Maintaining Security & Privacy w/in a Peer to Peer Network

Gregory Alan Bolcer

Endeavors Technology, Inc.

http://endeavors.com

magi

How is P2P Different?

Nothing inherently client-server in Web protocols; just most commonly deployed network architecture

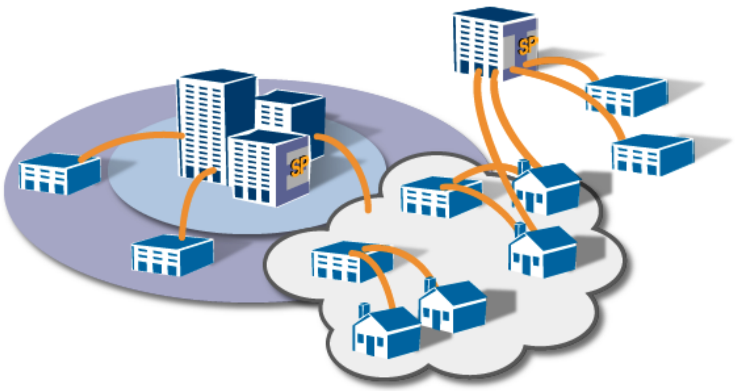
	Client-server	P2P
Network traffic	asymmetric, e.g., cable modem, ADSL	symmetric (threatens cable & ADSL)
Intellectual property	under the control of the server	under the control of each and every peer (threatens copyright)
Intranet control	firewalls protect servers, port 80 used by Web clients	firewalls restrict peer behavior, port 80 subverted
Addressing	primarily static DNS, Network Address Translation (NAT) for clients is transparent	uses dynamic real-time registries in place of DNS, NAT can be restrictive

Analyst Predictions for P2P

- IDC - 23.6% of large corporations will install an instant messaging system in the next year.
- Gartner - By 2002, >50% of global Internet users will regularly sign on to at least 2 P2P Internet applications
- Forrester - By 2002, 3 million households will use P2P applications to make their digital photos available to family and friends.
- Forrester - By 2004, 33% of the online population will use P2P services for storing and retrieving personal data.
- Forrester - By 2005, P2P services will come bundled in premium broadband fees and personal information-sharing applications from Adobe, Palm, and AOL.

Why Decentralize? It's Where the Data Is.

- 70% of enterprise data is not located in a centralized server or database,(Gartner/Bear-Stearns)
  - It's on the desktops, laptops, palmtops, PDAs, smartphones, etc.
  - Need to centrally scale the business logic with access to information "in place"
- Decentralizing IT Administration is difficult
  - Users don't have the skills to secure their own data
  - Preventing access is extremely difficult
  - Revocations difficult to update



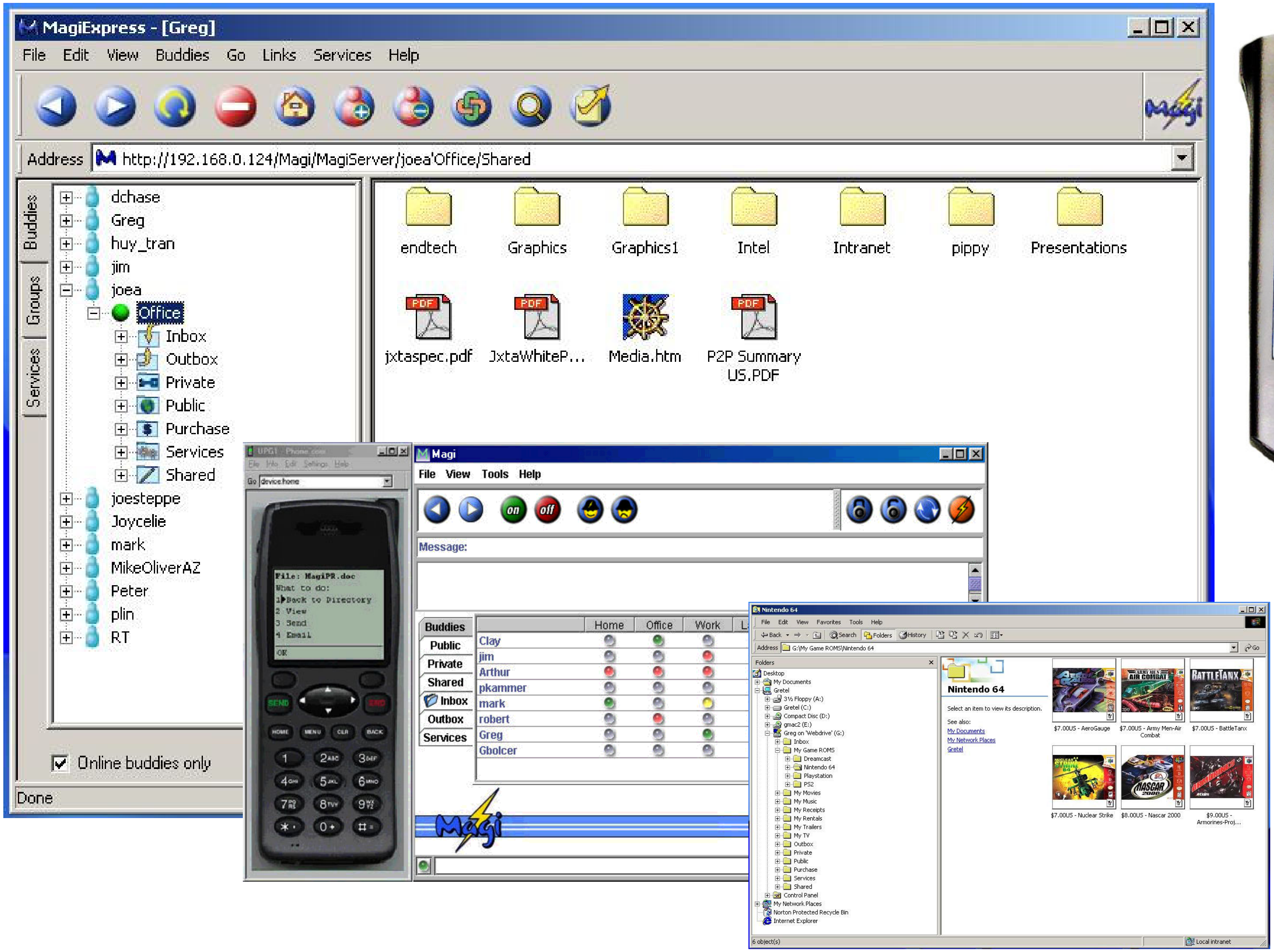
- It is an expensive operation to centralize data,
  - It's constantly changing
  - Centralizing metadata is a much cheaper operation
  - ERP & Large Database systems have discovered this
- Tracking, Status, Audit, Search is difficult
  - Human nature, I want to copy it and do it myself
  - Human work not easily segmentable, overlap of work leads to social and political problems

Why Magi P2P?

- Searching is as crucial to P2P as it is to the Web
- Scale Web Protocols to billions instead of millions
  - Nothing inherently client-server in Web protocols
  - Just the most commonly deployed architecture
  - Internet-Scale architecture versus Enterprise-Scale architecture
- Not pure P2P, but can be
  - Thin-server on every device to speak HTTP and WebDAV
  - Naming, security, registration, tracking can all be centralized
- Smart Proxying and Value added Web Svcs.
- Similar to Freenet, Gnutella, Napster but doesn’t reinvent the Web;
  - Apache or Tomcat HTTP server & plugins & other p2p protocols
  - Extensible Java/Python/C interoperable protocol implementations
  - XML-based access controls using user controlled “Buddy lists”
  - Dynamic authentication controls; IT friendly, parseable vocabulary
  - Public & Private Key certificates & OpenSSL
  - WML and X.10 modules

magi

Open GUI w/ Multiple Pathways to Data



MagiExpress - [Greg]

Address: http://192.168.0.124/Magi/MagiServer/joea'Office/Shared

Go: device.home

```
File: MagiPR.doc
What to do:
1 Back to Directory
2 View
3 Send
4 Email
OK
```

Address: G:\My Game ROMS\Nintendo 64

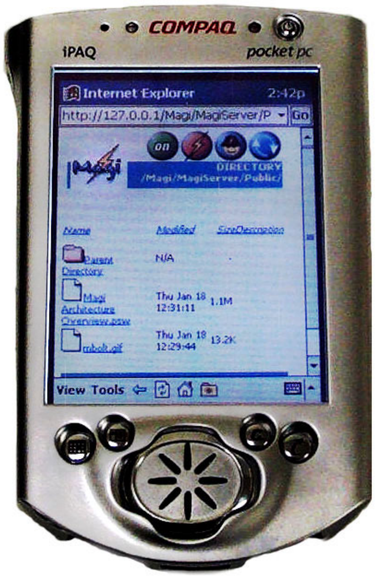
Nintendo 64

Select an item to view its description.

