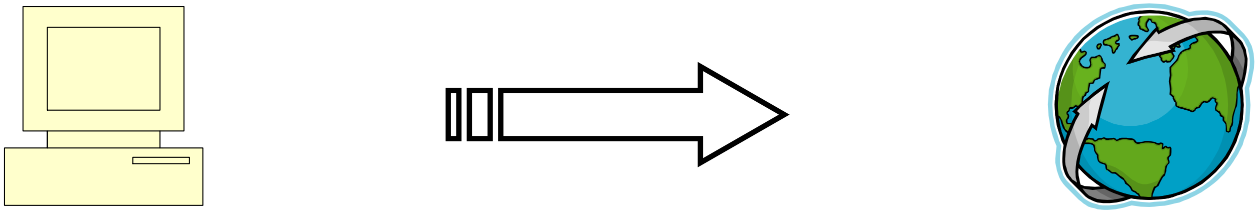
Organizing and Sharing Distributed Web Objects with Menagerie

Roxana Geambasu, Cherie Cheung, Alex Moshchuk, Steve Gribble, Hank Levy

University of Washington

The transition onto the Web



Desktop

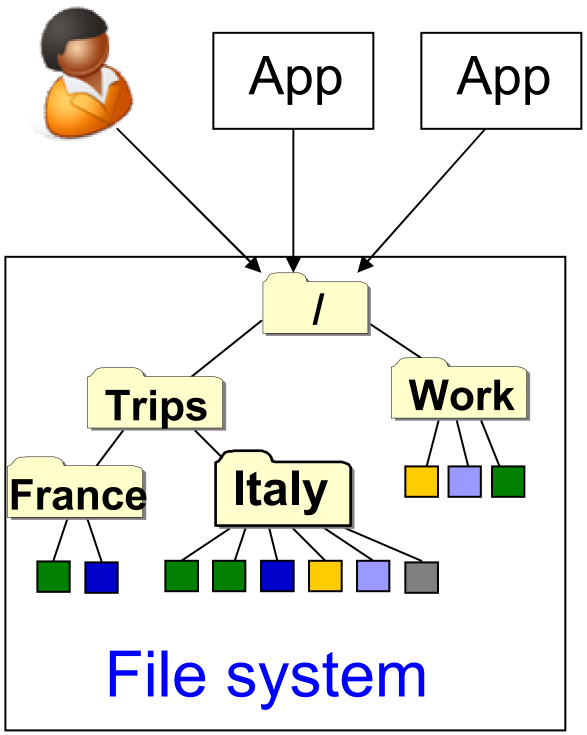
- Desktop applications
  - □ Office productivity
  - □ Email, news clients
  - □ File sharing (Kazaa)
- PC-based storage
  - □ Ext3, NFS

Web Services

- Software-as-a-service apps
  - □ThinkFree, Google Docs
  - □ Web-based email, news
  - □ Social networking sites
- Internet-based storage
  - □ Amazon S3, iBackup

Desktop advantages

- The user’s data is tightly integrated within a single FS



🟩 photos
🟨 spreadsheets
🟦 videos
… …

Users can:

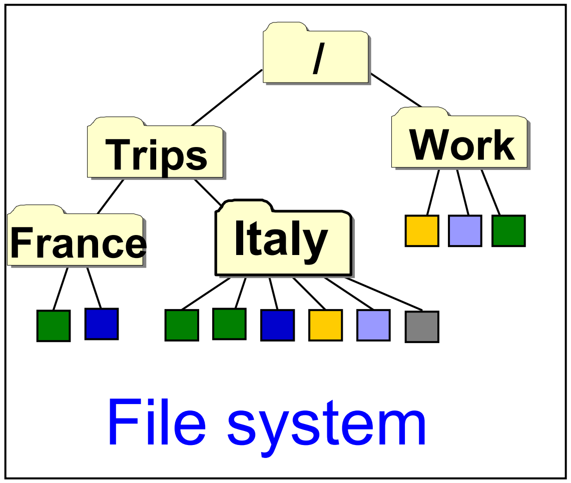
- Organize their files into folders
- Process files using applications
- Protectively share files with other users of the system

The FS provides functions to support these tasks

The Web lacks these advantages

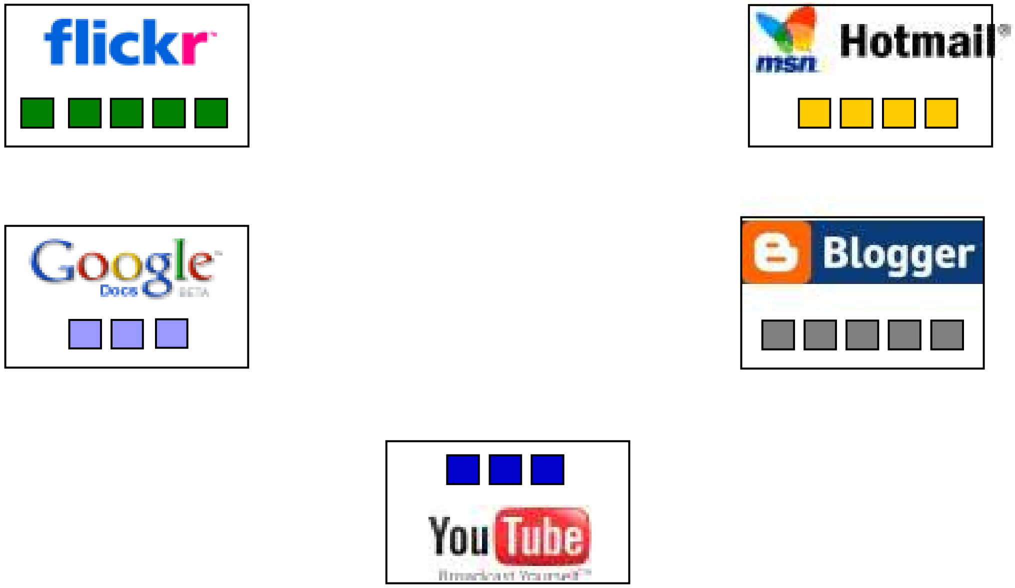
Desktop → Web

Data is integrated



File system

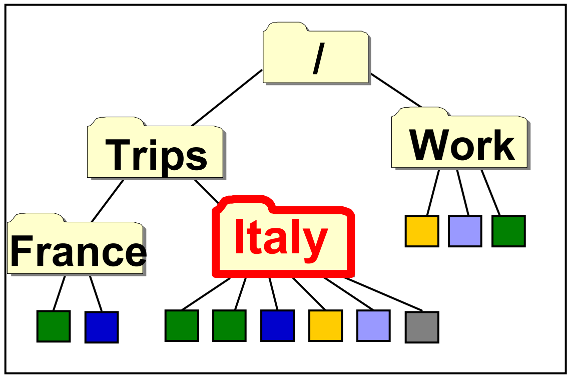
Data is scattered

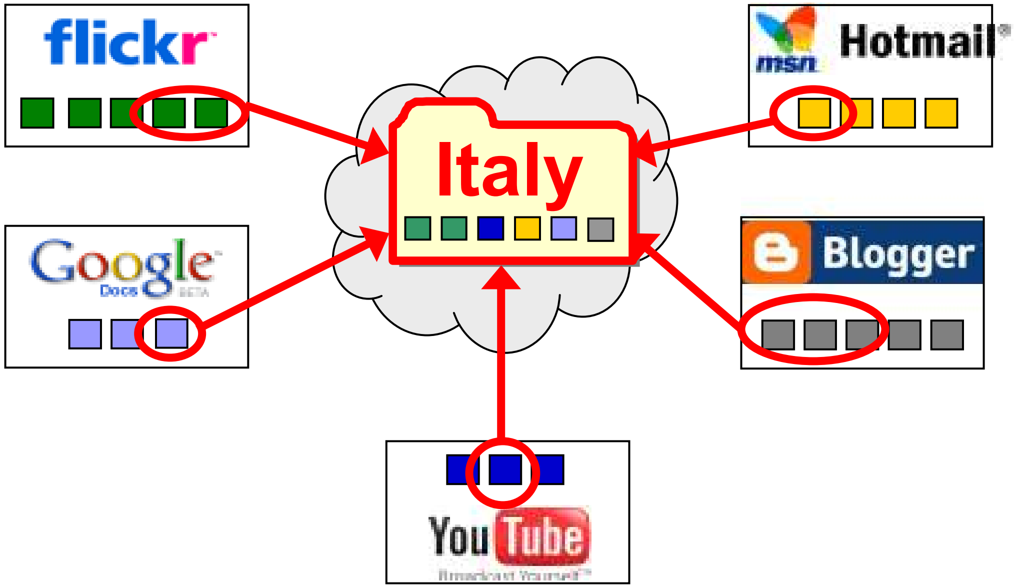


Challenge 1: Organizing personal data

Desktop → Web

How to organize?

