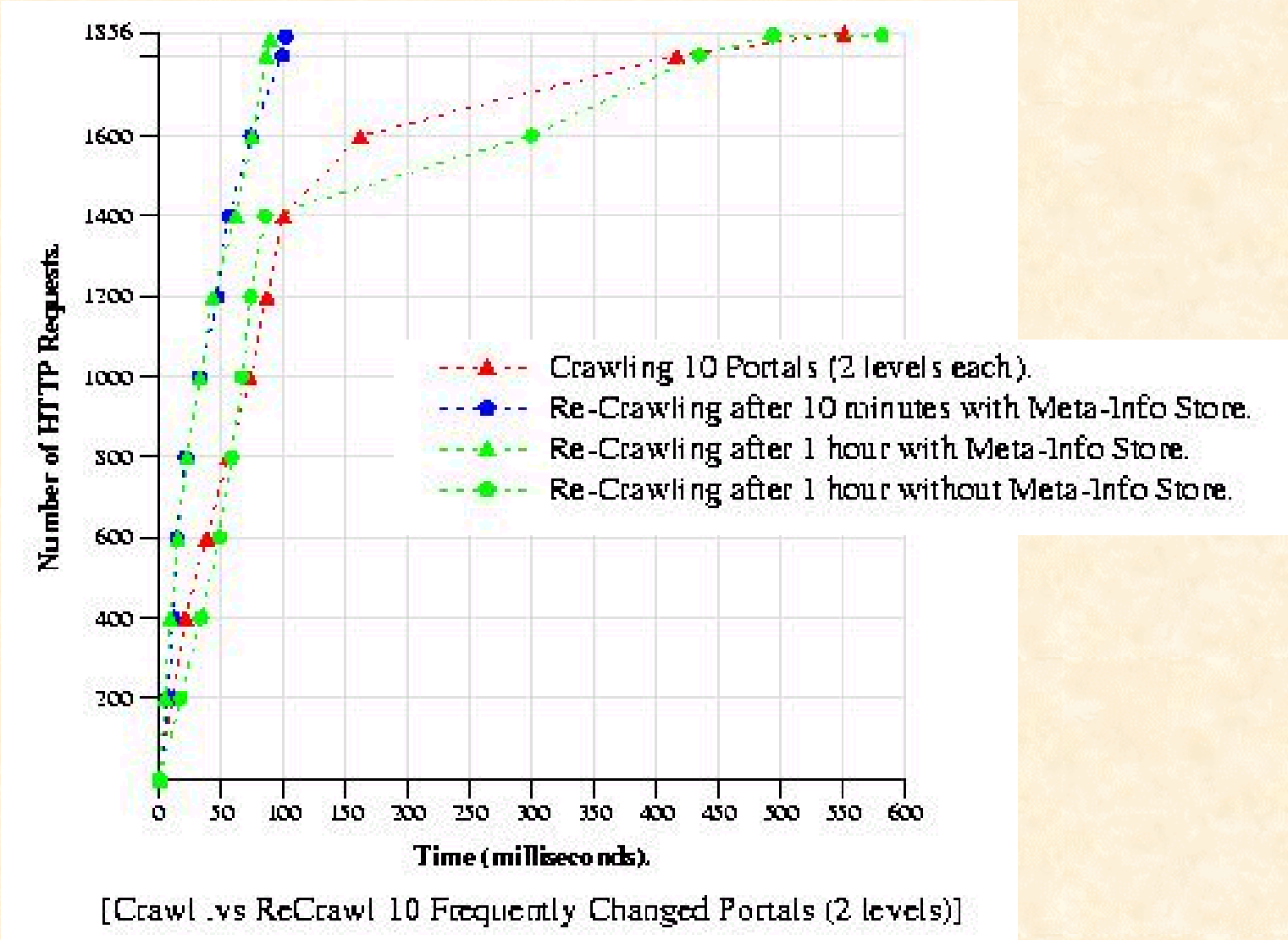
3. If the URL is not in the Object Cache, download it. Else, use the Meta-Info time-stamp to check with the origin server whether the document has changed since. If yes, download the document, store it in main memory and proceed to step 4. Else, extract links from Meta-Info file and proceed to step 8.

Caching Crawling Information



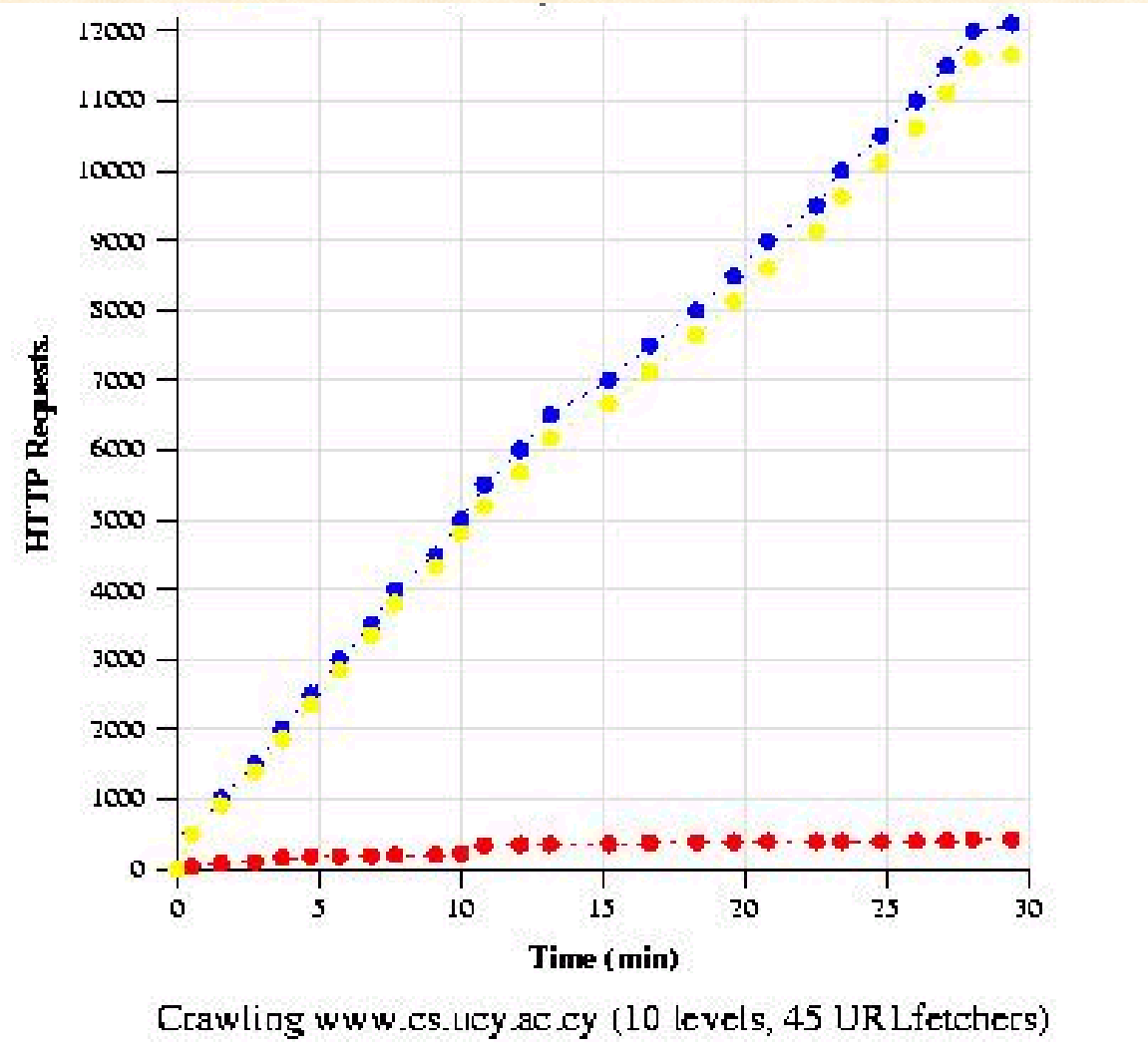
[Crawl vs. Re-crawl, 25 US Universities (2 levels)]

