# YouServ: A Web Hosting and Content Sharing Tool for the Masses

Roberto Bayardo

IBM Almaden Research Center

Joint work with Rakesh Agrawal, Daniel Gruhl, and Amit Somani

# Goal

- Allow people to *easily* share *as much* stuff on *the web* as they please with *little to no cost*.

# Solution

- Provide a system and software (called YouServ) that users run to serve content on the web with their own machines.
- Not just a webserver, but a webserving community:
  - users cooperatively improve availability (through site replication/mirroring)
  - users cooperatively liberate firewalled content (through P2P proxying/relaying)
  - access to specific content can be restricted by simply listing who in the community can access it (through communal single sign-on).
  - site always available at the same URL regardless of physical location of content

# Alternatives

- Run your own webserver software (e.g. Apache httpd, Microsoft IIS, etc..)
- Centralized Storage (e.g. free or paid hosting services)
- Other P2P apps (Napster, Gnutella, XDegrees, Freenet, etc...)

# YouServ vs. Centralized Storage

- Cheaper. Uses storage, compute power, & bandwidth you already have.
- Easier.
  - Download & install, login, you're good to go.
  - Shared files always local (for disconnected operation).
  - Functions geared towards effective file sharing (e.g. built in ZIP function for easily sharing multi-file content).
- More private
- Fewer restrictions (e.g. some hosting services forbid MP3's)
- Automatic load distribution
- Know exactly who is accessing what, and when.

# YouServ vs. other P2P Apps

- With other P2P apps, accessing content requires:
  - you install special client software, or
  - you install a special purpose browser plugin, or
  - you route through (semi)centralized web proxy

With YouServ, ALL content is ALWAYS served directly from the peers via standard web protocols (DNS + HTTP)

# Deployment Details

- Running in IBM for about 1 year (though many important features were completed more recently)
  - Any IBM employee can use it to publish.
  - Anyone on IBM Intranet can access content.
- Deployed at Carnegie Mellon University last month.
  - Anyone with a cmu.edu e-mail address can publish.
  - Anyone on the internet can access (secured) content.

`http://youserv.com/`

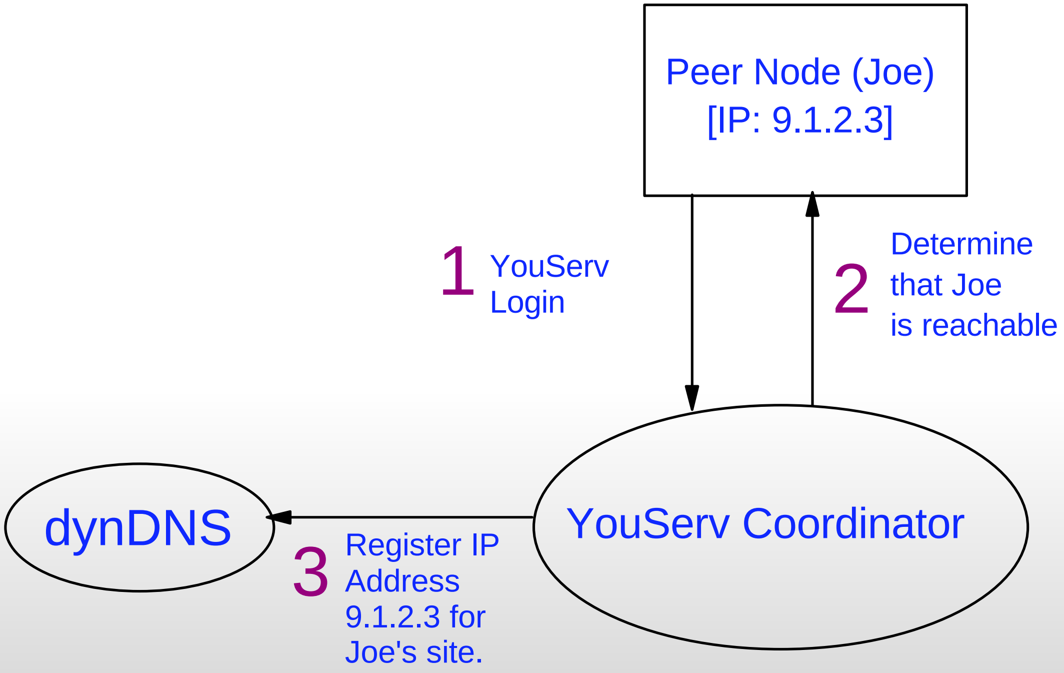
# Usage (IBM)

- 3400+ unique individuals have published a site with uServ.
- 1300+ of those sites were available in the last week.
- 800+ sites available simultaneously during peak hours, 400+ on weeknights, 300+ on weekends.
- Used quite differently than typical webserver software: many users share NO html content, only files:
  - digital photos
  - presentations, papers, work documents
  - live video feeds from their offices!
  - ...