See also:
My Documents
My Network Places
Gretel

$7.00US - AeroGauge
$7.00US - Army Men-Air Combat
$7.00US - BattleTanx
$7.00US - Nuclear Strike
$8.00US - Nascar 2000
$9.00US - Armorines-Proj...

6 object(s)



Dozens of Commercial Tools that are WebDAV compliant

magi

Magi is Standards “Smart”

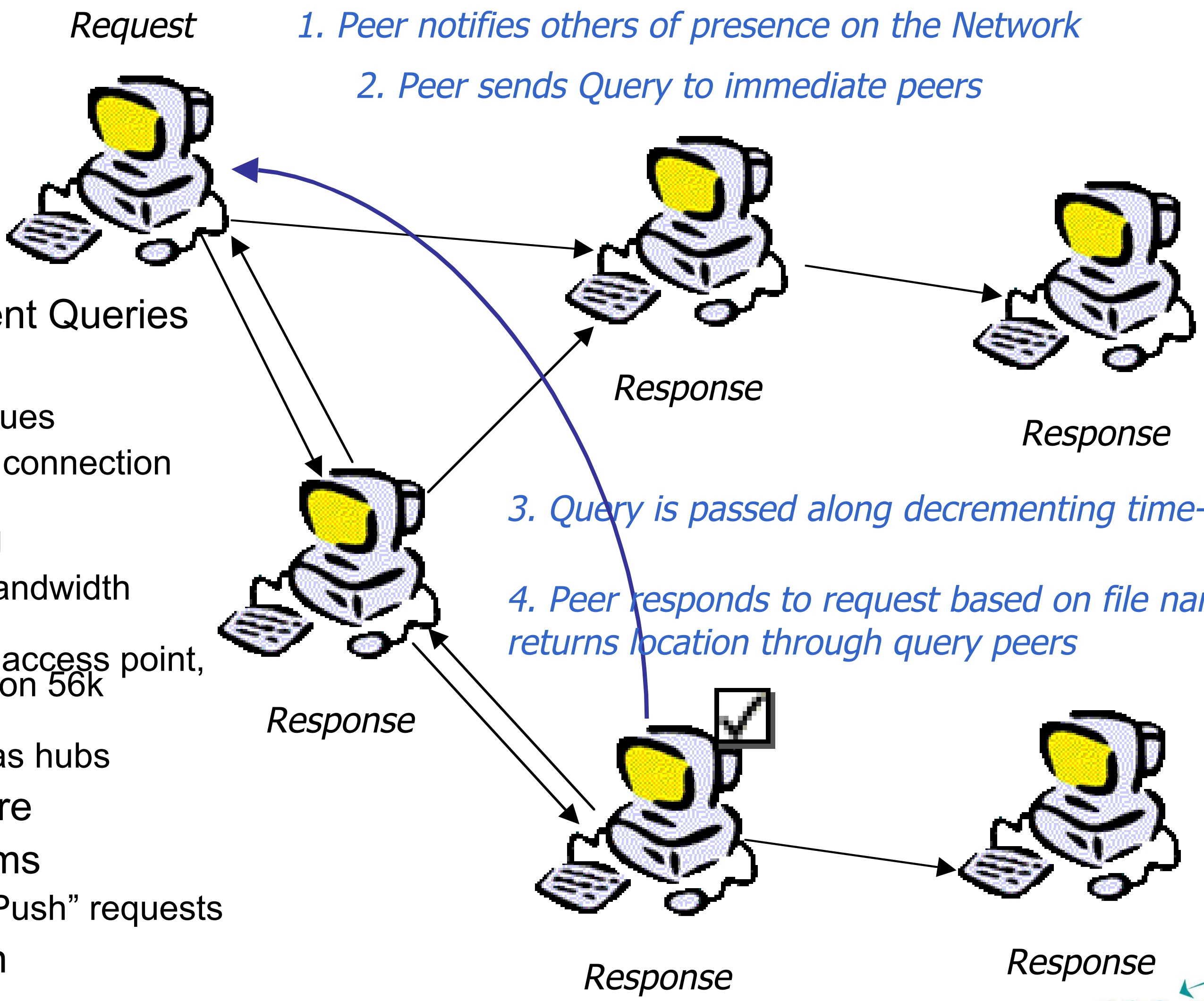
- Deep involvement in standards groups and efforts
- HTTP RFC 1945, 2068, 2616
- WebDAV RFC 2291, 2518
- XML, Java, Python
- OpenSSL, RSA keys, X509 certificates, X509 CRLs
- Universal Resource Identifiers RFC 1630, 2396
  - Locators RFC 1736,1738,1808
  - Names RFC 1737,2141

P2P Searching

- Gnutella
- Napster
- Magi/Web
- Filtered Search
- Network Architectures
- Unique Features

Gnutella/Infrasearch Queries

- Dynamic Content Queries
- Ping Flooding
  - Liveness issues
  - Guaranteed connection issue
- Query Flooding
  - Increased bandwidth usage
  - From dialup access point, 64k queries on 56k connection
  - Slow hosts as hubs
- Bugs in Software
- Scaling Problems
  - Broadcast “Push” requests
- Tool Integration
- Freeloading



1. Peer notifies others of presence on the Network

2. Peer sends Query to immediate peers

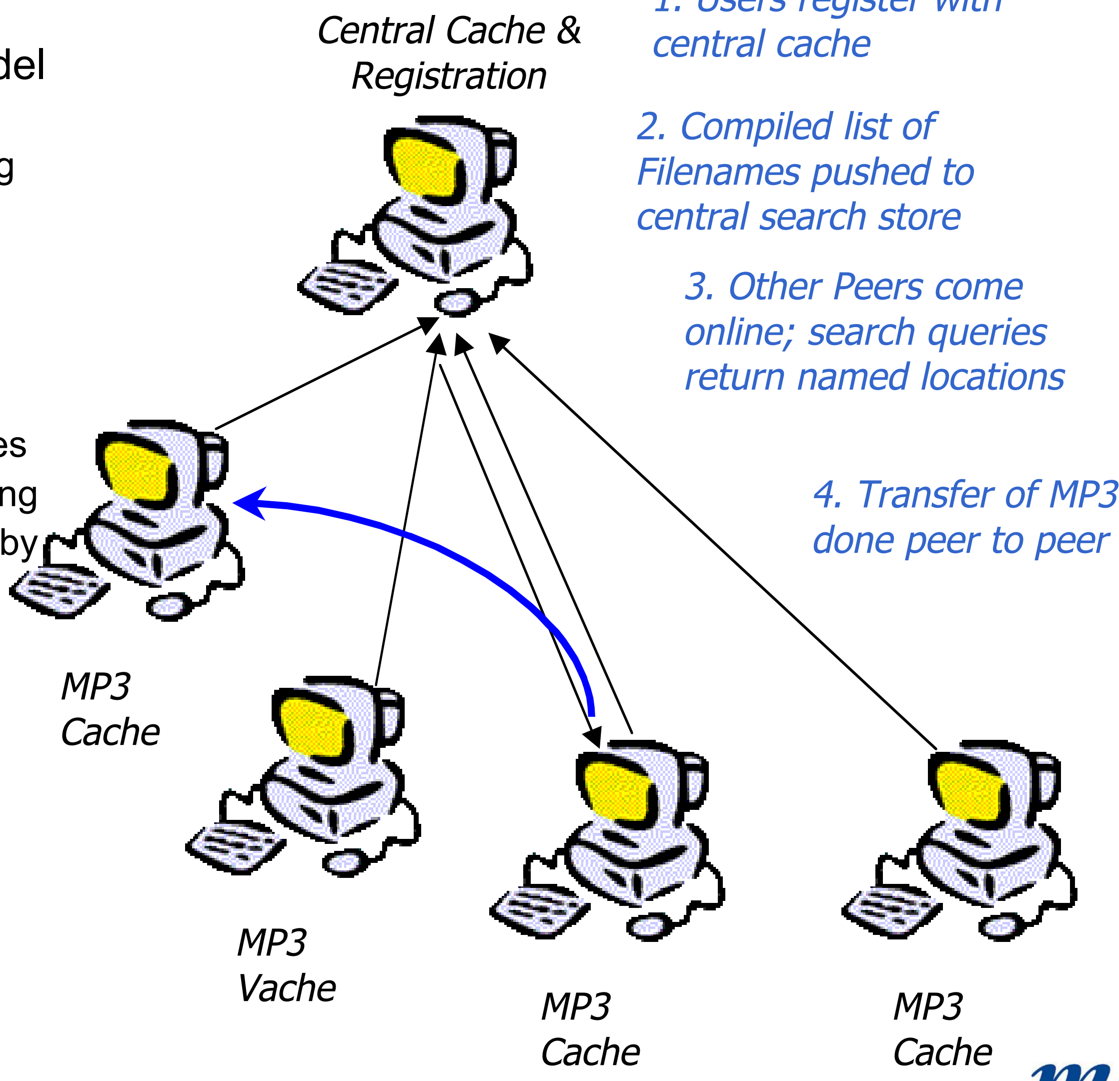
3. Query is passed along decrementing time-to-live