Caching Crawling Information



[Crawl .vs ReCrawl 10 Frequently Changed Portals (2 levels)]

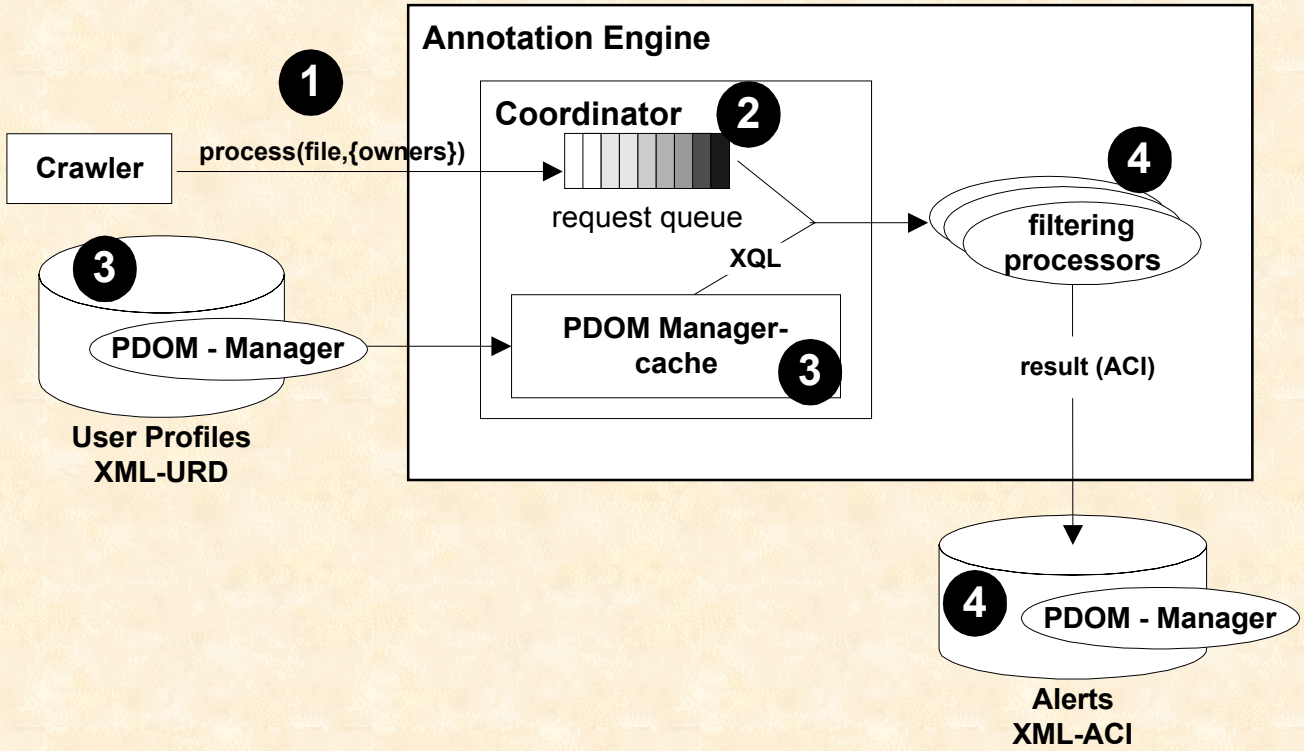
Caching Crawling Information



WebRACE Annotation Engine

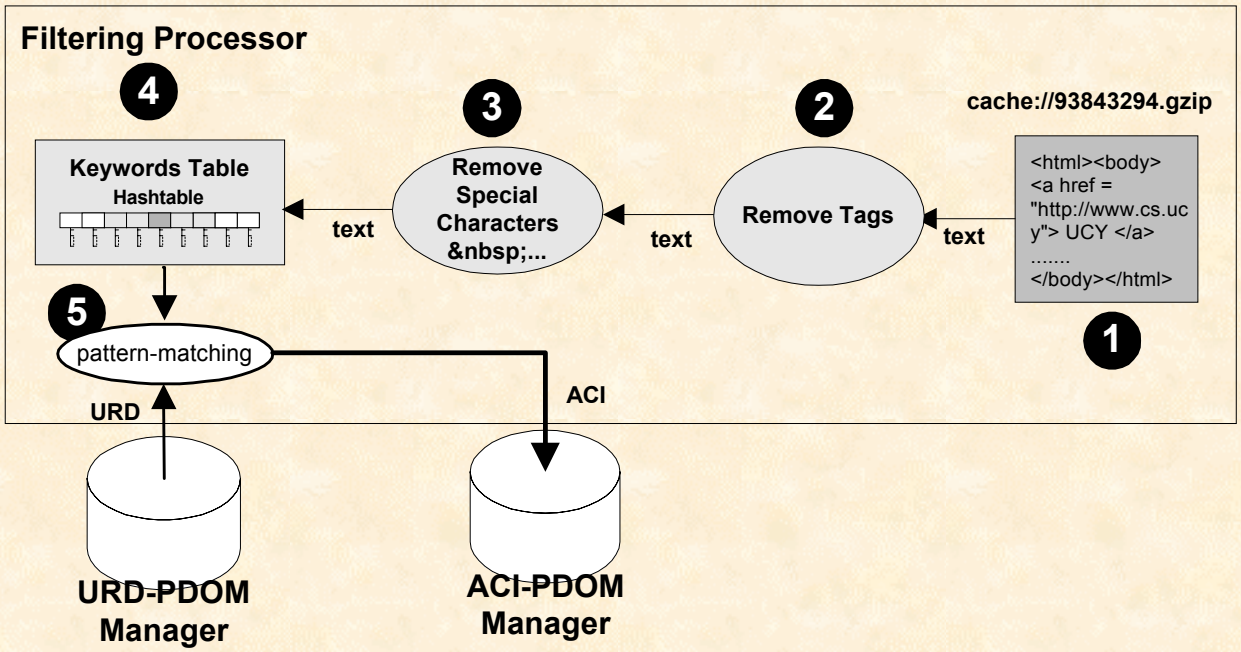
- Processes documents downloaded and stored in the Object Cache.
- Classifies documents according to eRACE profiles stored in an XML database (URD's).
- Meta-information produced by the AE is stored in an XML Annotation Cache as annotation linked to the cached document.
- Annotations are processed by the Content Distribution Agents of eRACE to produce user alerts.
- Irrelevant pages are marked as garbage and collected.
- In contrast to general-purpose Search Engines, the AE processes and indexes collected documents "on-the-fly."

WebRACE Annotation Engine



- Annotations are stored in a single XML-encoded document, managed by a persistent DOM data manager and XQL query processor by GMD (PDOM).
- PDOM is thread-safe, persistent and enables main-memory caching of XML nodes, facilitating fast searches in the DOM tree.

Filtering Processor



- 6-step pipe, takes on the average 200 ms to calculate the ACI's for a 70KB Web page with three potential recipients.