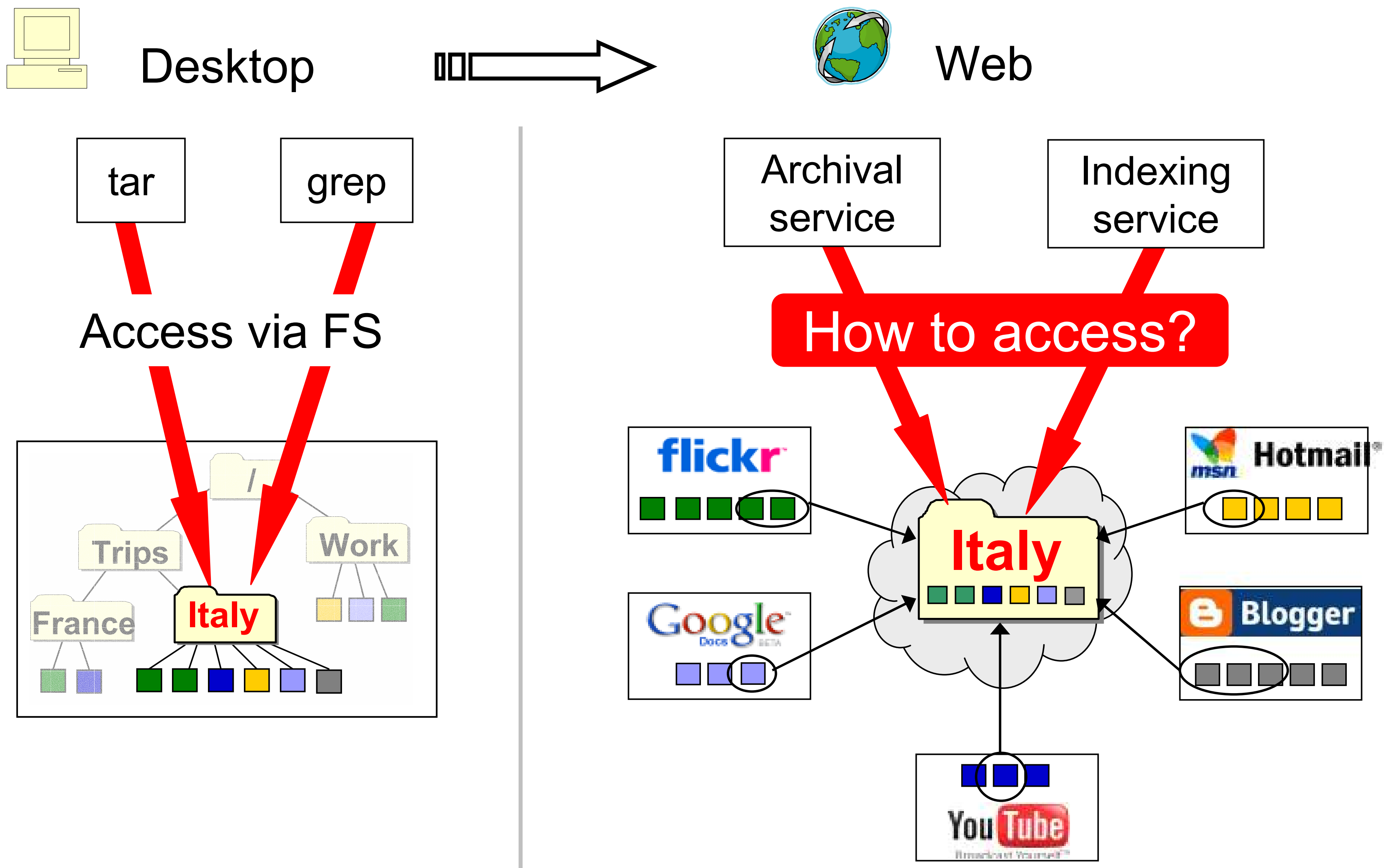




Lots of tools for organization:

File managers, PIM systems

Challenge 2: Processing data



Desktop

Web

tar

grep

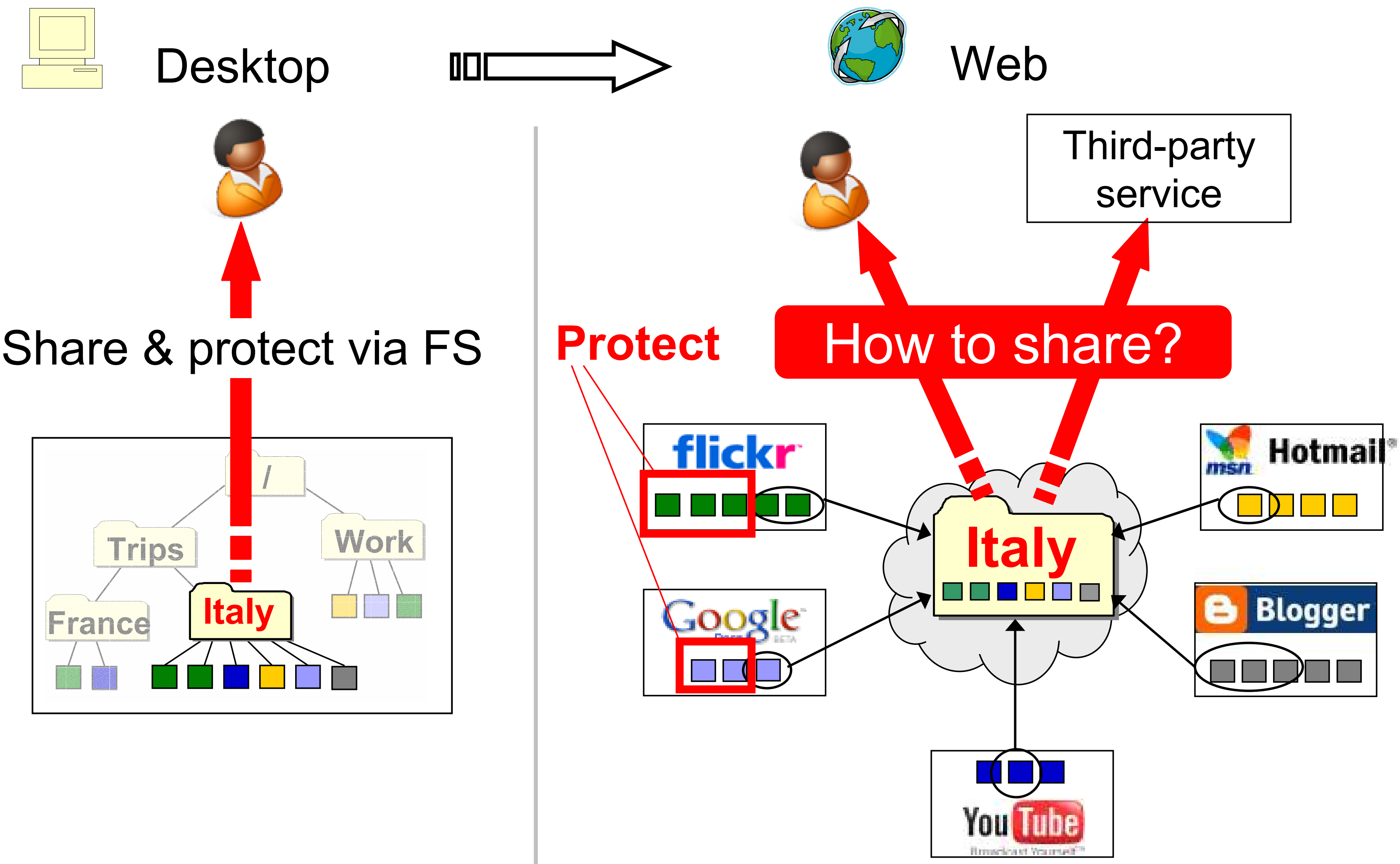
Access via FS

Archival service

Indexing service

How to access?