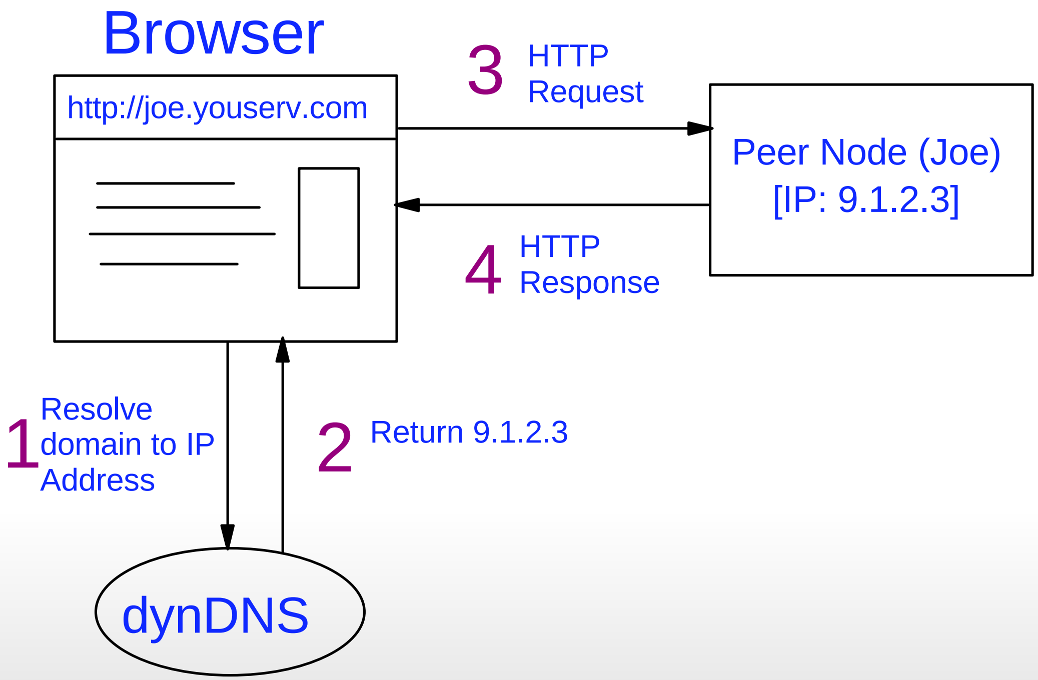
# How does it work?

- 4 system components
  - YouServ Coordinator (centralized)
  - YouServ Dynamic DNS (centralized)
  - YouServ Peer Nodes (end user publishers)
  - Browsers (end users accessing YouServ content)
- 3 access scenarios
  - Peer node is online: Standard site access
  - Peer node is offline: Peer-hosted site access
  - Peer node is firewalled: Proxied site access

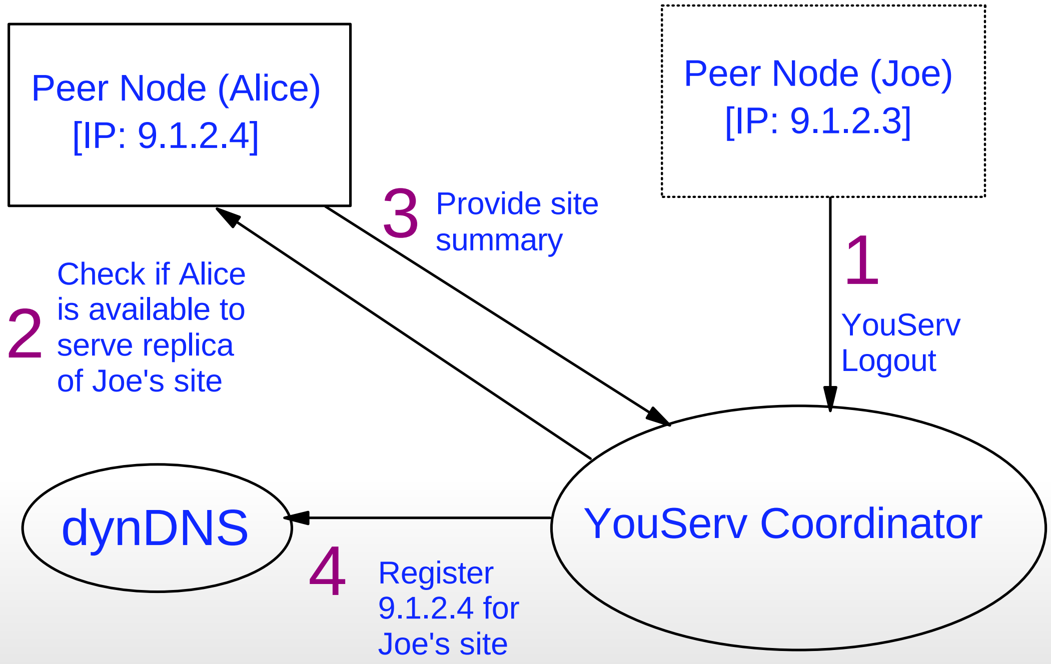
# Scenario 1: Standard (online node)