ACI Example

```
<aci owner="eleni" extension="html" format="html"
    relevance="18" updatetime="9787695000" filesize="2000">
    <uri>http://www.cs.ucy.ac.cy/default.html</uri>
    <urgency urgent="1"/>
    <docbase>969890.gzip</docbase>
    <expired expir="false">
    <summary>Document keywords…</summary>
</aci>
```

WebRACE Implementation

Java

- Platform independence
- Strong typing
- Multi-threading
- Automatic memory management

Mitsubishi Concordia Mobile Agents Platform

- Java support
- Support for distributed operations and code mobility
- Persistence
- Messaging, event programming and coordination

<XML> **W3C eXtensible Markup Language (XML)**

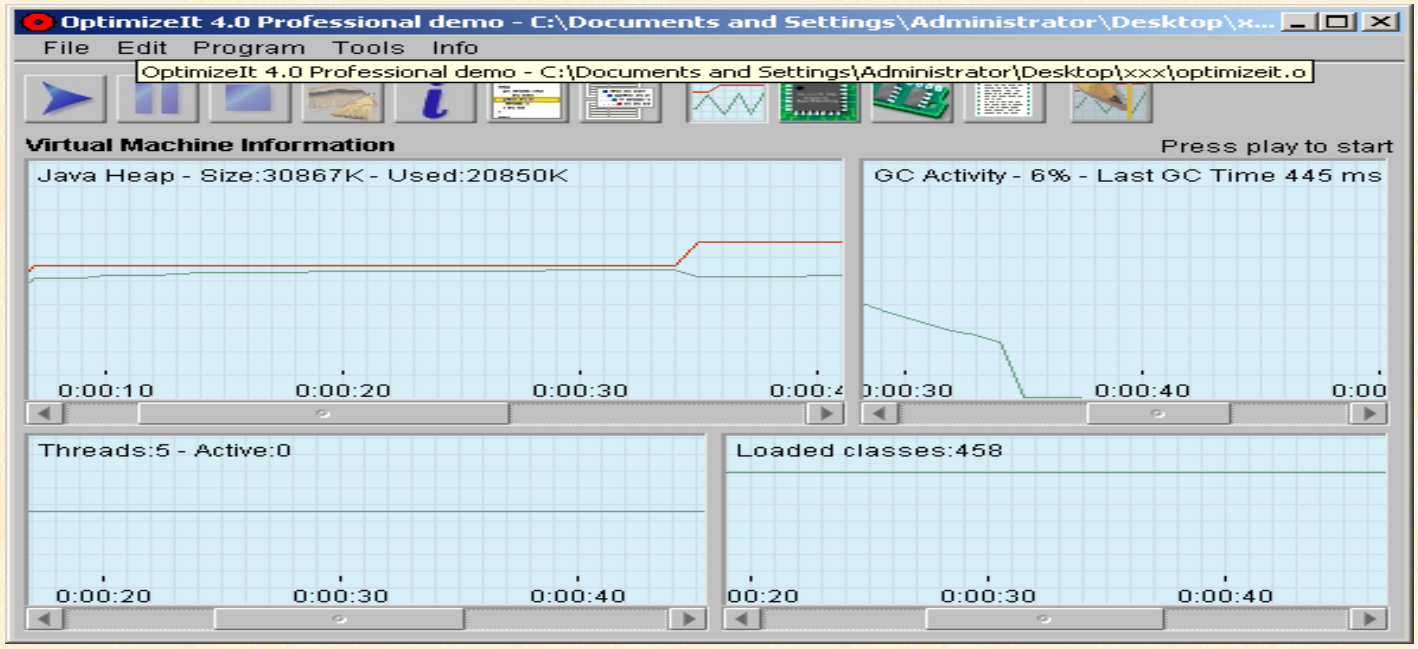
- Self-descriptive format for communication between components (decoupling)
- Extensible and “open” grammars to specify services, user profiles, etc.
- Reusability of tools (Java classes for XML parsers)

Java Servlets

- Interface programming
- Java support

Fine-Tuning Performance

- We are using *Intuitive Systems' OptimizeIt* for measuring various performance properties of both our Crawler and the Filtering Processor Engine.

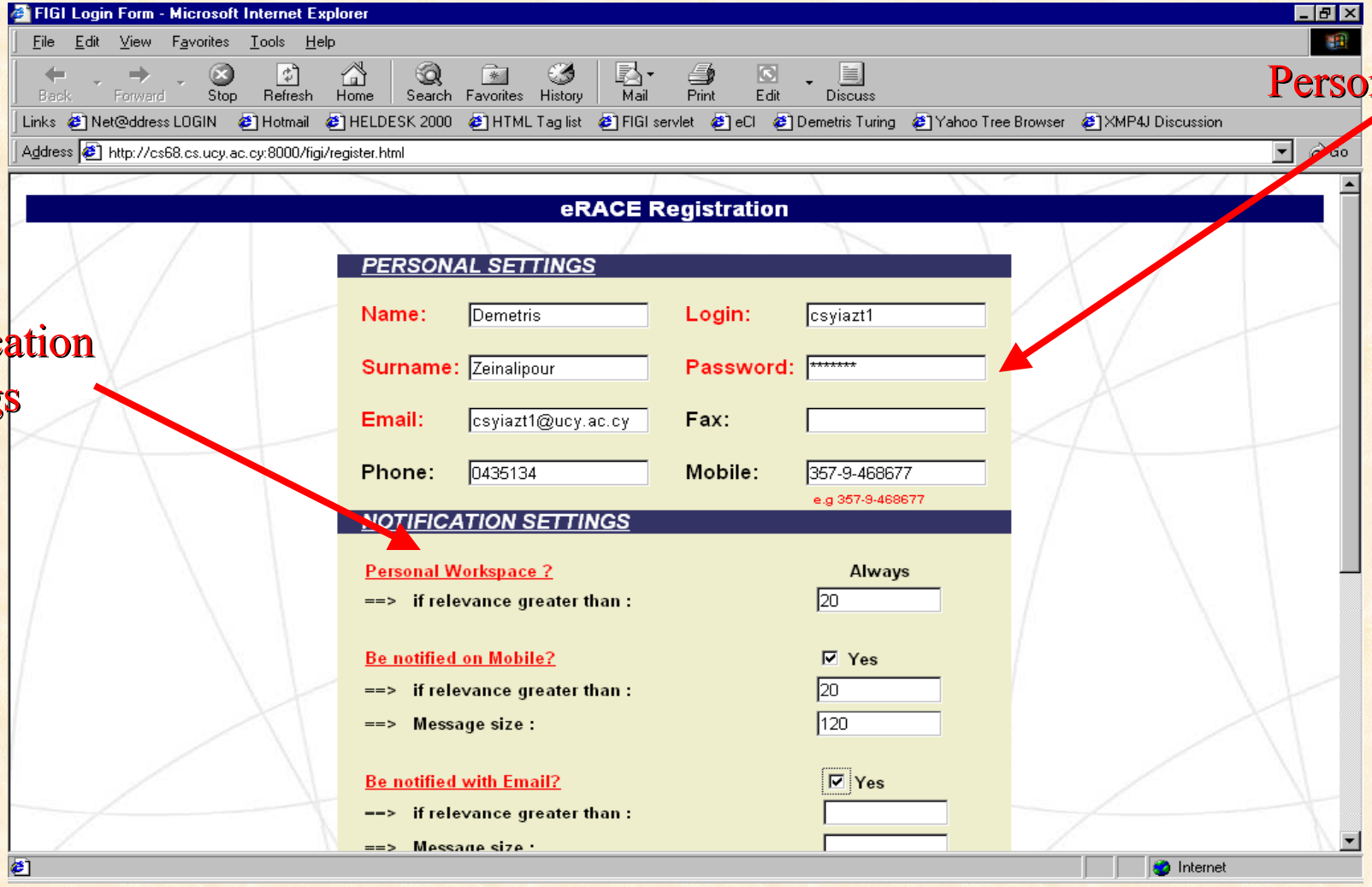


Outline of the Presentation

- Context and Motivation
- eRACE Infrastructure: An Overview
- WebRACE: Design & Implementation
- **User Interface: Personal Information Roadmap**
- Current Status and Future Work