Challenge 3: Sharing data protectively



Desktop

Web

Share & protect via FS

Third-party service

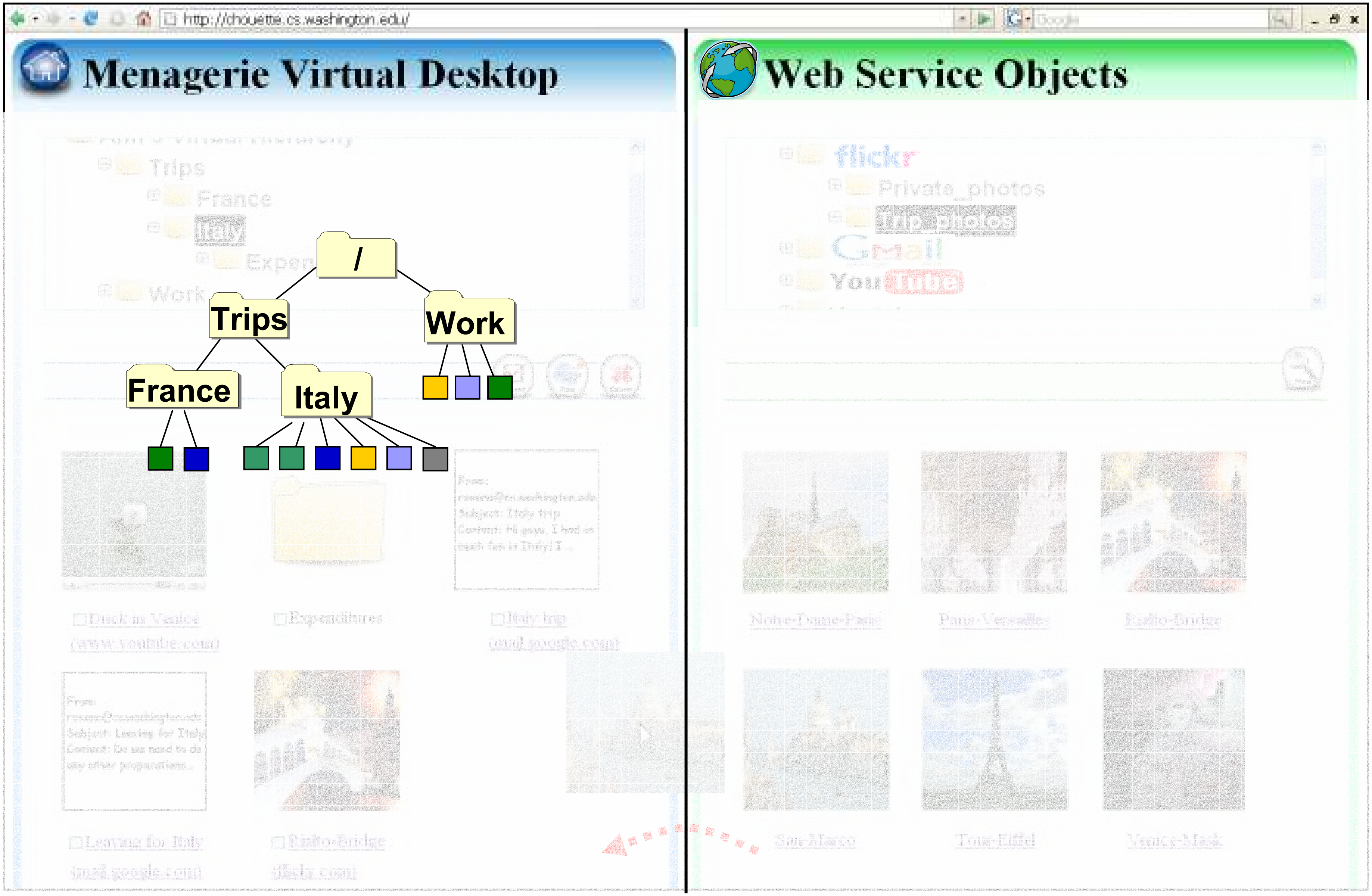
Protect

How to share?

Overview

- Three challenges:
  - □ *Organizing* web objects into heterogeneous folders
  - □ *Processing* heterogeneous folders
  - □ *Protected sharing* of heterogeneous folders
- Example application
- Menagerie
- Evaluation
- Related work
- Conclusions