4. Peer responds to request based on file naming; returns location through query peers

5. File is returned via direct call to Responder

Napster Searching

- Metadata “Push” model
  - File names & sizes
  - No content indexing
  - No keyword searching
  - Channel metadata
- User namespace
  - Identification only
  - No protected spaces
  - Dynamic IP matching
  - Collisions handled by demaind
- Centralized
  - Registration
  - Searching
  - File transfer done peer to peer
- Search Space
  - Segmented by registration server
  - No cross server queries



1. Users register with central cache

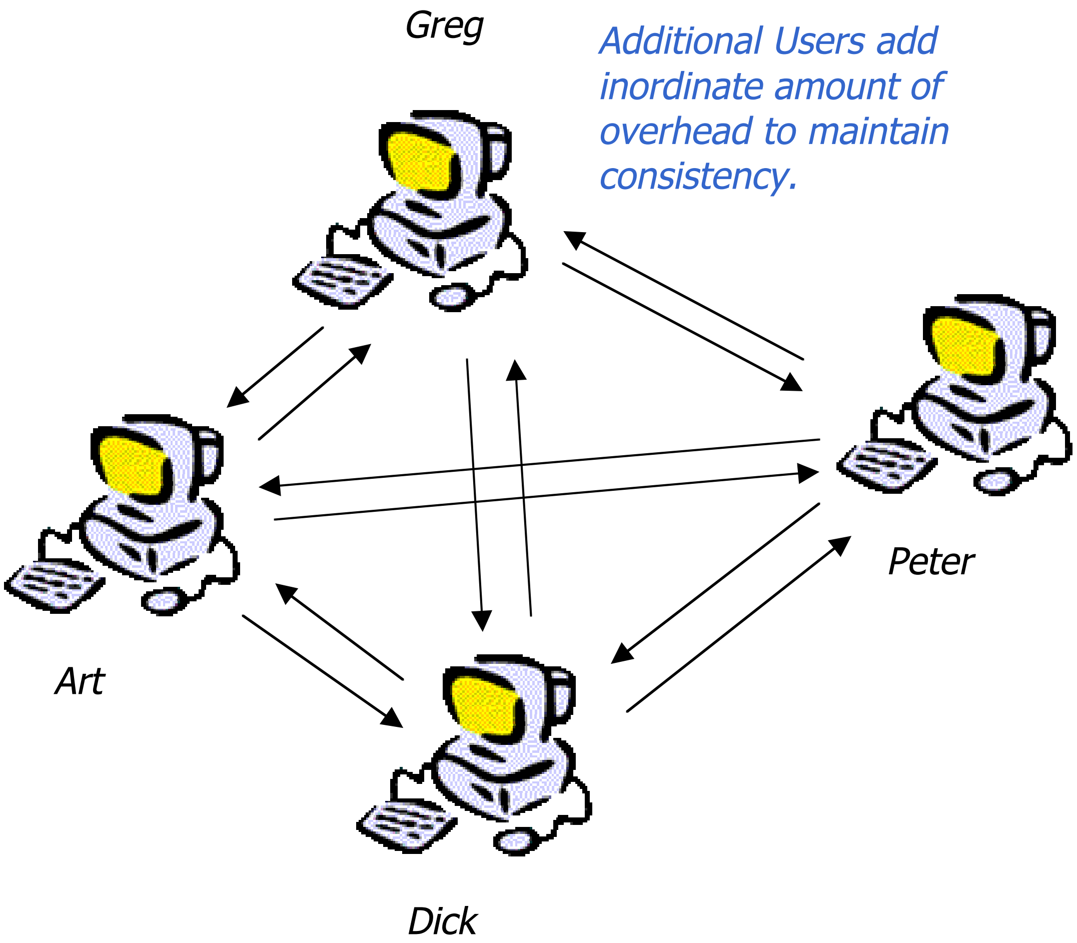
2. Compiled list of Filenames pushed to central search store

3. Other Peers come online; search queries return named locations

4. Transfer of MP3 files is done peer to peer

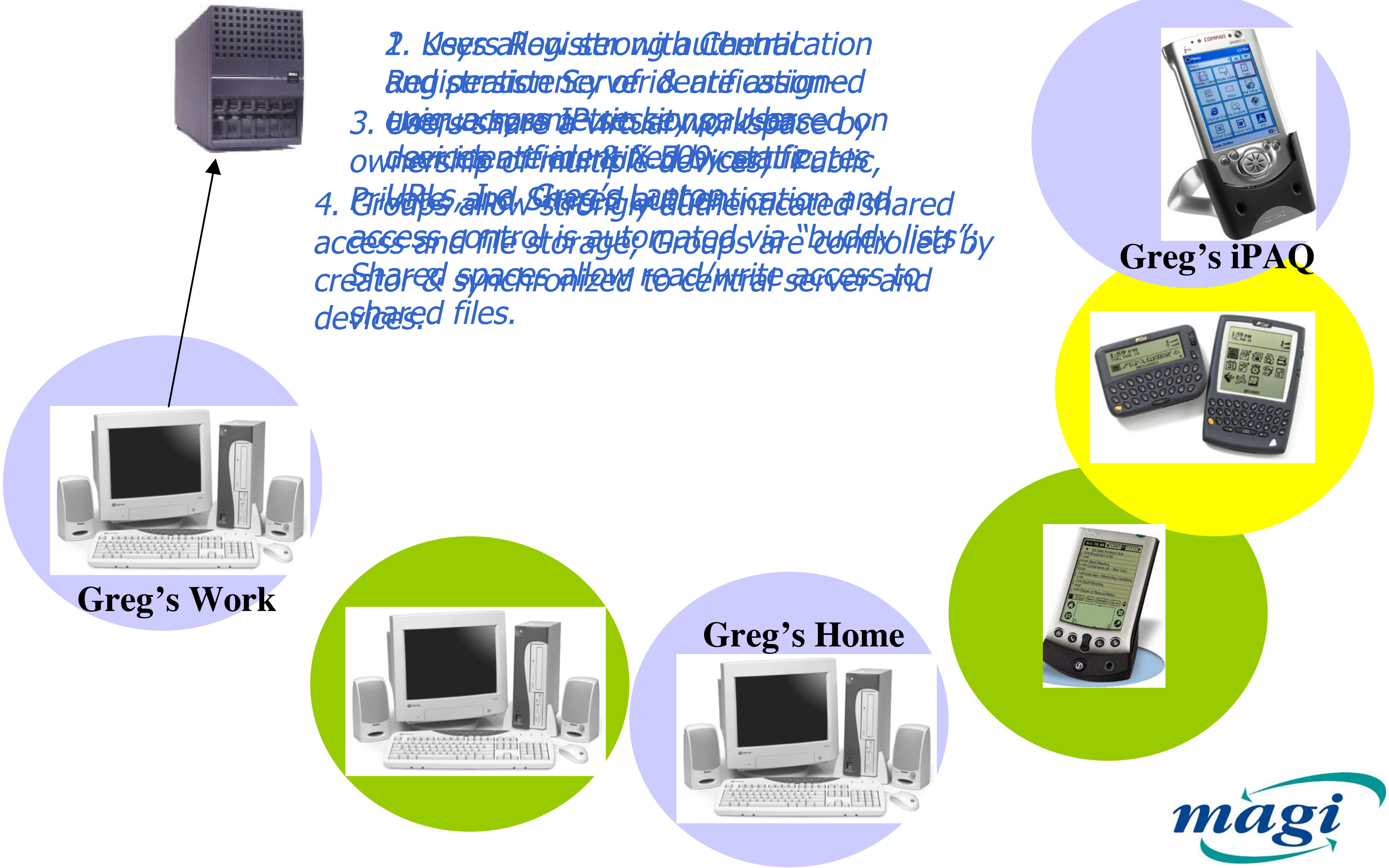
Groove Searching

- Users invite individuals into a group workspace
- Initial user authentication is done out of channel
- All data must be kept consistent across all participants
- Removing users leaves user with full permissions to copy of workspace data
- Adding users to workspace or large amount of content to workspace requires
  - High bandwidth requirements
  - High upfront synchronization costs
- Social changes to work habits required to take advantage of collaboration
- Searching is done on assumption that local workspace is consistent



Additional Users add inordinate amount of overhead to maintain consistency.