eRACE UI: Personal Information Roadmap

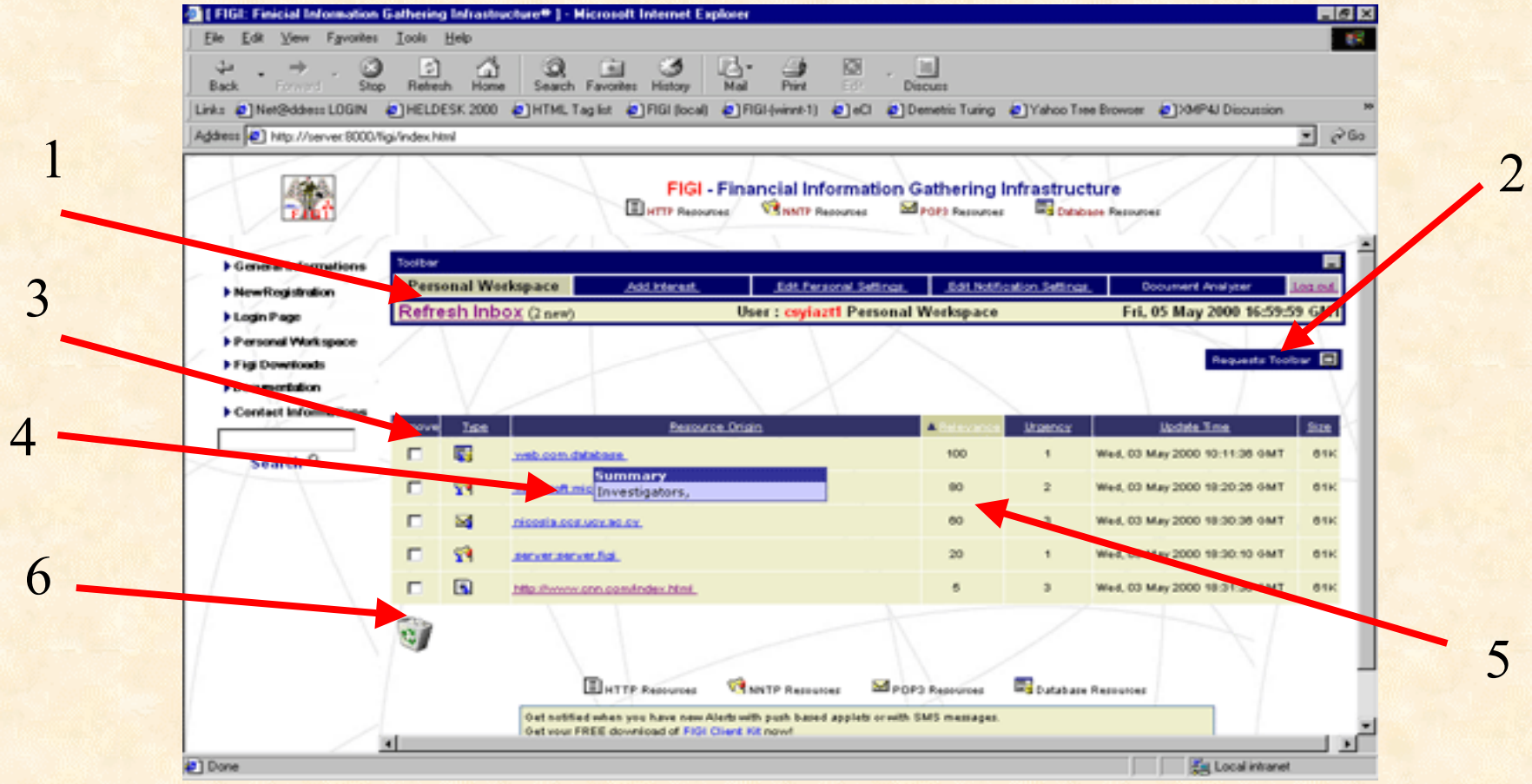
User Registration



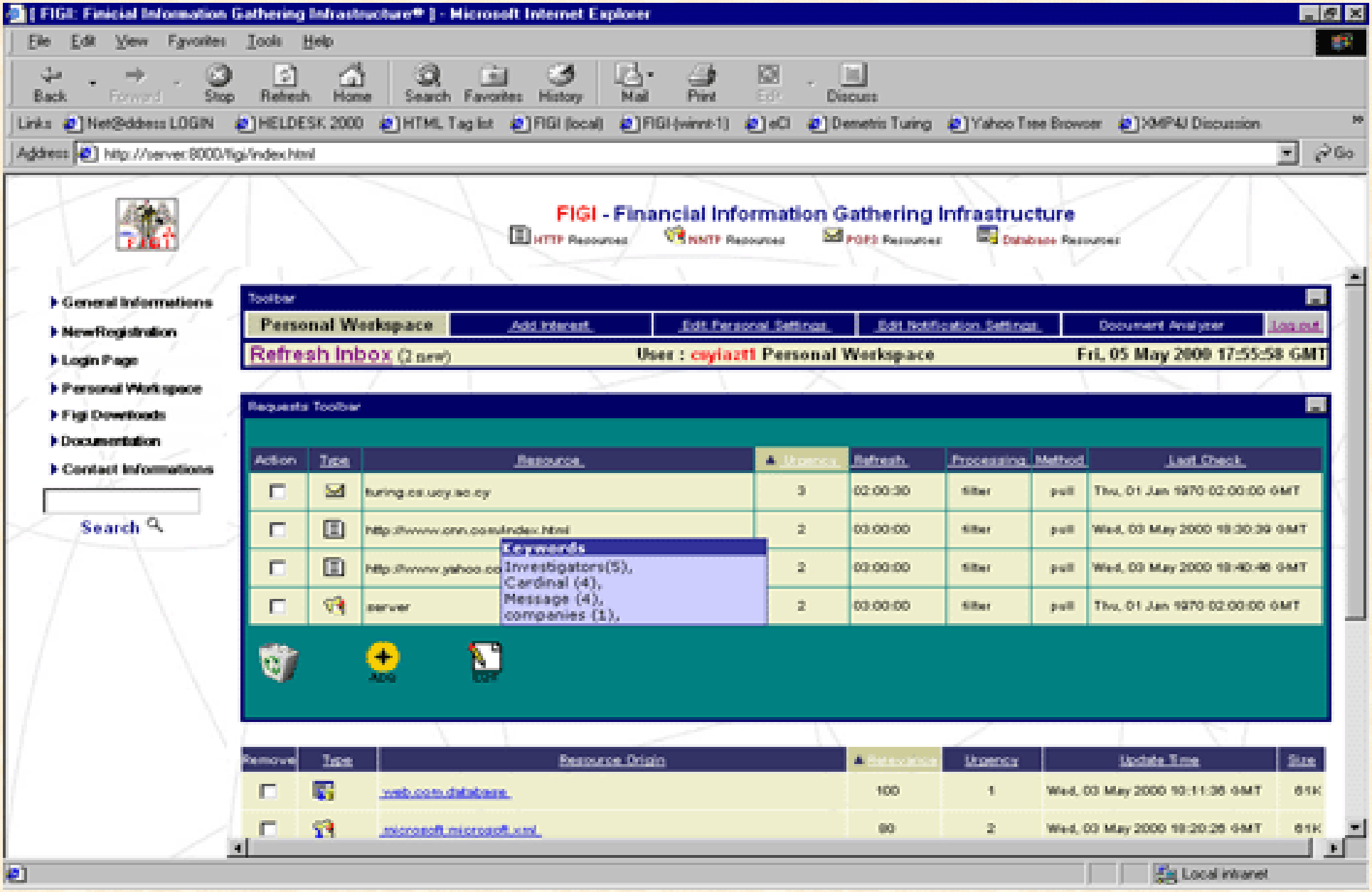
Personal Settings

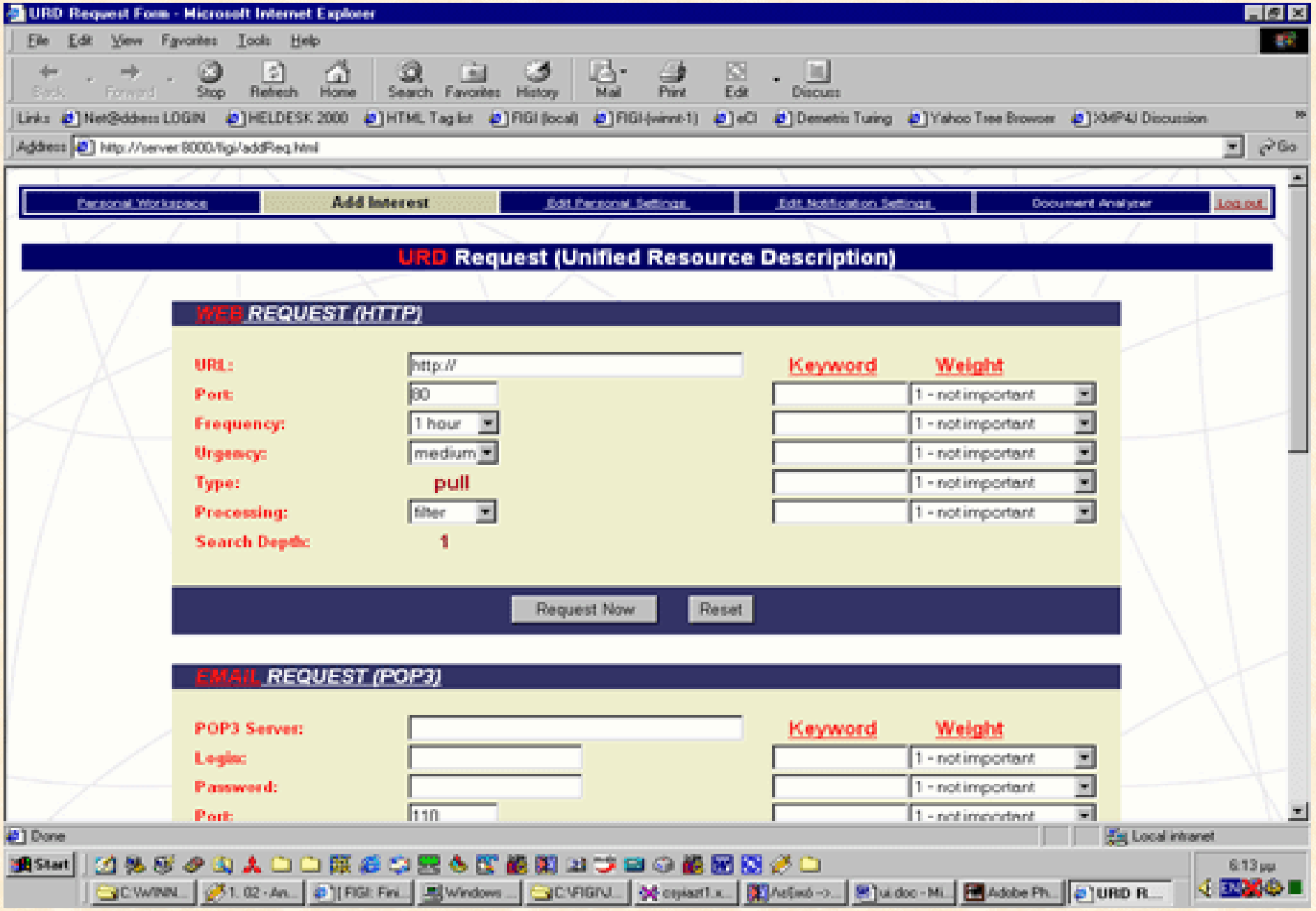
Notification Settings

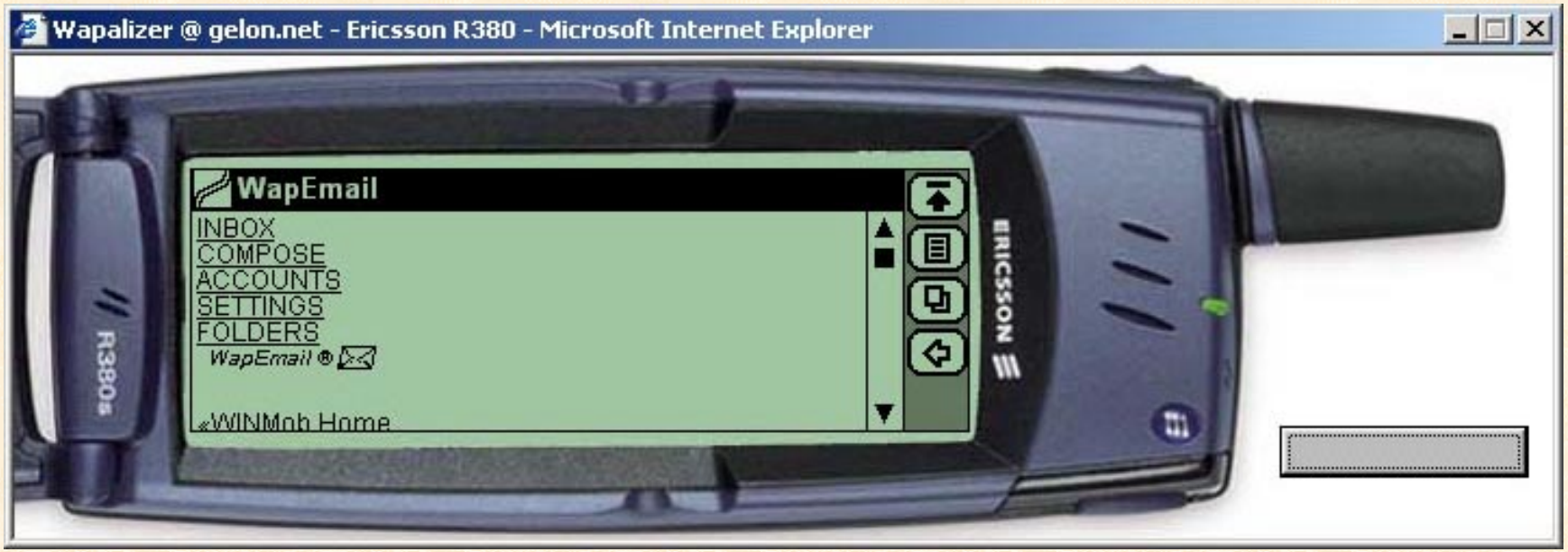
PIR Functionality



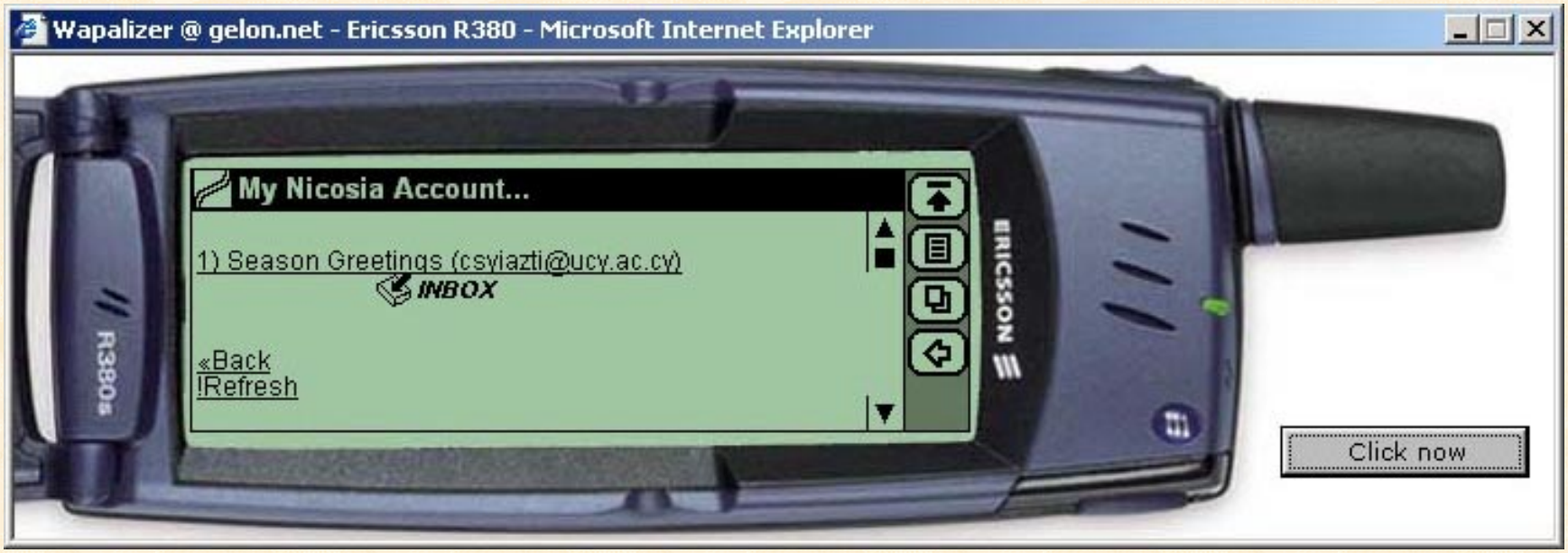
1. Maximized Navigation Toolbar
2. Minimized Requests Toolbar
3. Gathered Information Matrix
4. Summary Window
5. Sort By Column option
6. Recycle Bin


[FIGI: Finacial Information Gathering Infrastructure®] - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Links Net@ddress LOGIN HELDESK 2000 HTML Tag list FIGI (local) FIGI (winnt-1) eCl Demetris Turing Yahoo Tree Browser XMP4J Discussion
Address http://server:8000/figi/index.html
FIGI - Financial Information Gathering Infrastructure
HTTP Resources
NNTP Resources
POP3 Resources
Database Resources
General Informations