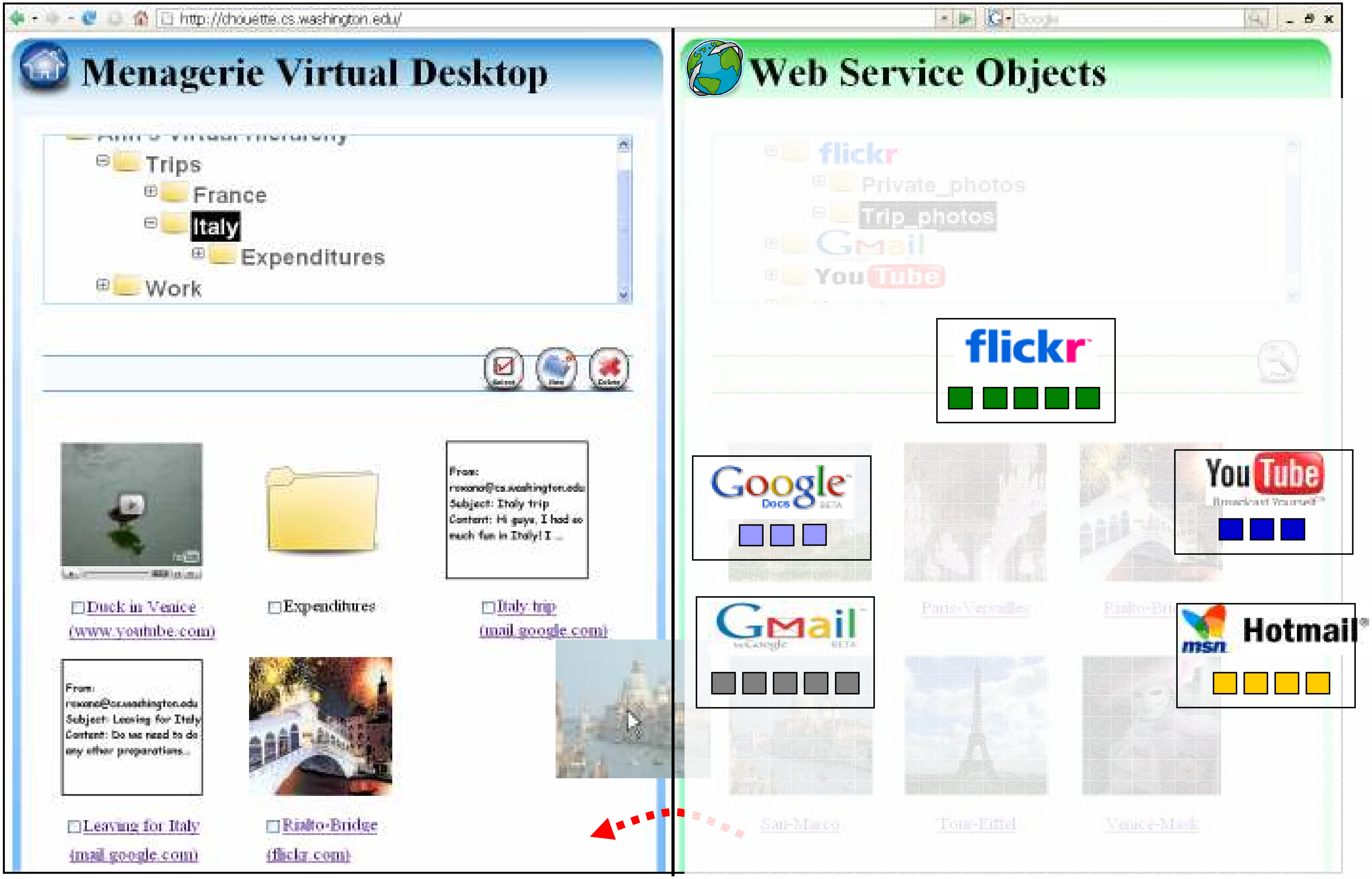
The Menagerie Web Object Manager


http://chouette.cs.washington.edu/
Menagerie Virtual Desktop
Web Service Objects
/
Trips
Work
France
Italy
Expen
flickr
Private_photos
Trip_photos
Gmail
YouTube
Duck in Venice (www.youtube.com)
Expenditures
Italy trip (mail.google.com)
Leaving for Italy (mail.google.com)
Rialto-Bridge (flickr.com)
Notre-Dame-Paris
Paris-Versailles
Rialto-Bridge
San-Marco
Tour-Eiffel
Venice-Mask

From: roxana@cs.washington.edu
Subject: Italy trip
Content: Hi guys, I had so much fun in Italy! I ...

From: roxana@cs.washington.edu
Subject: Leaving for Italy
Content: Do we need to do any other preparations...

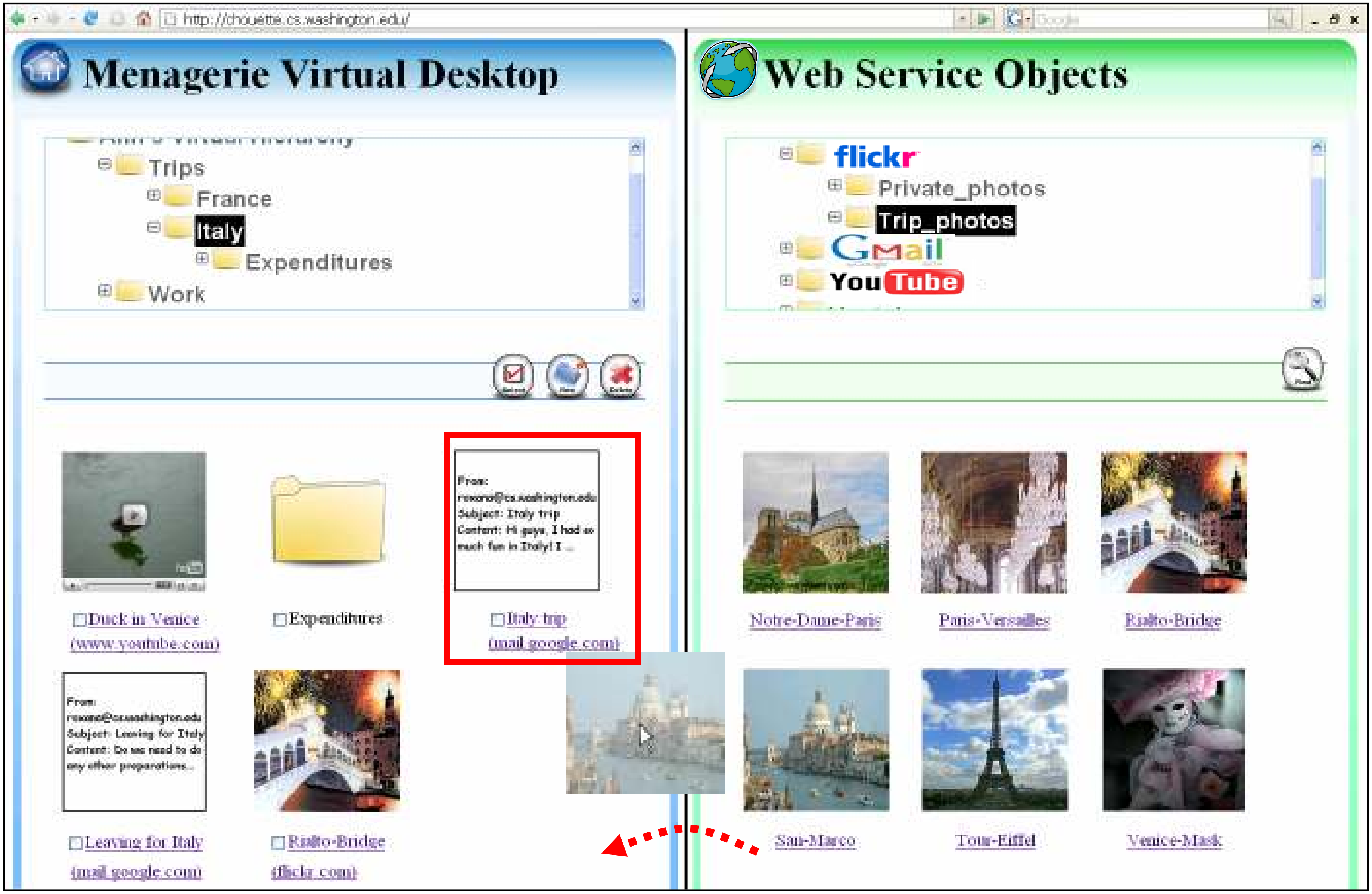
The Menagerie Web Object Manager



From: roxana@cs.washington.edu
Subject: Italy trip
Content: Hi guys, I had so much fun in Italy! I ...

From: roxana@cs.washington.edu
Subject: Leaving for Italy
Content: Do we need to do any other preparations...