Peer Node (Joe)
[IP: 9.1.2.3]
1 YouServ Login
2 Determine that Joe is reachable
dynDNS
YouServ Coordinator
3 Register IP Address 9.1.2.3 for Joe's site.

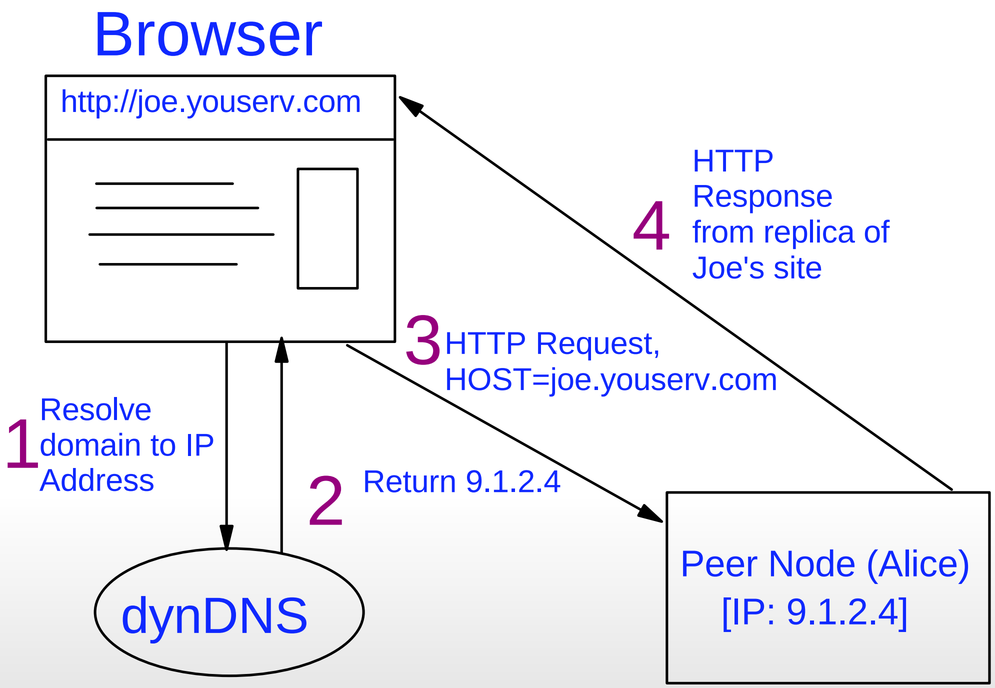
# Scenario 1: Standard (online node)



# Scenario 2: Peer Hosted



# Scenario 2: Peer Hosted


Browser
http://joe.youserv.com
1 Resolve domain to IP Address
2 Return 9.1.2.4
3 HTTP Request, HOST=joe.youserv.com
4 HTTP Response from replica of Joe's site
dynDNS
Peer Node (Alice)
[IP: 9.1.2.4]