NewRegistration
Login Page
Personal Workspace
Figi Downloads
Documentation
Contact Informations
Search
Toolbar
Personal Weakspace
Add Interest
Edit Personal Settings
Edit Notification Settings
Document Analyzer
Log out
Refresh Inbox (2 new)
User : cyiazzi Personal Weakspace
Fri, 05 May 2000 17:55:58 GMT
Requests Toolbar
Action
Type
Resource
Urgency
Refresh
Processing
Method
Last Check
turing.cs.ucy.ac.cy
3
02:00:30
filter
pull
Thu, 01 Jan 1970 02:00:00 GMT
2
03:00:00
filter
pull
Wed, 03 May 2000 18:30:39 GMT
2
03:00:00
filter
pull
Wed, 03 May 2000 18:40:48 GMT
server
2
03:00:00
filter
pull
Thu, 01 Jan 1970 02:00:00 GMT
Keywords
Investigators(5),
Cardinal (4),
Message (4),
companies (1),
ADD
COPY
Remove
Type
Resource Origin
Relevance
Urgency
Update Time
Size
web.com.database
100
1
Wed, 03 May 2000 18:11:38 GMT
61K
microsoft.microsoft.xml
80
2
Wed, 03 May 2000 18:20:26 GMT
61K
Local intranet