Searching with Magi



1. Users Register with Central Registration Server & are assigned unique, permanent IP address based on device identifier & X.509 certificates. URLs, I.e. Greg's Laptop

2. Keys allow strong authentication and persistency of identification – user access permissions, sessions [illegible]

3. Users share a virtual workspace by ownership of multiple devices; Public, Private and Shared authentication and access control is automated via "buddy lists". Shared spaces allow read/write access to shared files.

4. Groups allow strongly authenticated shared access and file storage; Groups are controlled by creator & synchronized to central server and devices.

magi

Searching with Magi

- *Indexing is done on capable peers;*
- *Small footrprint or limited peers allow proxy indexing*
- *Metadata is pushed to central search cache*
- *Search results are up-to-the-minute*
- *Search results return static "named" URL, not URL or IP where it was indexed*
- *All file transfers are done peer to peer using standard Web protocols*
- *Documents can be edited "in place" across the Web using Standard Adobe, Microsoft, other tools*

Magi Supports Group Filtering

- Groups can be managed centrally or locally
  - Endeavors@Greg'Work
  - Shared folder automatically created
  - Shared views automatically created
- Search results are filtered according to exhaustive access control
- Search metadata can be stored centrally for efficiency or locally for confidentiality and privacy.

Magi Unique Search Features

- Authentication, authorization, security
  - Mobile computing concerns
  - Strong authentication using X.509 symmetric keys
  - Access.xml access control & Web paths
  - Crawling over SSL
- Automated search space partitioning
  - Dynamic enrollment
  - Up to the minute search filtering
  - No lag between crawl and results
  - Push to Altavista central cache

Magi Unique Search Features

- Static naming; Dynamic IP addressing
  - Static naming across IP sessions
  - User friendly namespace, I.e. Greg’s Laptop
  - Index on one session, results point to live session
- Document Metadata
  - WebDAV Properties
  - Microsoft Office metadata
  - Web caching & metadata searches

Magi Unique Search Features

- Heterogenous document types
  - Full support for hundreds of file types using Altavista
- Unique Device Characteristics
  - Device.xml for device-friendly crawling
  - Sensitive to bandwidth constraints
- Resource and Web Service Proxying
  - Proxy services, indexing, crawling to more capable peers
- Intermittent access to the network
  - Can interrupt and continue crawling & caching
- Resource Caching
  - Browsing of offline directory structures
  - Access to last known copy via caching

P2P Collaboration

- Media Sharing
- Standards “Smart”
- Ad Hoc Collaboration/Collaborative Authoring
- Writable/Two-Way Web
- Plugin Architecture
- Smart Network Services
- Caching
- Workflow

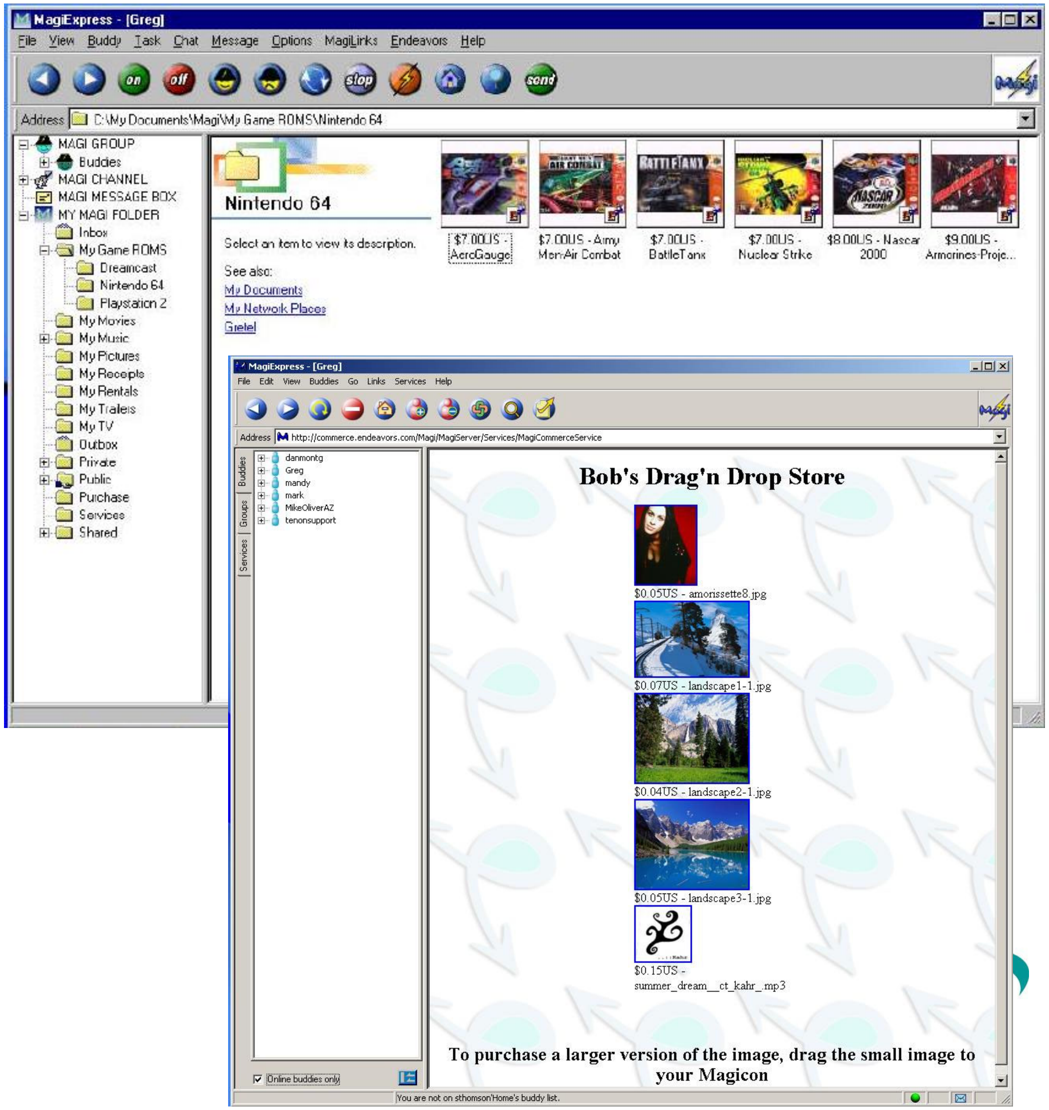
magi

Media Sharing

- Public – Read only, public viewable with browser or DAV client
- Private – Read/Write for owner of namespace only
- Shared – Read/Write for any Buddy
  - Shared/Group provides automated group permissions for sub-resources
- Dynamic Search Model
  - Configurable crawler pushes metadata
  - Filename, metadata tags, and indexed searches

Media Sharing - Photos

- Thumbnails may be used as metadata
- Metadata can be centralized for efficient searching or decentralized for ease of use
  - Thumbnails
  - Img tags
  - Small photo
  - Large photo
- Photo owners may want to retain control by keeping large photo or thumbnails on own machine
- May centralize thumbnails or small photos to provide offline searching capabilities



Select an item to view its description.

Bob's Drag'n Drop Store

To purchase a larger version of the image, drag the small image to your Magicon

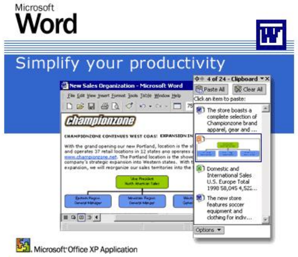
You are not on sthomson'Home's buddy list.