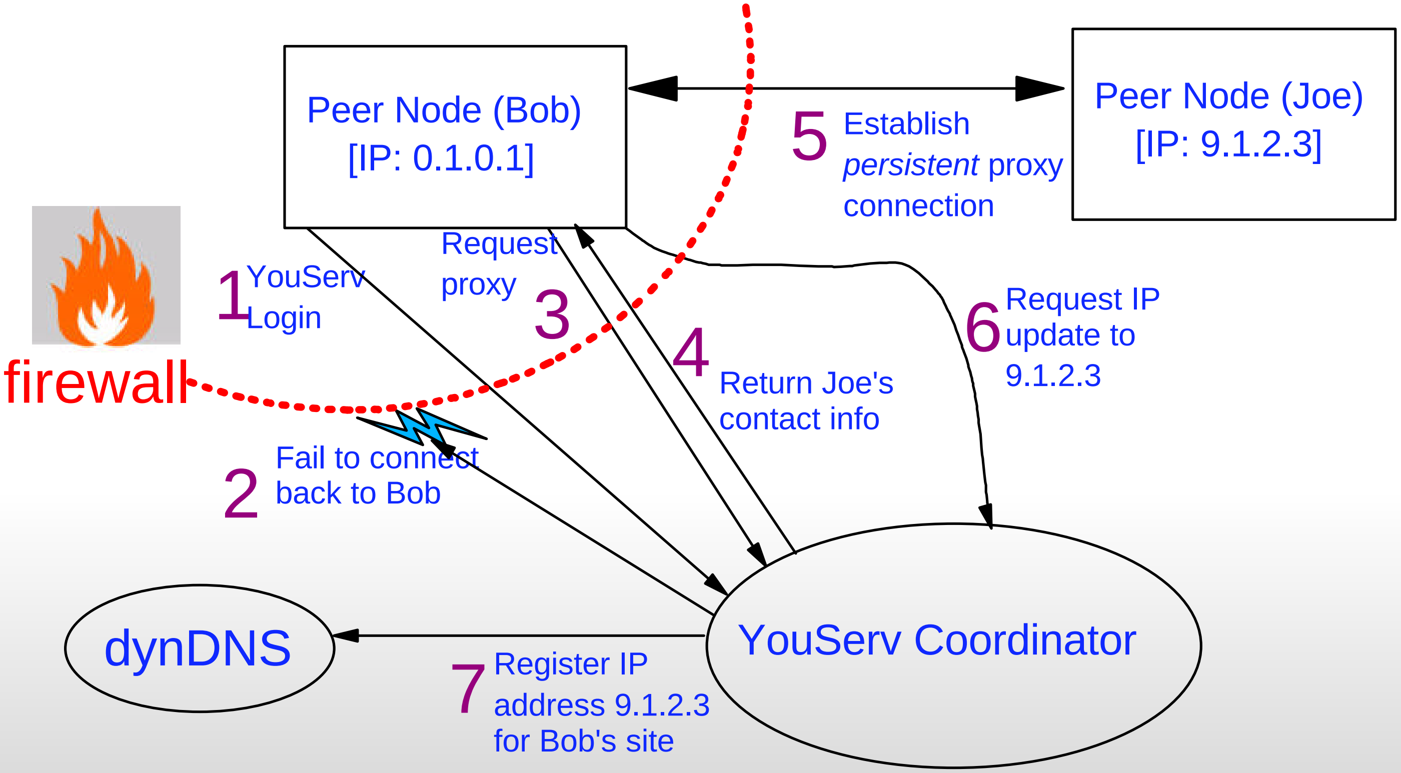
# Replication Details

- Peers themselves are almost entirely responsible for replica maintenance.
  - Coordinator's role is only to notify peers of presence and provide authenticating tokens for peers to communicate.
- Terminology:
  - *Replicator*: Peer who replicates some other peer's site
  - *Master*: Peer whose site is being replicated.
- A replicator periodically compares a "site summary" to that of the master.
  - Also serves to detect when a master site is unavailable, at which point the replicator will initiate replica failover.