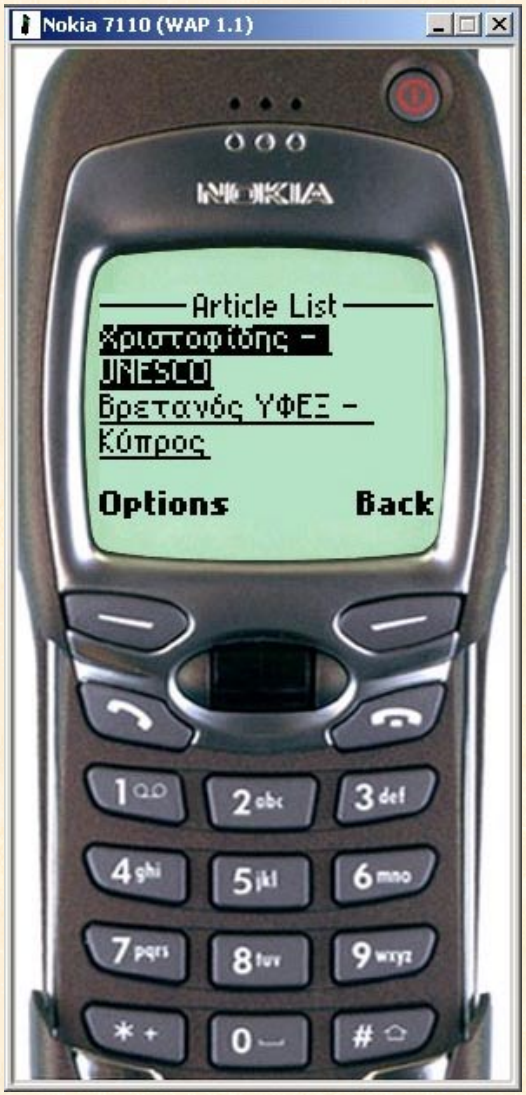

URD Request Form - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Links Net@ddress LOGIN HELDESK 2000 HTML Tag list FIGI (local) FIGI (winnt-1) eCl Demetrin Turing Yahoo Tree Browser XMPAJ Discussion
Address http://server:8000/figi/addReq.html
Go
Personal Workspace
Add Interest
Edit Personal Settings
Edit Notification Settings
Document Analyzer
Log out
URD Request (Unified Resource Description)
WEB REQUEST (HTTP)
URL:
http://
Port:
80
Frequency:
1 hour
Urgency:
medium
Type:
pull
Processing:
filter
Search Depth:
1
Keyword
Weight
1 - not important
Request Now
Reset
EMAIL REQUEST (POP3)
POP3 Server:
Login:
Password:
Port:
110
Done
Local intranet
Start

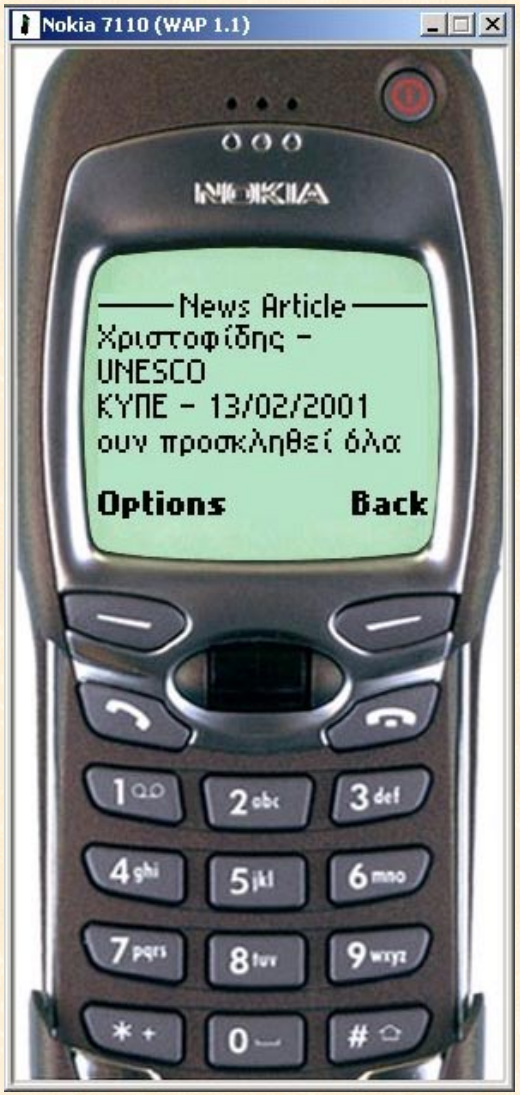

Wapalizer @ gelon.net - Ericsson R380 - Microsoft Internet Explorer
WapEmail
INBOX
COMPOSE
ACCOUNTS
SETTINGS
FOLDERS
WapEmail ®
«WINMob Home
ERICSSON
R380s