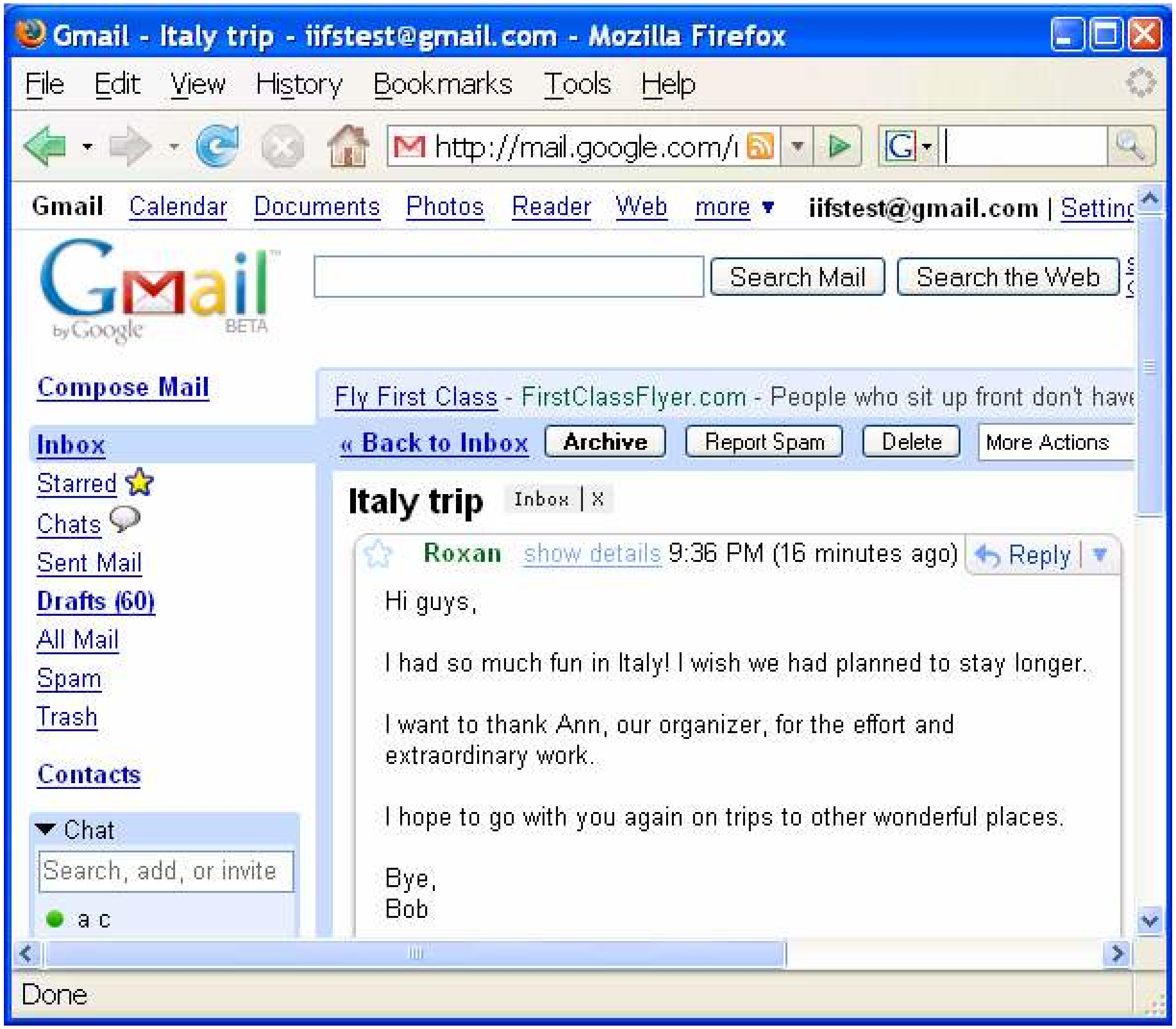
The Menagerie Web Object Manager



From: roxana@cs.washington.edu
Subject: Italy trip
Content: Hi guys, I had so much fun in Italy! I ...

From: roxana@cs.washington.edu
Subject: Leaving for Italy
Content: Do we need to do any other preparations...

The Menagerie Web Object Manager



Gmail - Italy trip - iifstest@gmail.com - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://mail.google.com/
Gmail Calendar Documents Photos Reader Web more ▾ iifstest@gmail.com | Settinc
Gmail by Google BETA
Search Mail
Search the Web
Compose Mail
Inbox
Starred
Chats
Sent Mail
Drafts (60)
All Mail
Spam
Trash
Contacts
Chat
Search, add, or invite
a c
Fly First Class - FirstClassFlyer.com - People who sit up front don't have
« Back to Inbox
Archive
Report Spam
Delete
More Actions
Inbox | X
Reply
Done

Italy trip

Roxan show details 9:36 PM (16 minutes ago)

Hi guys,

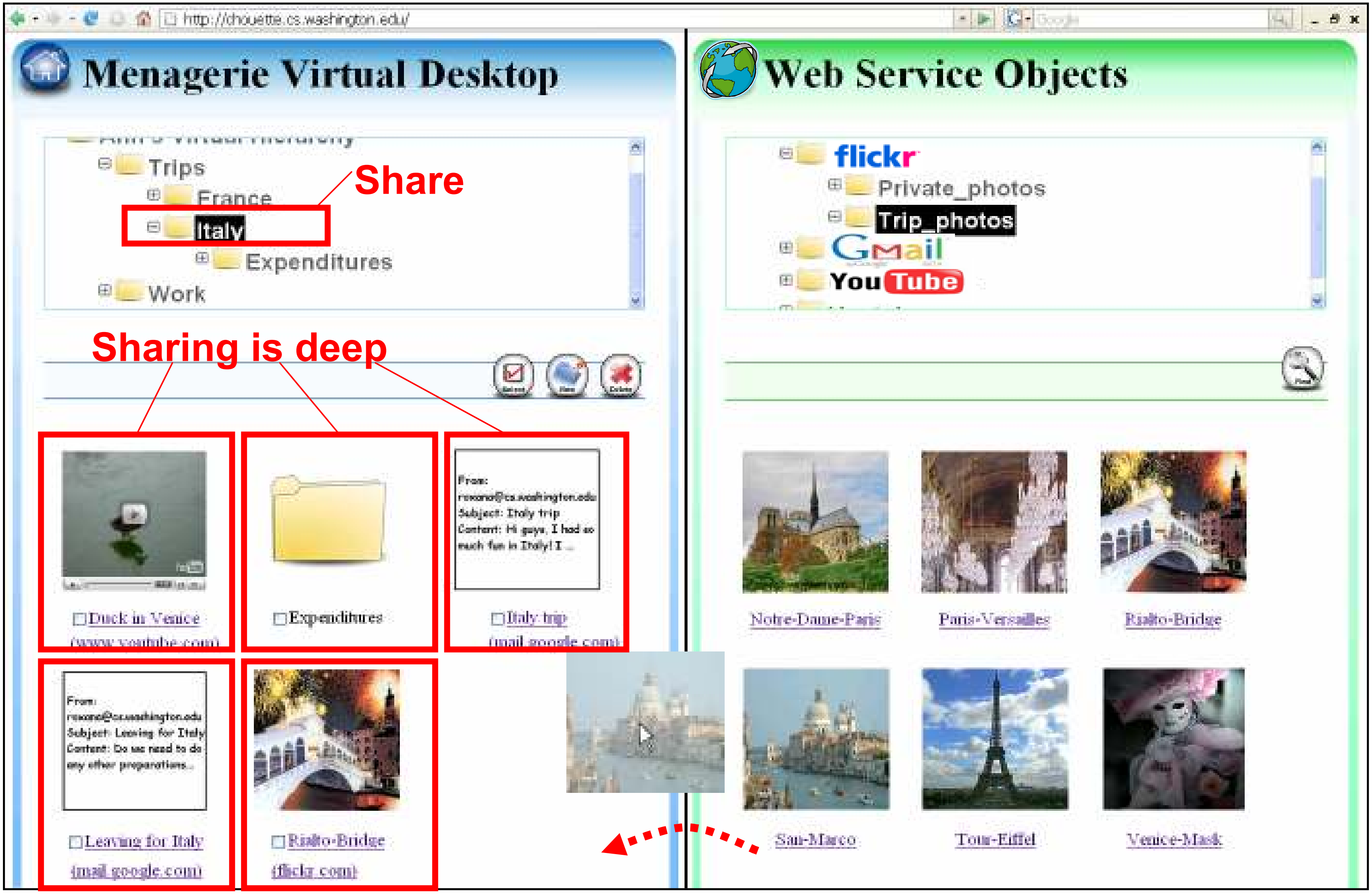
I had so much fun in Italy! I wish we had planned to stay longer.

I want to thank Ann, our organizer, for the effort and extraordinary work.

I hope to go with you again on trips to other wonderful places.

Bye,
Bob

The Menagerie Web Object Manager – Sharing



Share

Sharing is deep

From: roxana@cs.washington.edu
Subject: Italy trip
Content: Hi guys, I had so much fun in Italy! I ...