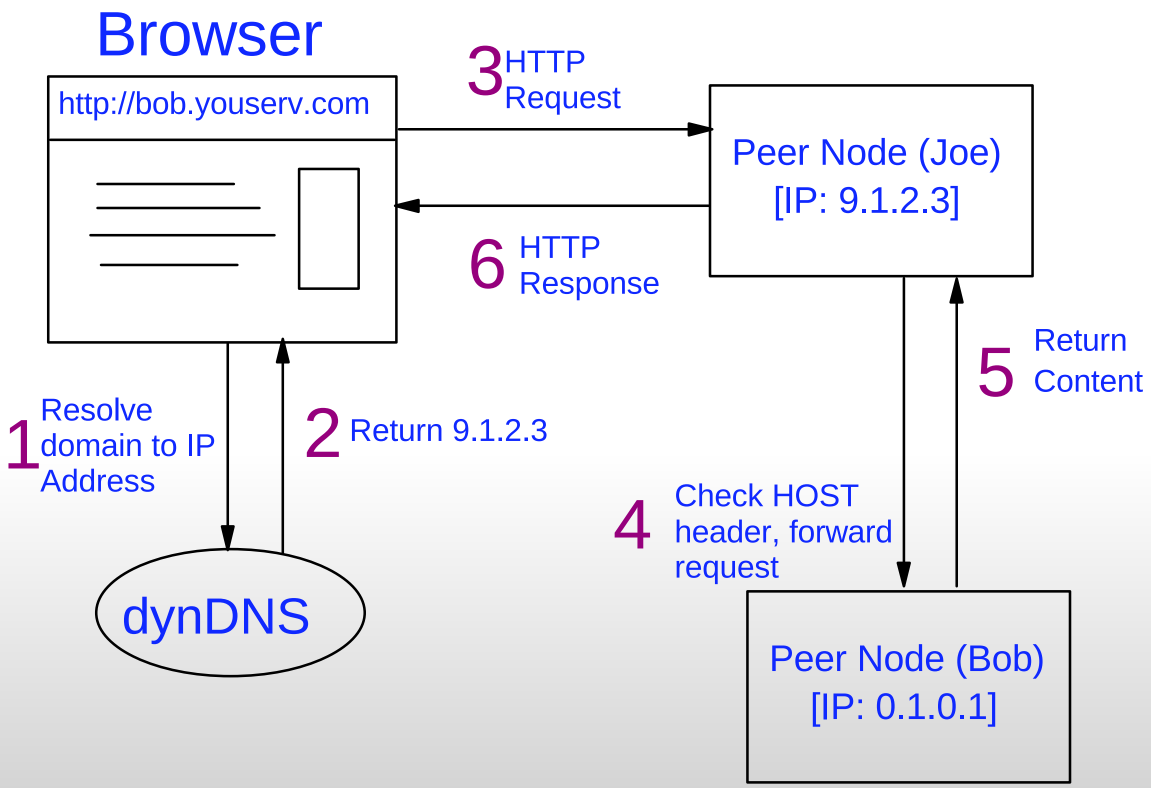
# Replication Details

- If site summary fails to match, the replicator will initiate a phase for determining precisely which files and folders need to be updated or deleted.
- Files that are new or have changed are downloaded via HTTP GET (in their entirety).

# Scenario 3: Proxied


Peer Node (Bob)
[IP: 0.1.0.1]
Peer Node (Joe)
[IP: 9.1.2.3]
firewall
1 YouServ Login
2 Fail to connect back to Bob
Request proxy 3
4 Return Joe's contact info
5 Establish *persistent* proxy connection
6 Request IP update to 9.1.2.3
7 Register IP address 9.1.2.3 for Bob's site
dynDNS
YouServ Coordinator

# Scenario 3: Proxied


Browser
http://bob.youserv.com
3 HTTP Request
Peer Node (Joe) [IP: 9.1.2.3]
6 HTTP Response
5 Return Content
1 Resolve domain to IP Address
2 Return 9.1.2.3
4 Check HOST header, forward request
dynDNS
Peer Node (Bob) [IP: 0.1.0.1]

# Add'l Proxying Details

- Coordinator maintains list of "good" proxy candidates
  - Responsive connection
  - Consistently available
- Location of proxy is heursitically determined by login ID. (E.g. "@us.ibm.com => US, @aus.ibm.com => Australia).
- Coordinator always tries to refer a user to a good candidate that is also proximal.