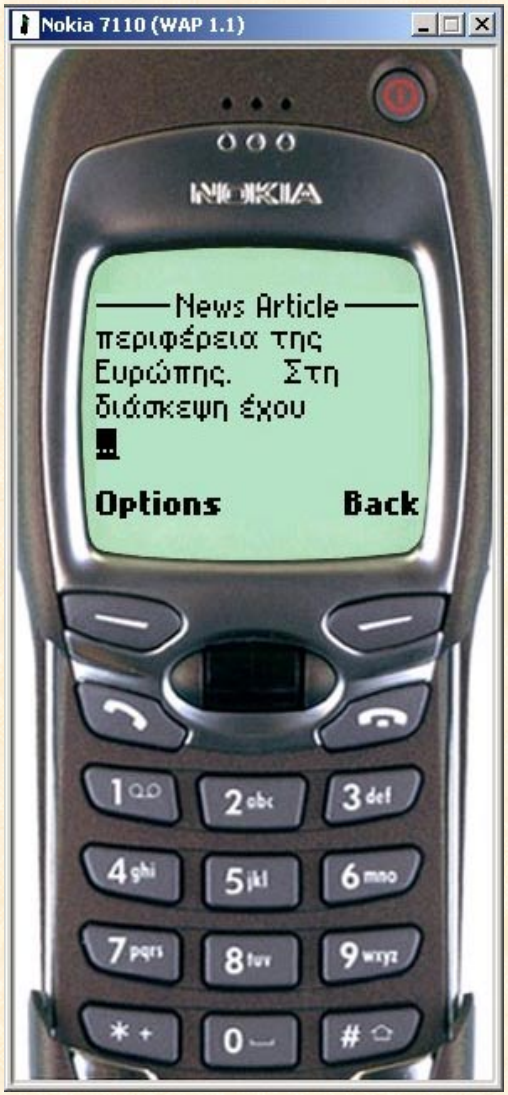

Wapalizer @ gelon.net - Ericsson R380 - Microsoft Internet Explorer
Select Account
1) My Nicosia Account
2) My University Account
3) My WinMob Account
INBOX
«Back
Click now
ERICSSON
R380s


Wapalizer @ gelon.net - Ericsson R380 - Microsoft Internet Explorer
My Nicosia Account...
1) Season Greetings (csyiazti@ucy.ac.cy)
INBOX
«Back
!Refresh
ERICSSON
R380s
Click now


Wapalizer @ gelon.net - Ericsson R380 - Microsoft Internet Explorer
Compose Email
To:
csyiazti@ucy
Save above Address? Yes
Subject:
hello
ERICSSON
R380s
Click now


Nokia 7110 (WAP 1.1)
NOKIA
Article List
Χριστοφίδης -
UNESCO
Βρετανός ΥΦΕΞ -
Κύπρος
Options
Back


Nokia 7110 (WAP 1.1)
NOKIA
News Article
Χριστοφίδης -
UNESCO
ΚΥΠΕ - 13/02/2001
ουν προσκληθεί όλα
Options
Back


Nokia 7110 (WAP 1.1)
NOKIA
News Article
περιφέρεια της
Ευρώπης. Στη
διάσκεψη έχου
Options
Back

Conclusions & Summary

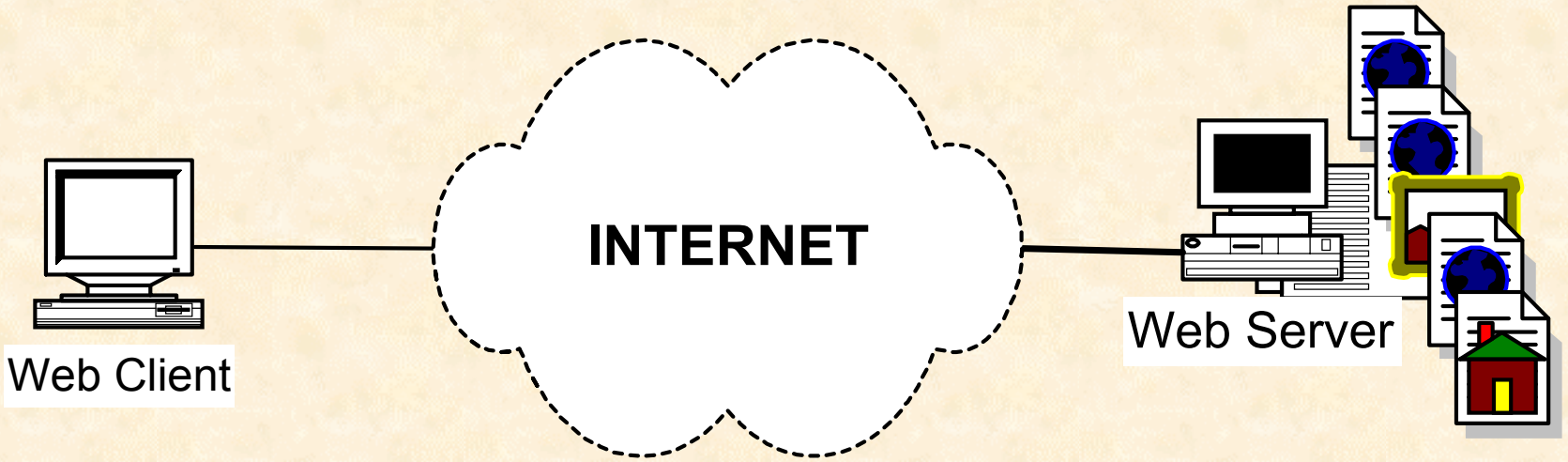
- An architecture for personalized and customisable Information Dissemination.
- An infrastructure to develop new services.
- A platform to investigate performance, scheduling, and QoS issues in the context of Internet services.
- Crawlers as component of Internet middleware.
- JAVA as platform for building user-driven crawlers.

Current Status & Future Work

- Finalization of the mailRACE proxy and its incorporation into a Wap gateway for email.
- Using WebRACE to generate dynamically & publish WML content.
- Description of services and service composition: XML, XQL?