From: roxana@cs.washington.edu
Subject: Leaving for Italy
Content: Do we need to do any other preparations...

Overview

- Three challenges:
  - *Organizing* web objects into heterogeneous folders
  - *Processing* heterogeneous folders
  - *Protected sharing* of heterogeneous folders
- Example application
- Menagerie
- Evaluation
- Related work
- Conclusions

Menagerie

- Framework for building applications for personal Web-data management and sharing
- Provides a set of unified infrastructure functions
  - On the desktop, the FS provided these functions

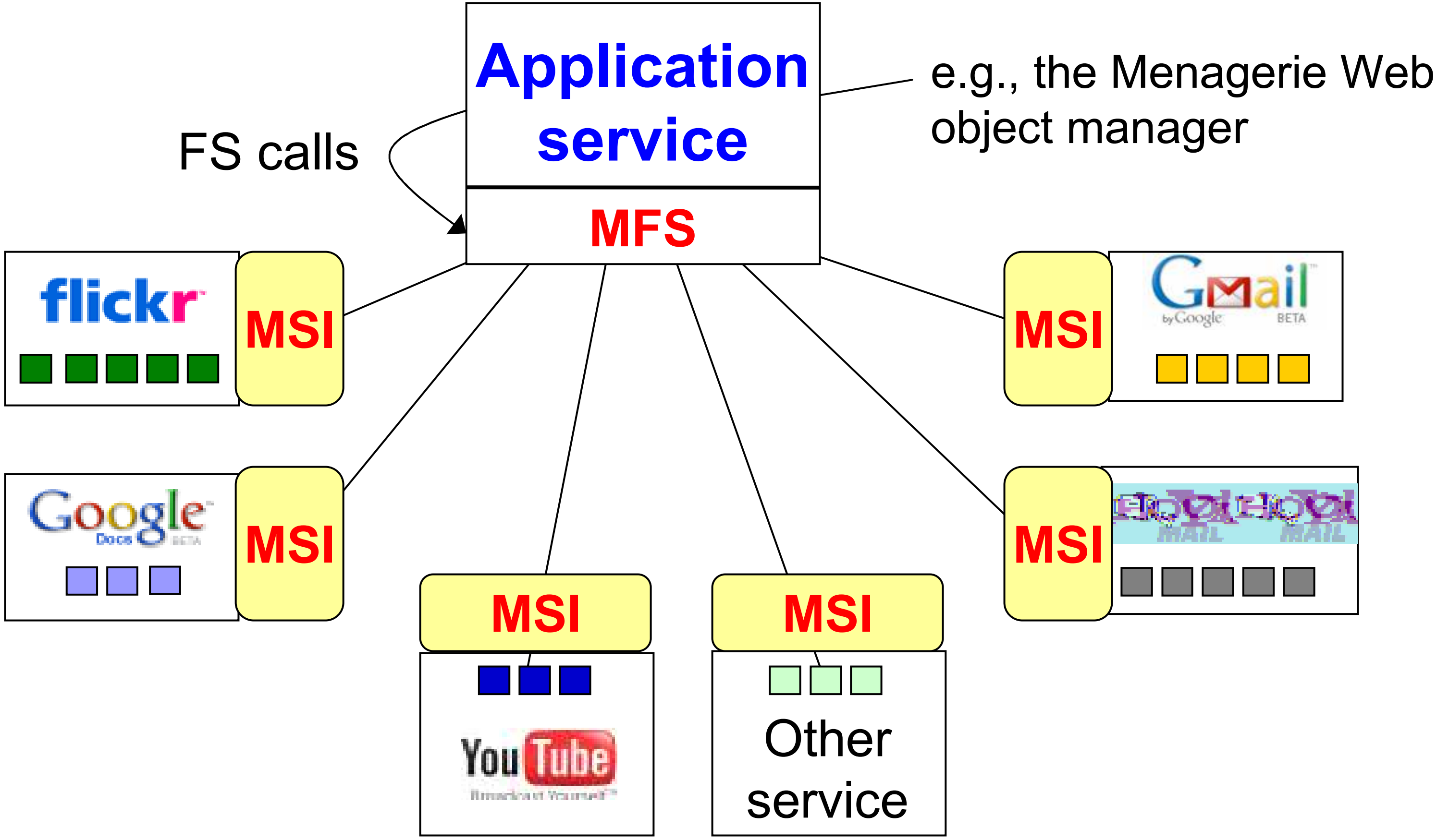
Menagerie functions:

1. Common object naming
2. Common access to object content
3. Common fine-grained protection

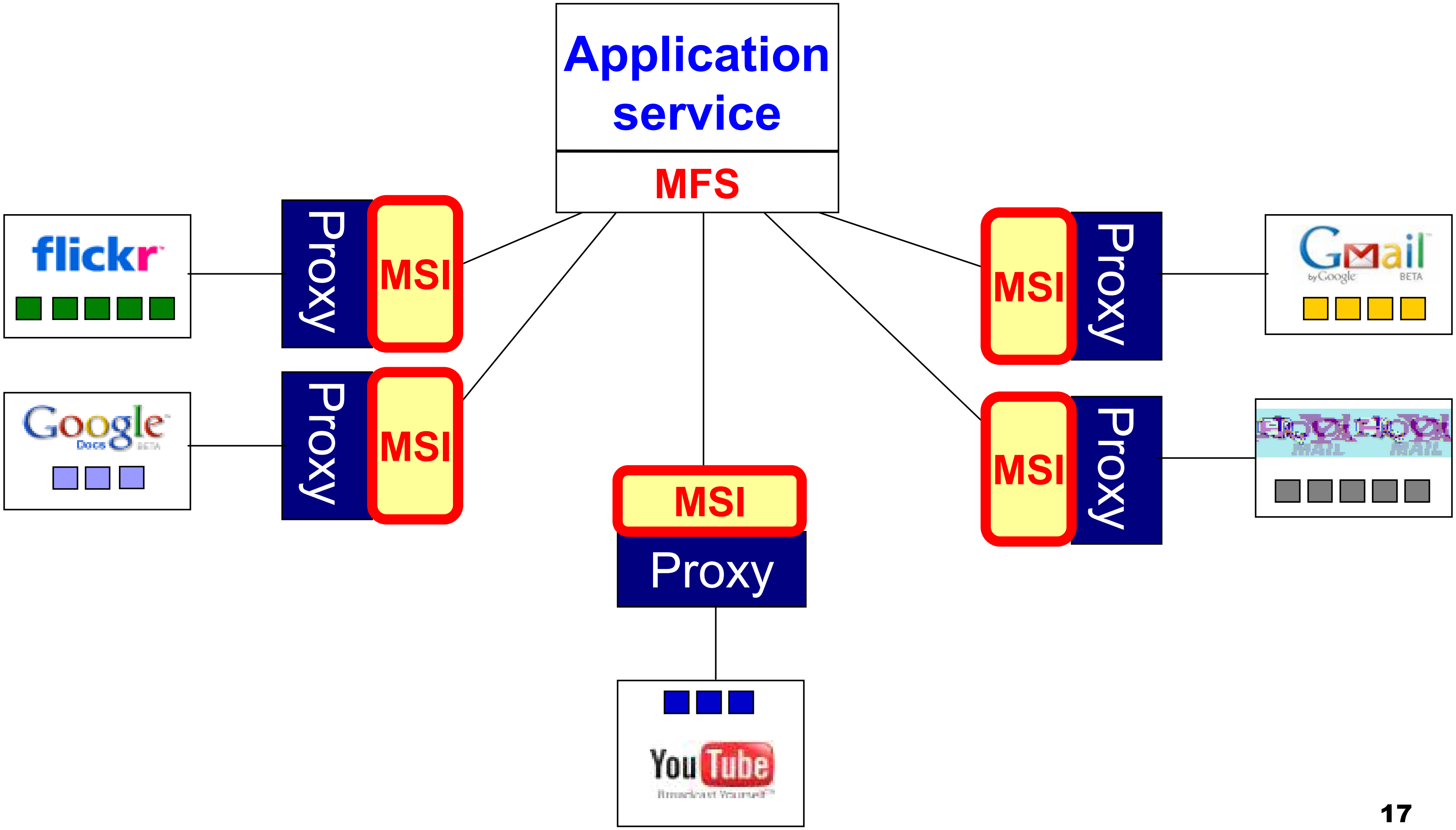
- These functions are motivated by the challenges identified in motivation