Ad Hoc Collaboration

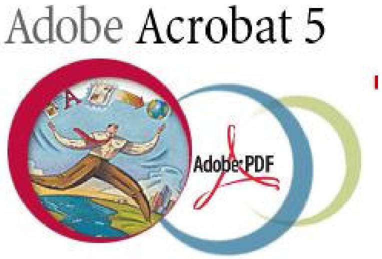
- SSL between any two points in the network
- Web File System
  - Double click to Open,
  - Cut/Copy/Paste,
  - Drag-and-Drop to Web, Save to Web,
  - File locking
- Collaboration across peers:
  - WebDAV file locking
  - CoBrowsing
  - NetMeeting link & launch
  - Other collaborative browser & server plugins (VNC, Citrix, Placeware, Exceed, etc.)
- User has own namespace, I.e. “Greg”; Greg’s Laptop, Greg’s Home Computer, etc.
- Public, Private, and Shared folders
- Groups require invitation
- Symmetric trust model for Read/Write

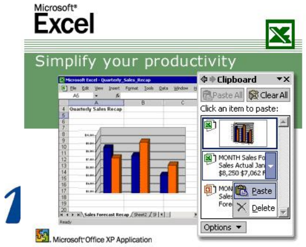
Web Authoring

- Evolving WebDAV IETF working groups & standards
- WebDAV, DASL, DeltaV, DAV ACL
- Resource locking, overwrite prevention, metadata mgt.
- Integration with any WebDAV compliant client tool
- Magi Apache 1.3.x/2.0 or Tomcat/Jakarta architecture

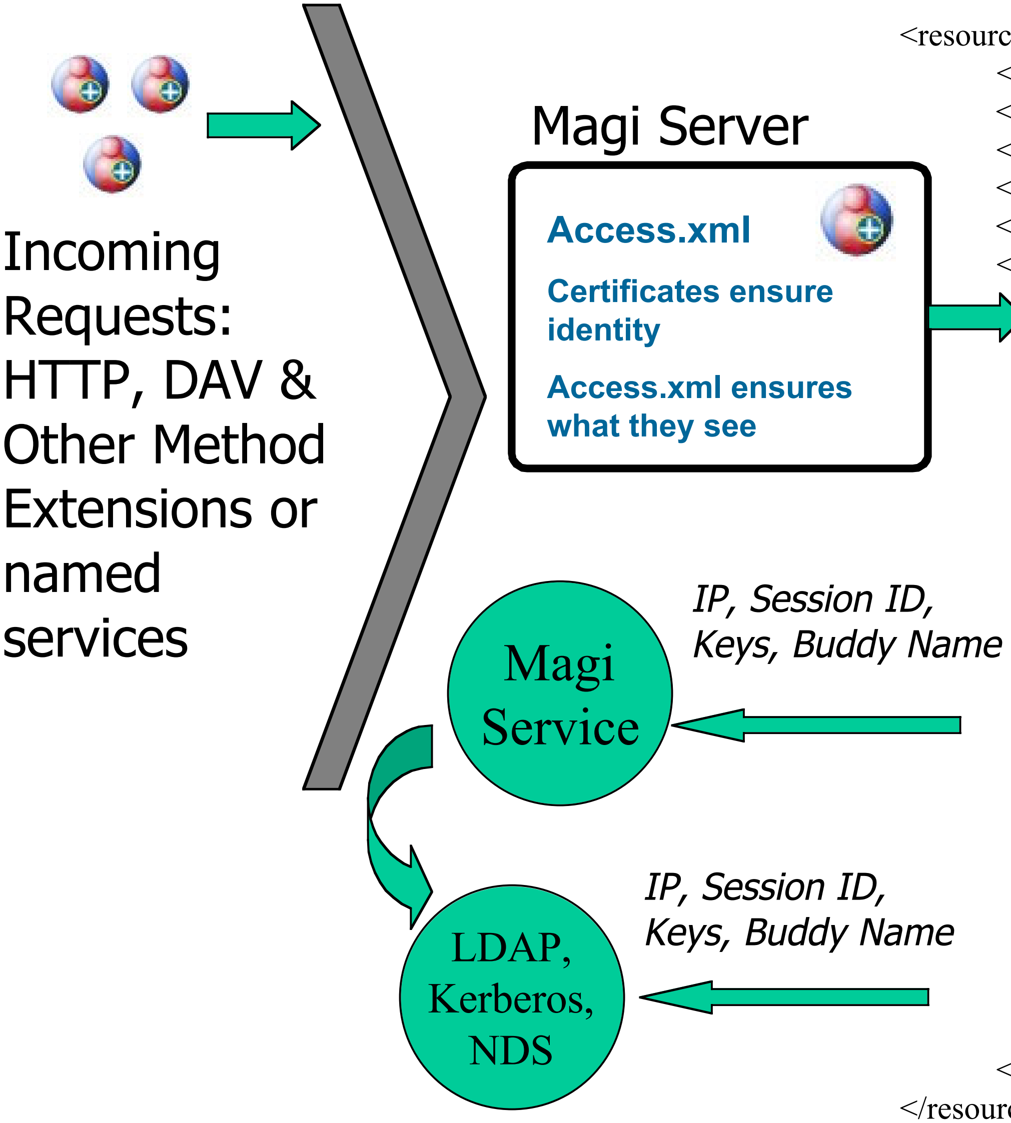


Microsoft PowerPoint
Create captivating presentations
Microsoft Office XP Application





WebDAV Access.XML



Incoming Requests: HTTP, DAV & Other Method Extensions or named services

Magi Server

Access.xml

Certificates ensure identity

Access.xml ensures what they see

Magi Service
IP, Session ID, Keys, Buddy Name
LDAP, Kerberos, NDS
IP, Session ID, Keys, Buddy Name

```
<resource type="directory">
        <pathname>C:\My Documents\Magi/Shared</pathname>
        <name>Shared</name>
        <creation-date/>
        <modified-date/>
        <size>n/a</size>
        <authentication method="basic">
          <dav-auth allow-overwrite="true">
             <allowed-method>DELETE</allowed-method>
             <allowed-method>MKCOL</allowed-method>
             <allowed-method>PROPFIND</allowed-method>
             <allowed-method>PROPPATCH</allowed-method>
             <allowed-method>COPY</allowed-method>
             <allowed-method>LOCK</allowed-method>
             <allowed-method>UNLOCK</allowed-method>
             <allowed-method>GET</allowed-method>
             <options>followsymlink</options>
             <options>multiviews</options>
          </dav-auth>
          <buddy-file>file:///Magi/buddy.xml</buddy-file>
          <soap-auth>
             <allowed-method>SoapAction:Copy</allowed-method>
          </soap-auth>
          <user>
             <username>Greg</username>
             <password>YtWp4g9nMw</password>
          </user>
        </authentication>
</resource>
```

Pluggable Architecture

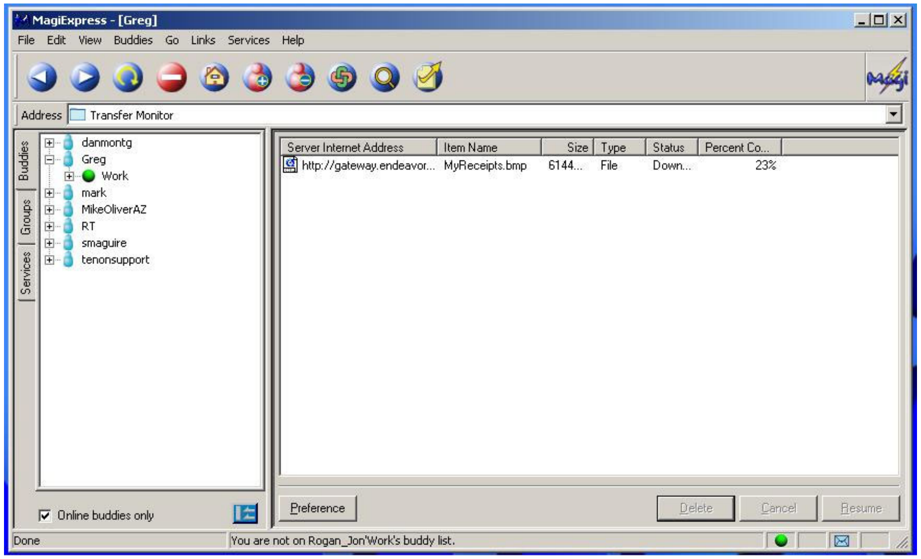
- Every peer is both a client and a server
- Client side based on IE 5.5 on Win platforms; Mozilla/Netscape 6.x engines and tools for other platforms
  - Supports standard browser plugins
- Server side based on Apache Module interface & CGI-based scripting languages and packages
  - Large number of packages and modules available through commercial, shareware, open source
- GUI independent of Peer
  - HTTP/XML interface to Peer