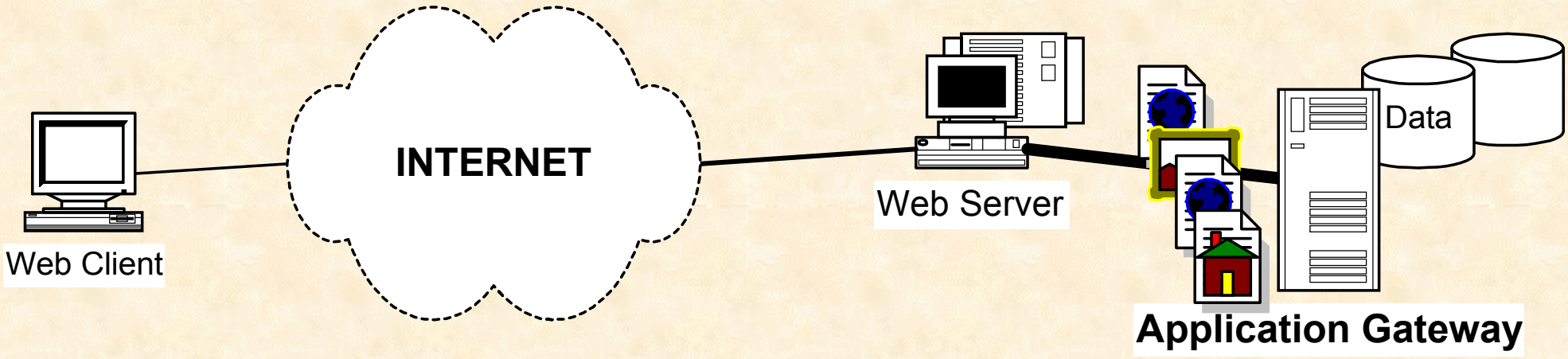
Backup Slides

From Web Servers to Web Services: I



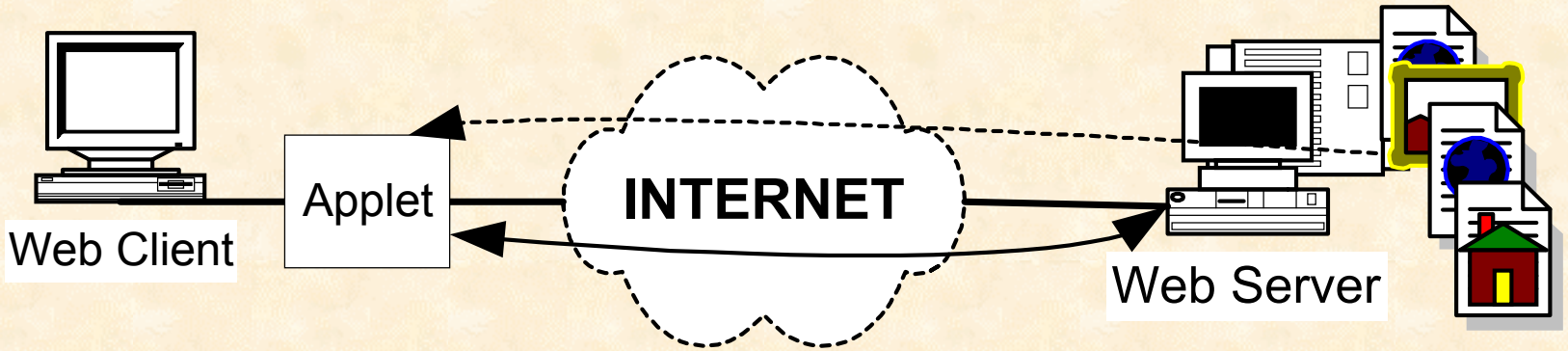
- *Typical client-server model*
- *Web server: repository of multimedia content*

From Web Servers to Web Services: II



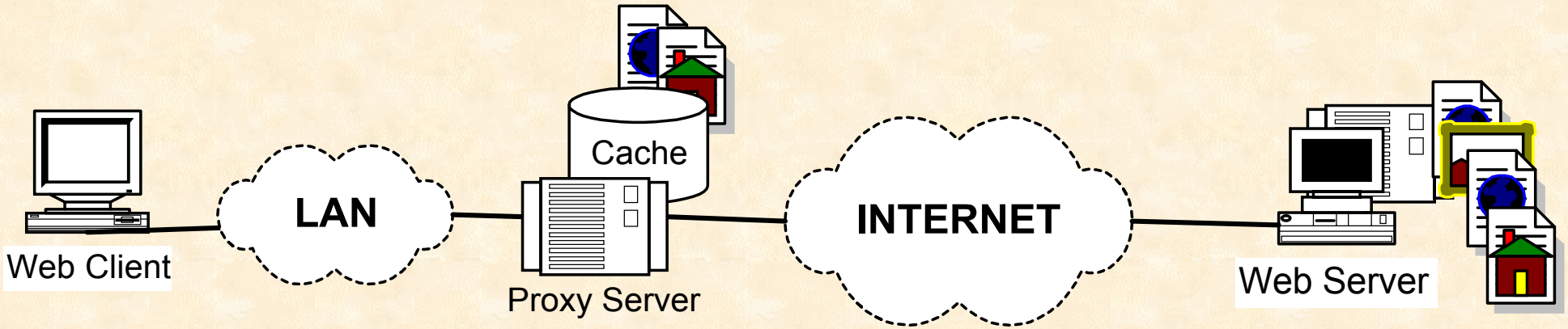
- *Typical client-server model*
- *Web server: provider of dynamic content*

From Web Servers to Web Services: III



- *Client-server model with dynamically enhanced clients*
- *Web server: repository of content & functionality*

From Web Servers to Web Services: IV



- *Proxy-server model*
- *Web caching, prefetching, information dissemination*
- *Content-Distribution Networks*

eRACE Information Architecture (ctd')

User-profile DTD encodes:

- Personal data
- Notification addresses (email, mobile phone)
- Resource information:
  - Resource description
  - Query options
  - User interests (keywords)
  - Notification Priorities

```
<!ELEMENT personal (name,surname,email,phone?,fax?,mobile?)>
```

eRACE Information Architecture

Users Manager DTD encodes:

- Account & authentication information for eRACE users
- Connection status

```
<!ELEMENT Accounts (Account*)>
   <!ATTLIST Accounts id ID #REQUIRED
                      location CDATA #REQUIRED
                      maxAccounts CDATA #REQUIRED
                    locked (false | true) "false">

                <!ELEMENT      Account  EMPTY>
                <!ATTLIST      Account  id ID #REQUIRED
                                state  (false | true) "true"
                                    docbase CDATA #REQUIRED>
```

eRACE Information Architecture (ctd')

- Unified Resource Description (URD):

```
<!ELEMENT source (uri,      type,      keywords?,   depth?,   urgency)>
<!-- Source Information  -->
<!ELEMENT uri (#PCDATA)>
        <!ATTLIST uri group CDATA #IMPLIED
                              login CDATA #IMPLIED
                                password CDATA #IMPLIED
                              port CDATA #REQUIRED
                                timing CDATA #REQUIRED
                                lastcheck CDATA #REQUIRED>
<!ELEMENT type EMPTY>
        <!ATTLIST type protocol (http | pop3 | nntp | rmi) "http"
                                          method (push | pull) "pull"
                             processtype (filter | nonfilter) "filter">
<!-- Processing - Filtering Info -->
<!ELEMENT keywords (keyword+)>
            <!ELEMENT keyword EMPTY>
                        <!ATTLIST keyword key CDATA #REQUIRED   weight (1 | 2 | 3 | 4 | 5) "3">
          <!ELEMENT depth EMPTY>
                    <!ATTLIST depth level (1 | 2 | 3) "1">

<!-- Urgency -->
<!ELEMENT urgency EMPTY>
                        <!ATTLIST urgency urgent (1 | 2 | 3) "2">
```

eRACE Information Architecture (ctd')

- Annotation Cache Interface (ACI): maintains structural & semantic information about collected content

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT cache (annotation+)>
<!ATTLIST cache location CDATA #REQUIRED
                size CDATA #REQUIRED
                maxsize CDATA #FIXED "50000"
                locked (false | true) #IMPLIED
                  unique_id CDATA #REQUIRED>
<!ELEMENT annotation (uri,urgency,docbase,expired,summary)>
   <!ATTLIST annotation id ID #REQUIRED
                        owner CDATA #REQUIRED
                        extension CDATA #REQUIRED
                        format (text | html | binary | multipart | unknown )
                        relevance CDATA #REQUIRED
                        updatetime CDATA #REQUIRED
                        filesize CDATA #REQUIRED>

      <!ELEMENT uri (#PCDATA)>
         <!ELEMENT urgency EMPTY>
            <!ATTLIST urgency urgent (1 | 2 | 3)  #REQUIRED>
      <!ELEMENT docbase (#PCDATA)>
      <!ELEMENT expired EMPTY>
         <!ATTLIST expired expir (true | false) "false">
      <!ELEMENT summary (#PCDATA)>
```