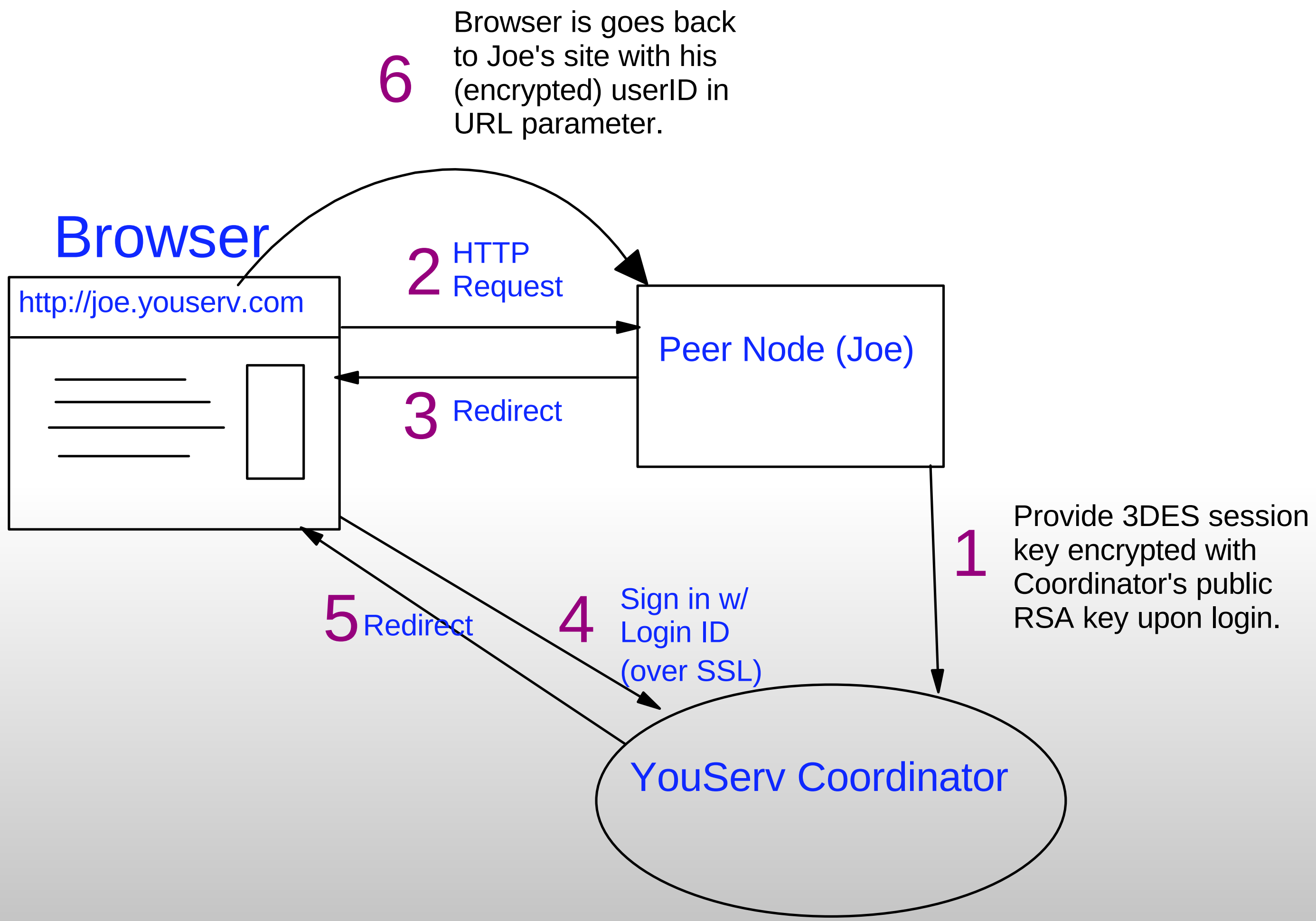
# Access Control & Authentication in YouServ

- Accessing secure content across multiple YouServ sites should be seamless.
  - Don't want to require sites assign their own accounts and passwords to secure content.
- Accessing secure content should not require YouServ sites to be trusted.
  - Don't want sites to directly receive a single-signon password to avoid password stashing.

# Authentication

- Authentication provided via single sign on scheme similar to Microsoft Passport.
  - Passwords are *never* directed to individual YouServ sites.
  - Passwords are only validated through a secure authentication server over SSL.
  - After signing in once with your password, you can authenticate with *any* YouServ site with a single click (until browser session ends).

# Single Sign-on Authentication



6 Browser is goes back to Joe's site with his (encrypted) userID in URL parameter.

Browser

http://joe.youserv.com

2 HTTP Request

Peer Node (Joe)