Smart Network Services

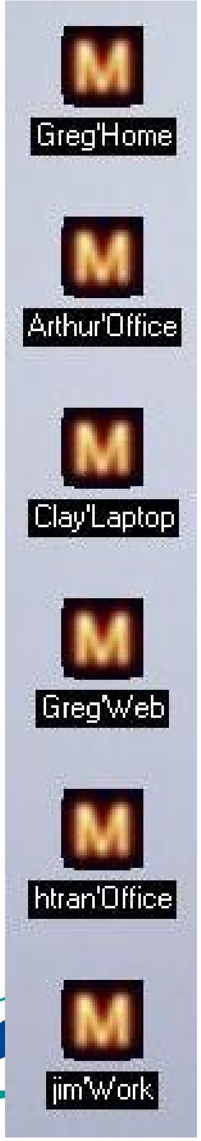
- Any Magi peer can serve as a store & forward service
- Can be used when two peers unlikely to be online at the same time
- Also used for overcoming firewalls that don’t allowing incoming HTTP traffic
- Event Store & Forward
  - Property set in config file
  - Works for instant messaging, notification, pending file
  - Application and End User events
- File Store & Forward
  - General purpose subject to EULA & copyright restrictions, a.k.a. “touching the file”
  - Event service combined with S&F cache
    - “Push” file to S&F cache
    - Notify peer that there is a file pending
    - Peer “Pulls” file from S&F cache
  - SSL between cache and peers; restricted pickup access

Efficient Web Doc Management

- Both GET and PUT
- Magi Web Folder view allows Right-Click and “Download” monitoring or drag to buddy icon
- Support for one-time tickets & multi-issue
- Compression using mod_gzip & others
- Xfers are done using HTTP and DAV
  - – Support for byte range “GET” using HTTP
  - – DAV “PUT”
  - – Incremental downloads
  - – HTTPR & SRMP for reliable & resume



Server Internet Address	Item Name	Size	Type	Status	Percent Co...
http://gateway.endeavor...	MyReceipts.bmp	6144...	File	Down...	23%

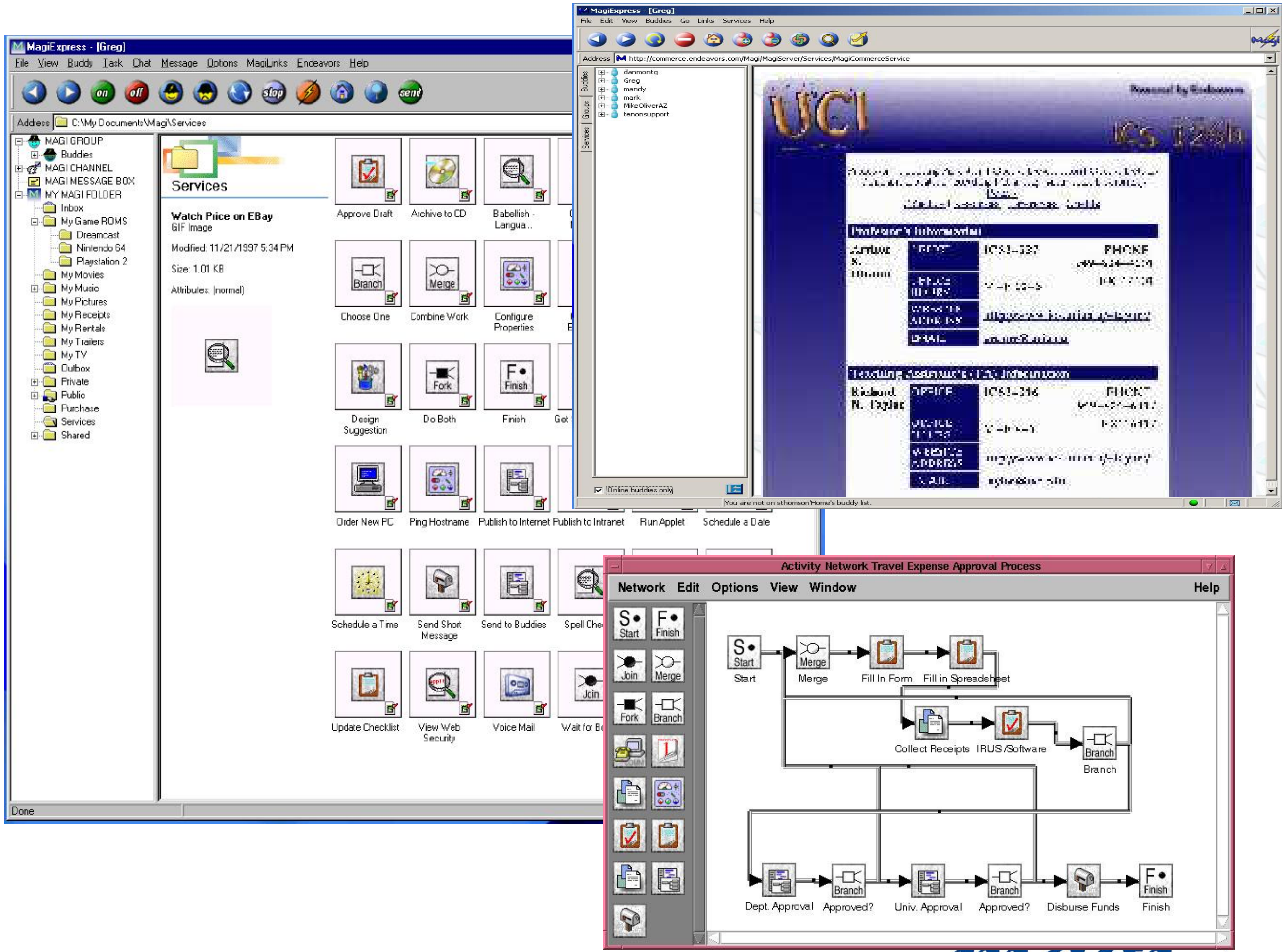


Caching

- Any Magi peer can serve as a cache
- Caching is done at 3 levels:
  - Search cache supports file download & comparison with offline peers
  - Peer-side Web caching controlled through Web browser integration
  - Enterprise Web caching through traditional Web caching models
- Web caching model supports
  - Resources reference by URLs
  - Domain is the authority on resolution
  - Allows resource naming by reference, comparison using HTTP HEAD method, conditional GET, and metadata
  - Avoids resource spoofing of other p2p file systems

Wide Area Web Services

- Workflow components work in concert:
  - Process.xml
  - Shared Work Across Peers WebDAV derived protocol
  - Endeavors Java workflow engine
- Services are network loadable servlet plugins
- Individual Magi peers can advertise services
- Template and JSP to provide end user views
- eProcesses can be built across peers using network editor