Menagerie architecture

- The Menagerie Service Interface (MSI)
  - Similar to **OpenSocial**, but for personal data organization
- The Menagerie File System (MFS)



The Menagerie prototype



The Menagerie Service Interface (MSI)

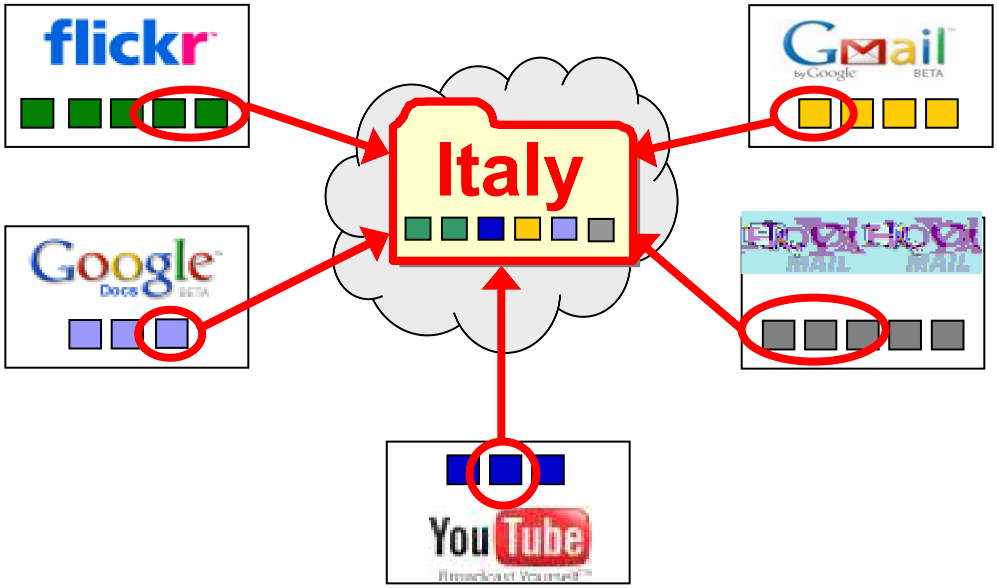
- Common service API

Contains operations for:

1. Common object naming
2. Common access to object content
3. Common fine-grained protection

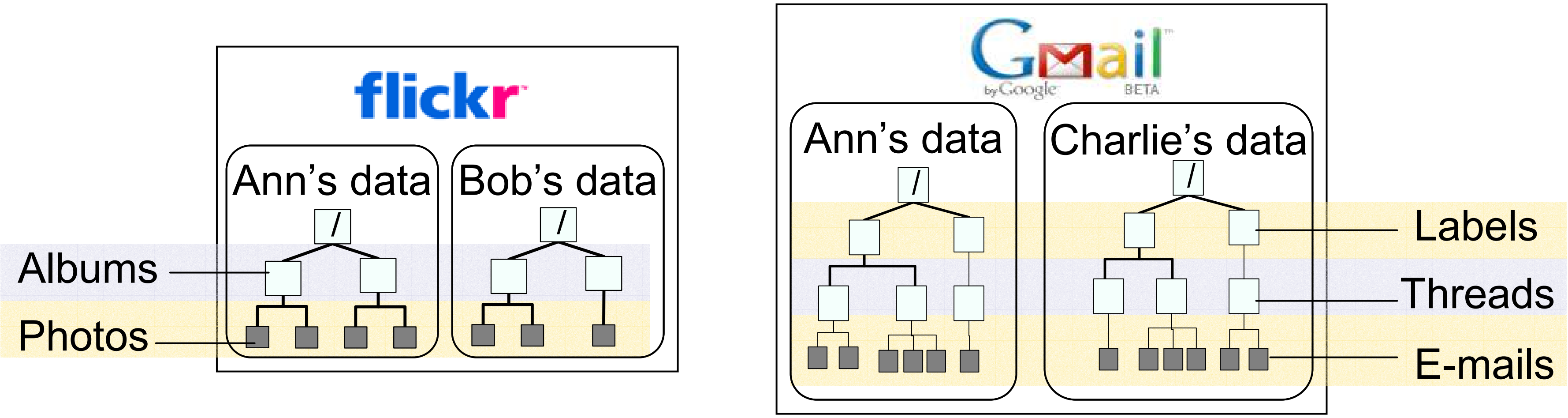
1. Common object naming

- Supports creation of heterogeneous folders



1. Common object naming

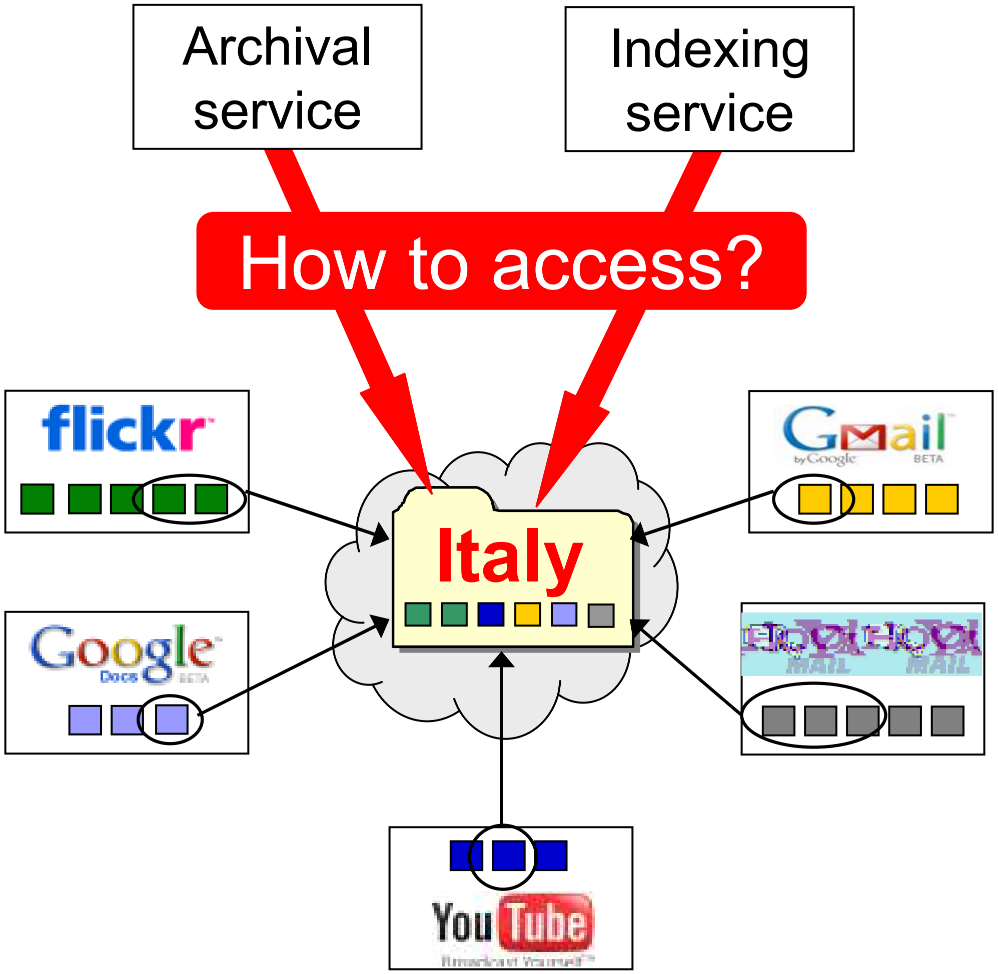
- Supports creation of heterogeneous folders
- Each service exports a hierarchical namespace of each user’s objects



- MSI has operations for navigating and altering the namespace: **`ls(), mkdir()`**

2. Common access to object content

- Supports processing of objects and collections



Archival service

Indexing service

How to access?