3 Redirect

1 Provide 3DES session key encrypted with Coordinator's public RSA key upon login.

4 Sign in w/ Login ID (over SSL)

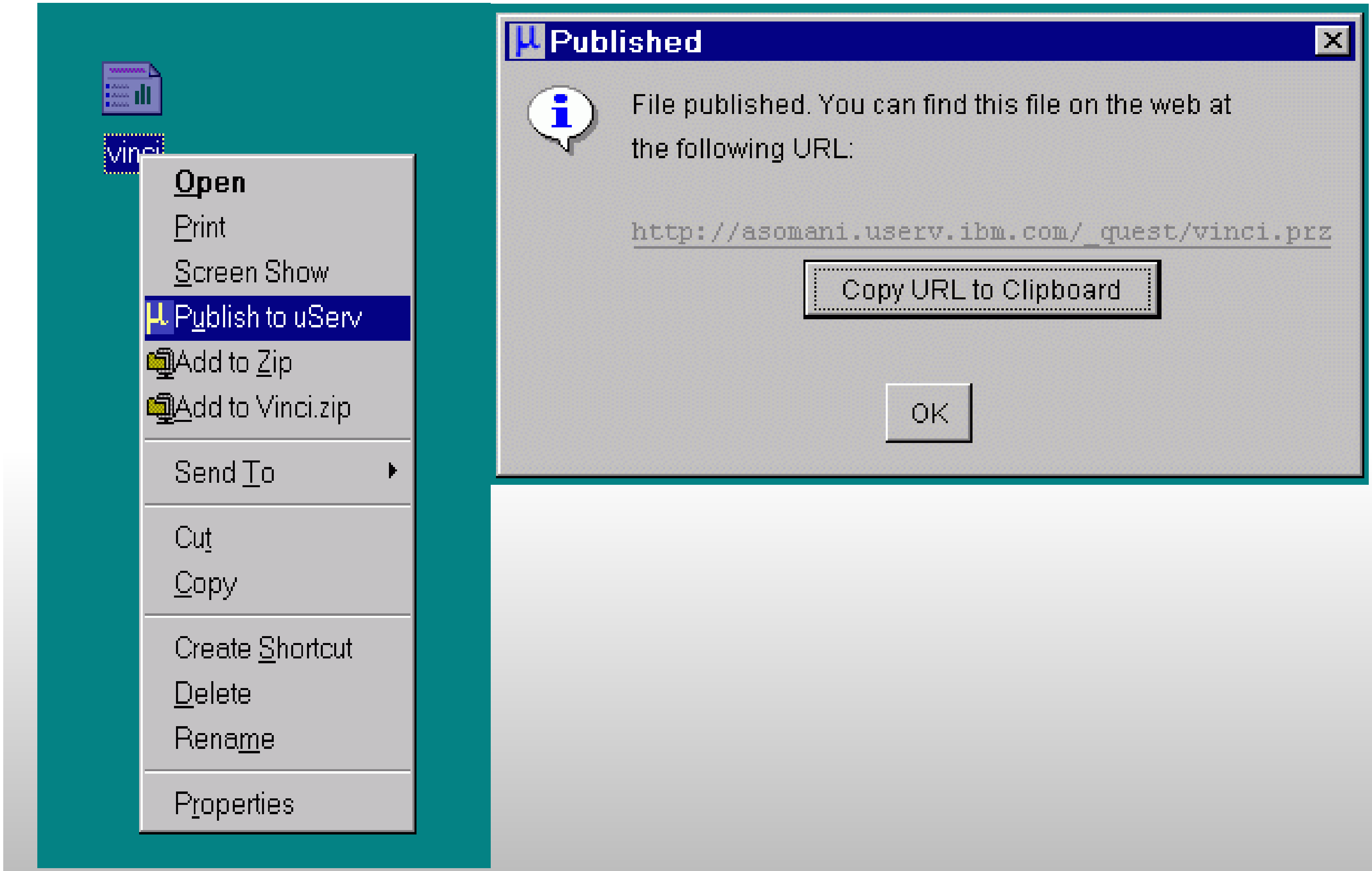
5 Redirect

YouServ Coordinator

# Scalability

- Potential Bottlenecks: DNS & Coordinator
- Coordinator:
  - validates passwords (low bandwidth, easy to scale)
  - Monitors availability
    - Availability assitance provided by replicator peers polling their masters.
- DNS:
  - Low bandwidth
  - highly optimized nameservers (BIND), easy to distribute
  - Existing services known to be highly scalable and cheap to operate (some funded by donations alone).