P2P Security

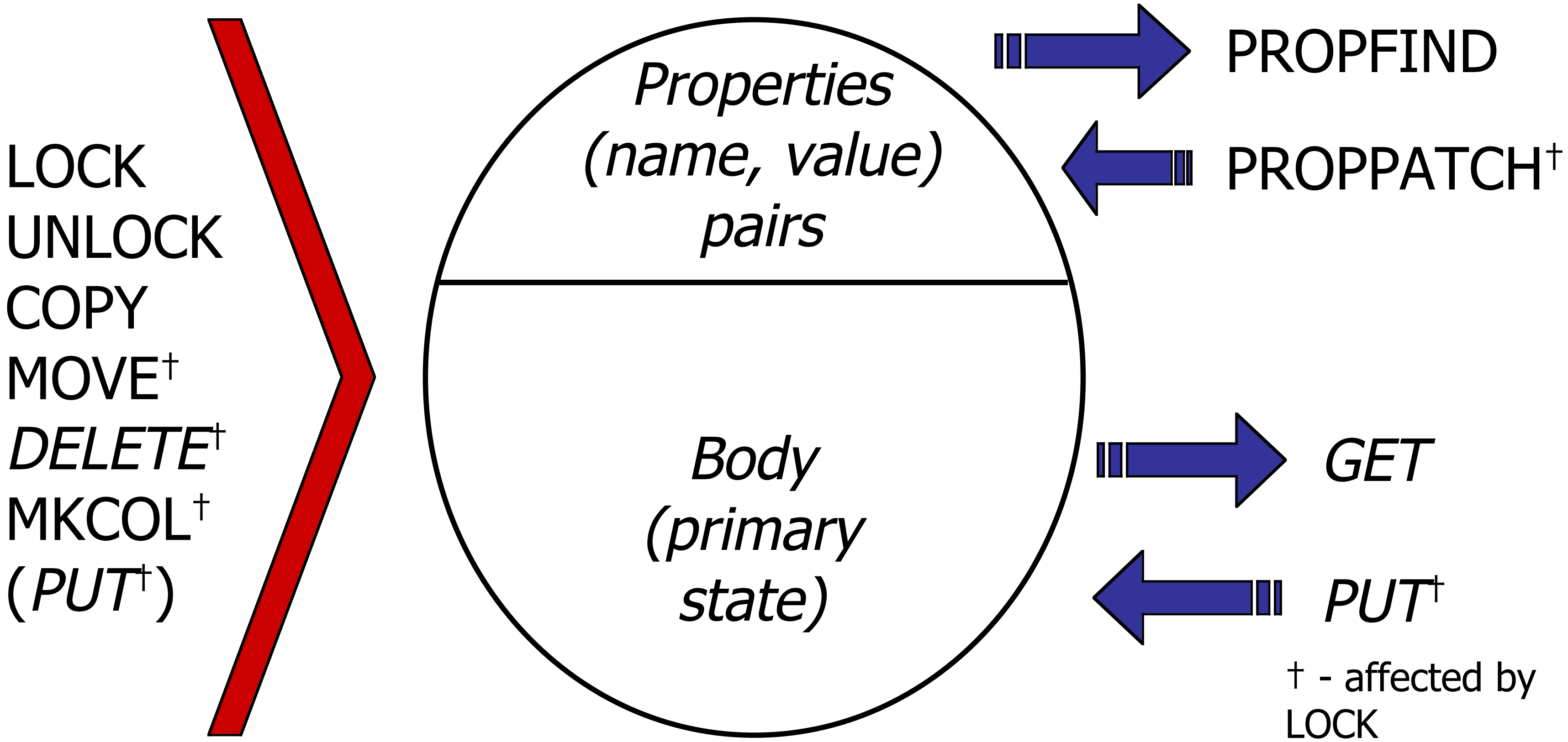
- Media Sharing
- Standards “Smart”
- Ad Hoc Collaboration/Collaborative Authoring
- Writable/Two-Way Web
- Plugin Architecture
- Smart Network Services
- Caching
- Workflow

Overview of Security Concerns

- Authentication/Authorization
  - Who are you?
  - What do you get to look at?
- Integrity
  - Has the message been tampered with?
- Confidentiality
  - Is the message hidden from others?
- Auditing/Logs
  - Who's been here?

Generic Interface & PropertDAVw/

Web Resource



LOCK
UNLOCK
COPY
MOVE†
DELETE†
MKCOL†
(*PUT*†)

Properties (name, value) pairs

Body (primary state)

PROPFIND

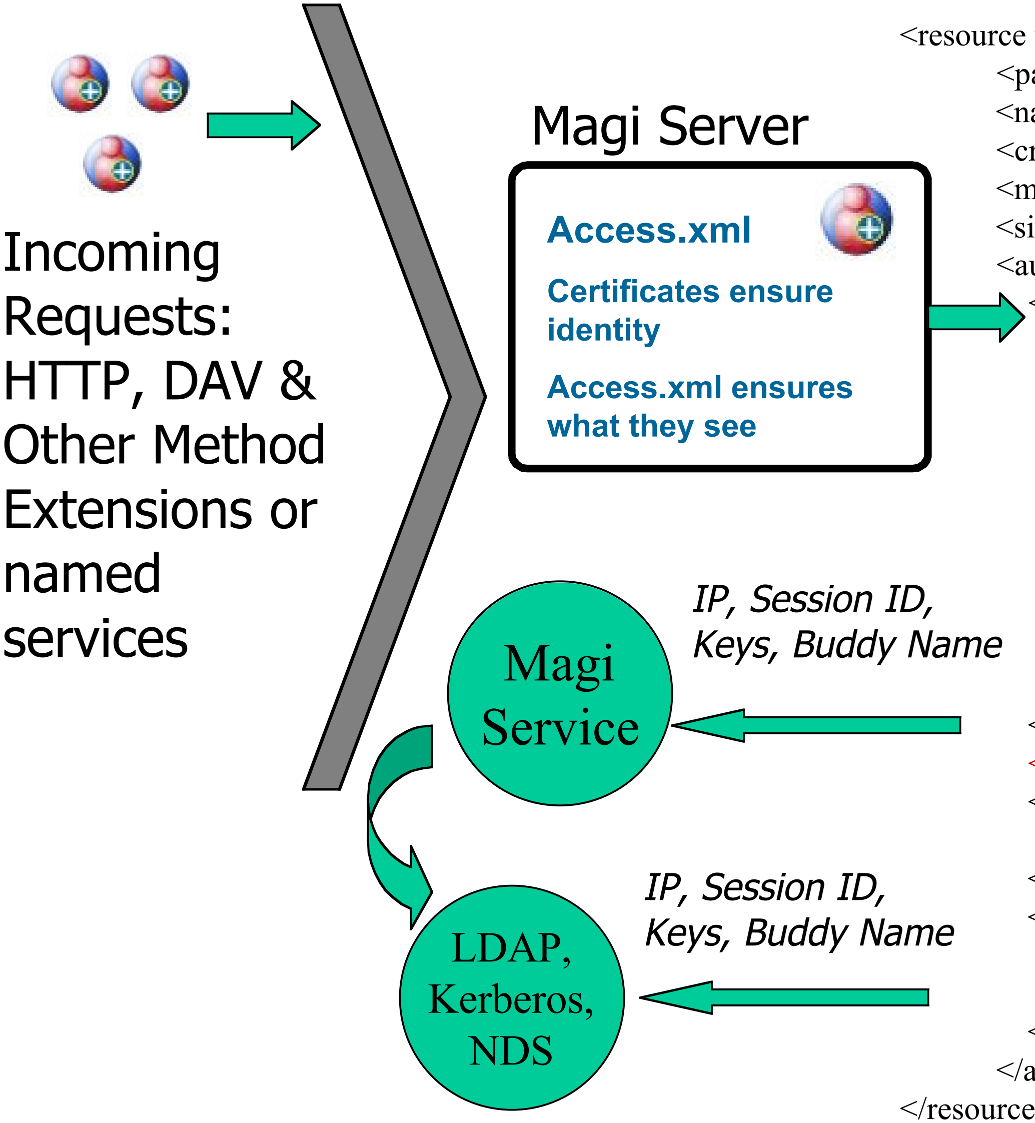
PROPPATCH†

GET

PUT†

† - affected by LOCK

Access.xml & Filtering



Incoming Requests: HTTP, DAV & Other Method Extensions or named services

Magi Server

Access.xml

Certificates ensure identity

Access.xml ensures what they see

Magi Service

IP, Session ID, Keys, Buddy Name

LDAP, Kerberos, NDS

IP, Session ID, Keys, Buddy Name

```
<resource type="directory">
        <pathname>C:\My Documents\Magi/Shared</pathname>
        <name>Shared</name>
        <creation-date/>
        <modified-date/>
        <size>n/a</size>
        <authentication method="basic">
          <dav-auth allow-overwrite="true">
            <allowed-method>DELETE</allowed-method>
            <allowed-method>MKCOL</allowed-method>
            <allowed-method>PROPFIND</allowed-method>
            <allowed-method>PROPPATCH</allowed-method>
            <allowed-method>COPY</allowed-method>
            <allowed-method>LOCK</allowed-method>
            <allowed-method>UNLOCK</allowed-method>
            <allowed-method>GET</allowed-method>
            <options>followsymlink</options>
            <options>multiviews</options>
          </dav-auth>
          <buddy-file>file:///Magi/buddy.xml</buddy-file>
          <soap-auth>
             <allowed-method>SoapAction:Copy</allowed-method>
          </soap-auth>
          <user>
             <username>Greg</username>
             <password>YtWp4g9nMw</password>
          </user>
        </authentication>
</resource>
```

Magi Security Machinery

- Magi Certificate Authority
  - The authority on who’s who in Magi space.
    - Issues certificates.
    - Issues CRLs.
    - Keeps a database of all certificates and all revoked certificates.
- Magi Public Key Infrastructure
  - RSA keys, X509 certificates, X509 CRLs
  - Magi certificate authority server
  - Run time configuration
- RSA keys, X509 certificates, X509 CRLs
  - Magi generates its own key pair.
    - RSA key pair 1024 bit.
    - Private key is stored in Triple DES encoded PEM file.
  - Magi registers the public key with the Magi certificate authority.
    - Magi CA establishes name space for this Magi.
  - Magi uses custom X509 CRLs.
    - At regular intervals Magi queries the Magi CA for a CRL.