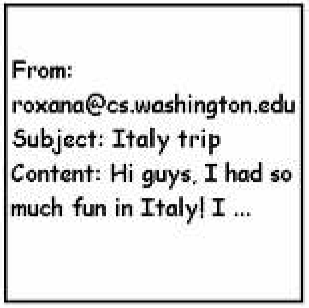
Italy

2. Common access to object content

- Supports processing of objects and collections
- Opaque object content operations

```
read(), write()
```

- Embedded rendering
  - □ Each service provides a summary HTML tag for each object



From:
roxana@cs.washington.edu
Subject: Italy trip
Content: Hi guys, I had so much fun in Italy! I ...

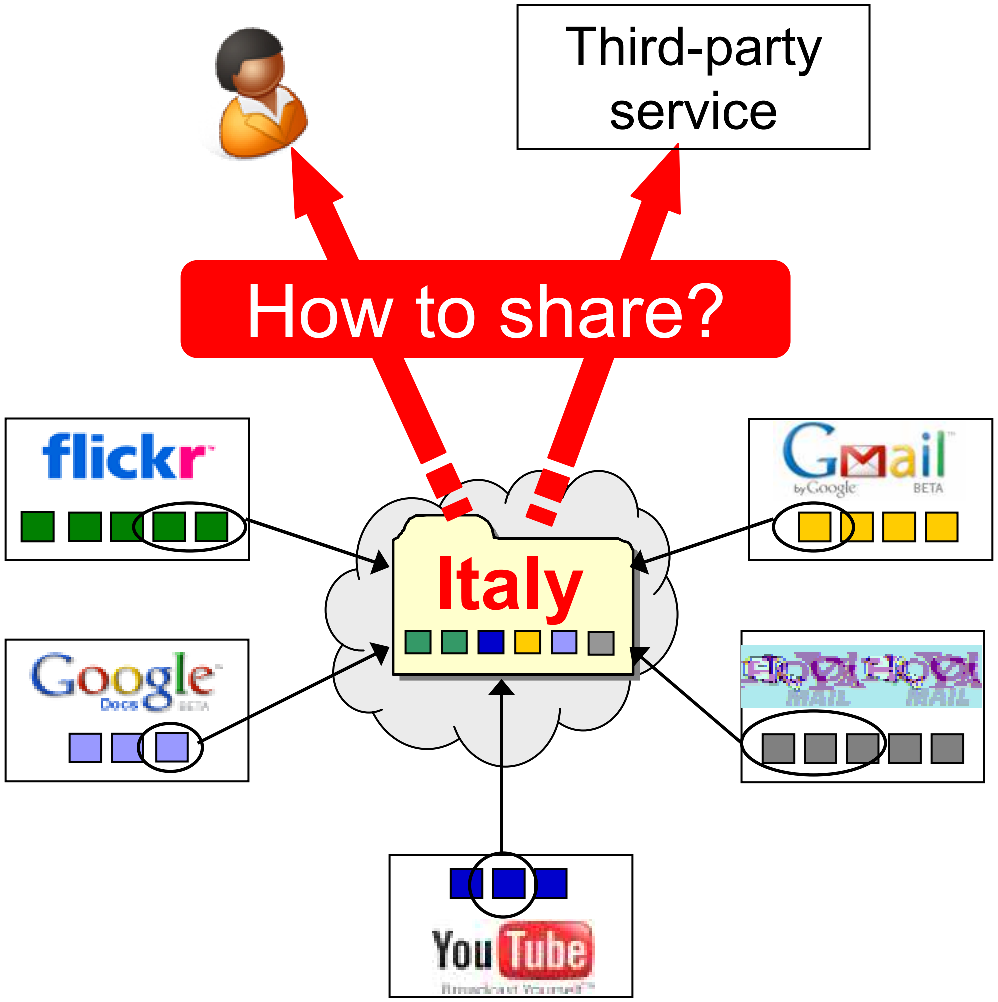
Email
(HTML snippet)

Youtube
(<object> tag)

Flickr
(<img> thumbnail)

3. Common fine-grained protection

- Supports protected sharing of heterogeneous folders



3. Common fine-grained protection

- Supports protected sharing of heterogeneous folders

- To facilitate fine-grained sharing we use capabilities

- A Menagerie capability is an unforgeable token
  - Bundles together a globally unique object name & a set of access rights
  - Provides the holder with authority to execute the specified actions on the named object

- Sharing using capabilities is just like emailing URLs

3. Common fine-grained protection

- Menagerie capabilities give services a choice:
  - □ Allow *direct access* to web objects based on capabilities
  - □ Require *authentication* in addition to the capability to provide access
- Authentication allows services to track and control access to their objects
- MSI protection functions:

```
create_capa(), revoke_capa()
```

Overview

- Three challenges:
  - *Organizing* web objects into heterogeneous folders
  - *Processing* heterogeneous folders
  - *Protected sharing* of heterogeneous folders
- Example application
- Menagerie
- Evaluation
- Related work
- Conclusions

Easy to build apps atop Menagerie

Application	Description	LOC
Menagerie Web Object Manager	Service for organizing and sharing web objects of any type	275 (php)
Web object Group Sharing	Service for sharing web objects of *any* type with a group (uses Gallery)	167 (php)
Web-data backup	Back up heterogeneous collections of web objects (uses **`tar`**)	10 (bash)
Contact synchronizer	Uses **`unison`** to synchronize contacts	20 (bash)

Overview

- Three challenges:
  - □ *Organizing* web objects into heterogeneous folders
  - □ *Processing* heterogeneous folders
  - □ *Protected sharing* of heterogeneous folders
- Example application
- Menagerie
- Evaluation
- Related work
- Conclusions

Related work

- Common web service interfaces
  - □ OpenSocial to support social networking apps [Google07]

  →Menagerie is more general and it is designed for personal data object management and sharing apps

- Web-data aggregation and clipping applications
  - □ iGoogle, SecondBrain, Yahoo! Pipes, Backpack

  →Menagerie can serve as infrastructure for such applications

  →Most do not support fine-grained sharing

- World Wide Web Without Walls (W5) [HotNets07]

  →Menagerie has similar vision, but adds concrete API and implementation

- Using OS abstractions to address Web problems
  - □ WebDAV[EuropeanConf99], Web file systems [TOCS98]

Conclusions

- The shift from the desktop to the Web raises problems:
  - data organization
  - data processing
  - protected sharing

- A small set of common operations enable powerful, generic applications on Web objects and folders
  - naming
  - content access
  - protection

- Menagerie brings these functions onto the Web

Appendix

Menagerie vs. OpenSocial: Similarities

- Facilitate applications by having services adhere to common API
  - Adherence to OpenSocial gives us hope that Menagerie will be adopted
- Similar concepts: uniform naming, uniform protection

Menagerie vs. OpenSocial: Differences

- Deal with different kinds of data:
  - Menagerie works with many types of personal data objects: photos, videos, word documents, spreadsheets, etc.
  - OpenSocial designed for social networking data: friends, their activities
- Designed for different types of applications:
  - Menagerie: web-object management and fine-grained sharing
  - OpenSocial: social networking apps and more coarse-grained sharing
- Different protection mechanisms:
  - OpenSocial: ACL-based sharing among friends
  - Menagerie: Capability-based protection that facilitates fine-grained sharing