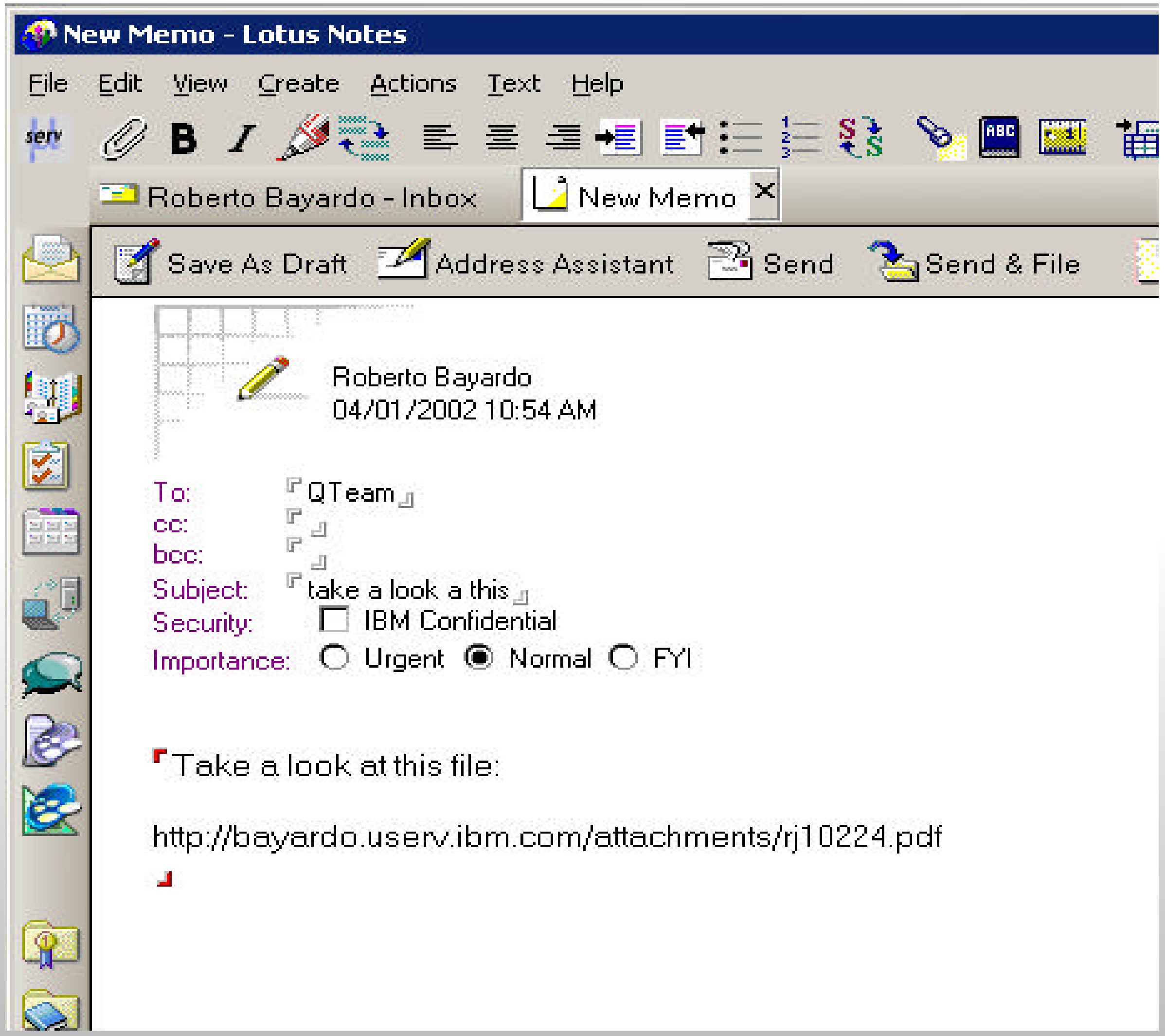
# Desktop Integration



File published. You can find this file on the web at the following URL:

http://asomani.userv.ibm.com/_quest/vinci.prz

# Application Integration



Roberto Bayardo
04/01/2002 10:54 AM

To: QTeam

cc:

bcc:

Subject: take a look a this

Security: □ IBM Confidential

Importance: ○ Urgent ● Normal ○ FYI

Take a look at this file:

http://bayardo.userv.ibm.com/attachments/rj10224.pdf

# Future Plans

- Search capability
  - While YouServ sites can be indexed in the "standard" way by search index crawlers....
    - YouServ sites are more dynamic than typical sites
    - Many YouServ sites still quite transient
  - Routing based search methods are still not very good.
    - Limited horizon
    - bandwidth hog
    - theoretical log / root(n) approaches not proven in practice
  - How can we provide fast, effective, up to date search over YouServ content?

# Future Plans

- Plugin API for adding services to YouServ servers
  - Allow extending the functionality without access to kernel code.
  - Similar to WinAmp skins, but for function, not for appearance.
- Open source?

# In Closing...

- YouServ: an end-user P2P application with uses other than piracy :-)
  - Everyone should be able to easily publish whatever they want and as much as they want on the web.
- Key challenge: engineering the system to work within the constraints imposed by standard browser software and web protocols.
  - Limitations are becoming painful.
  - Will protocols (and implementations) have to start evolving to get much further?