Communication Machinery

- SSL
  - Accepted Standard
    - Choice of cipher suites
    - Timestamps, nonce values, hashes, signatures…
  - Limitations
    - Point to point
    - Store and forward
    - Chat and Instant messaging
- SSL alternatives
  - Signed Events
    - Secure Authentication
    - Tamperproof
  - Shared Symmetric Keys
    - Content based encryption

Magi Security Machinery

- HTTP Event Service using SSL – It’s really that simple!
  - public static HttpEvent sendRequest(String host, HttpEvent evt, int ssl)
  - SSL limits itself to contracts with known entities with fixed IP
  - Store and Forward or Chat & IM break model
  - Really need signed Content-based encryption and signed events
- Content-based Encryption & Signed Events
  - Authentication/Authorization – Who are you? What do you get to see?
  - Integrity – Has the message been tampered with? During transport?
  - Confidentiality – Is the message hidden from others?
  - Auditing – Who’s been here? What did they want?

```
<EVENT>
      <EVENT_TYPE>Yes</EVENT_TYPE>
      <EVENT_BEHAVIOR>NOTIFICATION_EVENT</EVENT_BEHAVIOR>
      <EVENT_VERSION>1.0</EVENT_VERSION>
      <IP>192.168.0.108</IP>
      <TIMESTAMP>985155300289</TIMESTAMP>
      <EVENT_ID>192.168.0.108:985155300289:20131:-1029658595</EVENT_ID>
      <EVENT_COUNT>20131</EVENT_COUNT>
      <PRIORITY>0</PRIORITY>
      <SOURCE>jim'tomcat1</SOURCE>
      <PARAMETERS><Username>jim'tomcat1</Username></PARAMETERS>
      <SIGNATURE>8F754FBCD2833A95746345D10350BC233FA95E520523C93348BCB656A18F17F8
A1B41BB64C12B2E79E71F8648CF9ECDD7BB8DB1E7086C2F4F46F3150D80C8A53E9A5EDE63BF053F276F
772F7F7BB4D7D5135E2D6FECBEF1E3BDD314D88722B2B4284BDD43DAD83F286413305D670C04D9C0177
98A3A2F9940B80CDC44698B8A9</SIGNATURE>
      <X509>E3D45B14F2551F4754F367D3C7DA47525A06F8AD</X509>
</EVENT>
```

Communication Machinery

- Audit/Logging
  - Corporate environments demand accountability
  - Permanent records must be maintained regarding who or what accessed or modified critical data, services, or systems configuration
  - System infrastructure must maintain its own log as well as provide facilities for applications to log events
- Intrusion Detection
  - Real-time event monitoring and analysis to detect abuse

CryptoManager

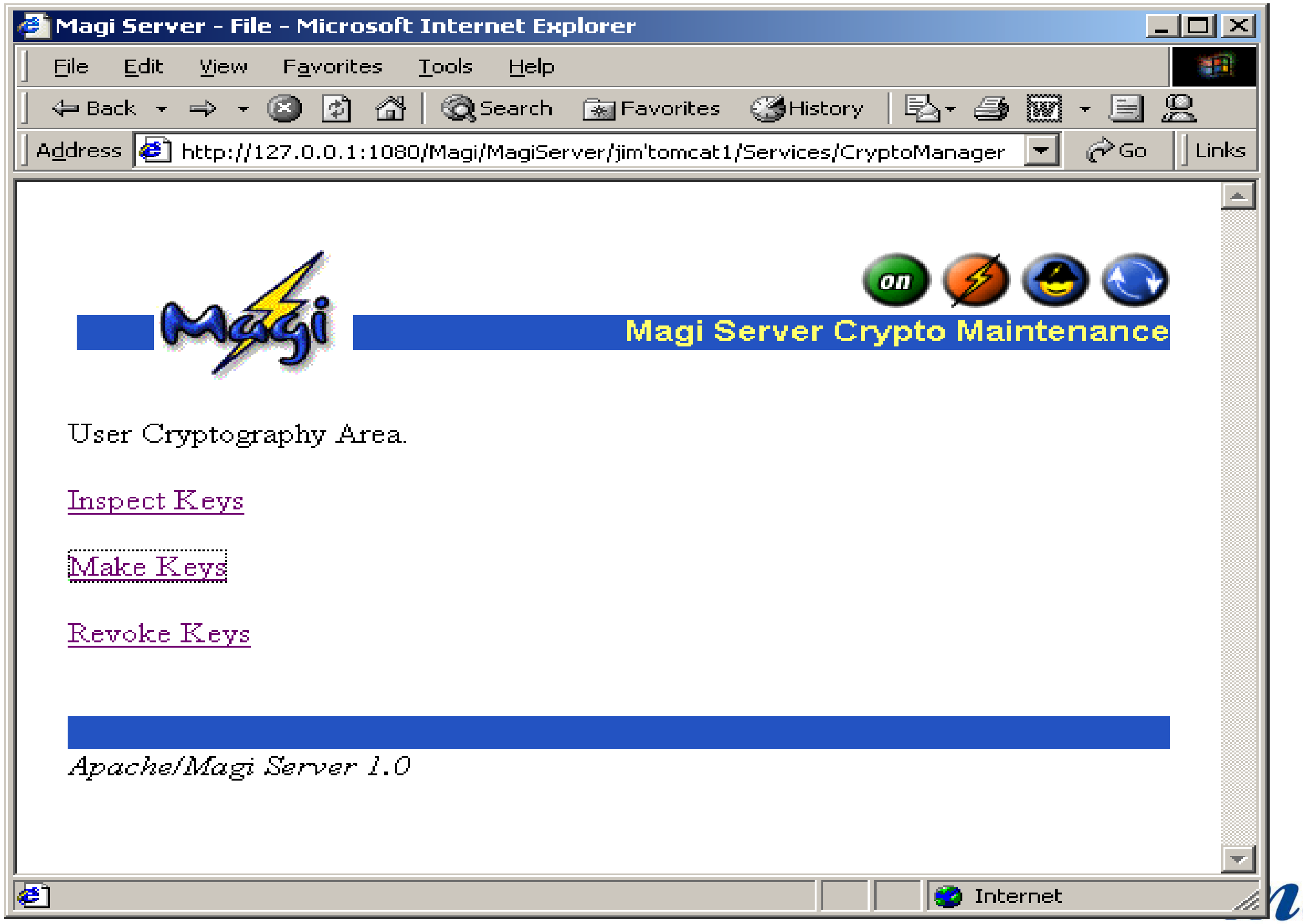
- Manages cryptographic functionality
  - Manages all key material
  - Performs all cryptographic manipulations
- Provides services to other parts of Magi
  - CryptoManager presents itself as a service

CryptoManager Services

```
public static HttpEvent sendRequest(String host, HttpEvent evt, int ssl)

public HttpEvent(MagiContext context, String type, String behavior, String parameters, HttpEvent responseTo,
                    HttpEvent[] eventList, boolean signed)

public int getSignatureStatus() { return signatureStatus; }

public static final int SIGNED_AND_VALID                  = 1;
public static final int SIGNED_AND_NOT_VALID              = 2;
public static final int NOT_SIGNED                        = 3;
public static final int SIGNED_MISSING_CERTIFICATE        = 4;
public static final int UNKNOWN                           = 5;
```

```
//Inside an event instance requesting CryptoManager for a signature
MagiLog.log("=====>>>> Querying for CryptoManager Service.");
CryptoManagerInterface cryptoManagerInstance = (CryptoManagerInterface)
MagiServiceManager.queryService( "org.endeavors.magi.services.secure.CryptoManagerInterface");
//various code to make sure cryptoMangerInstance is not null.
String[] retVals = cryptoManagerInstance.signEvent(this.context, this.toXML());
if ( retVals != null )
{
   this.shaRSASignature = retVals[0];
   this.x509Cert = retVals[1];
   MagiLog.log("=====>>>> Signing was successfull.");
}
```

CryptoManager Interface



Magi Server - File - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://127.0.0.1:1080/Magi/MagiServer/jim'tomcat1/Services/CryptoManager
Go Links
Magi
on
Internet

Magi Server Crypto Maintenance

User Cryptography Area.

Inspect Keys

Make Keys

Revoke Keys

Apache/Magi Server 1.0

Examples

- What would you use?
  - File transfers
  - HTTP Events
  - Chat
  - Instant Messages