

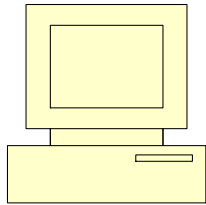
# Organizing and Sharing Distributed Web Objects with Menagerie

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

**University of Washington**

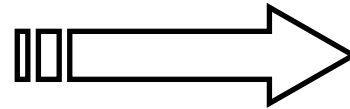


# The transition onto the Web



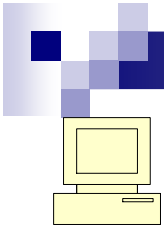
## Desktop

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



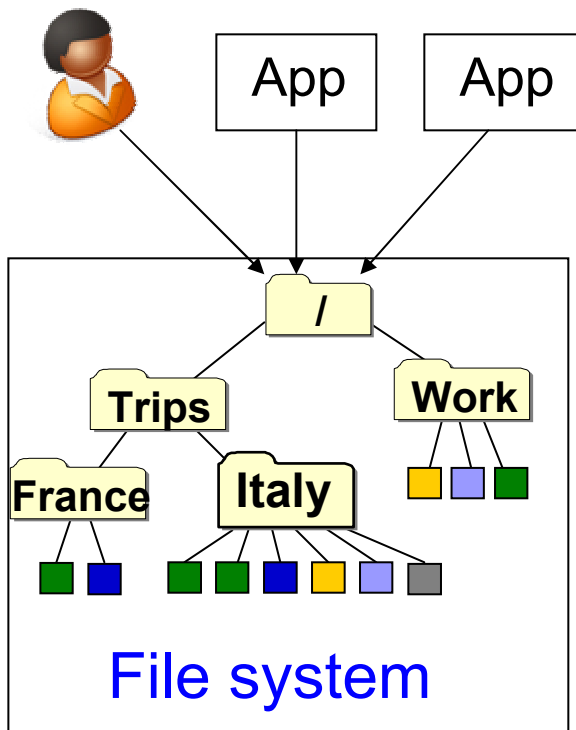
## Web Services

- **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**



Users can:

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

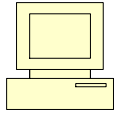
- photos
- spreadsheets
- videos

... ..

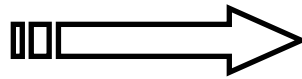
The FS provides functions to support these tasks



# The Web lacks these advantages



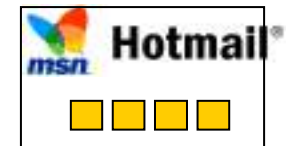
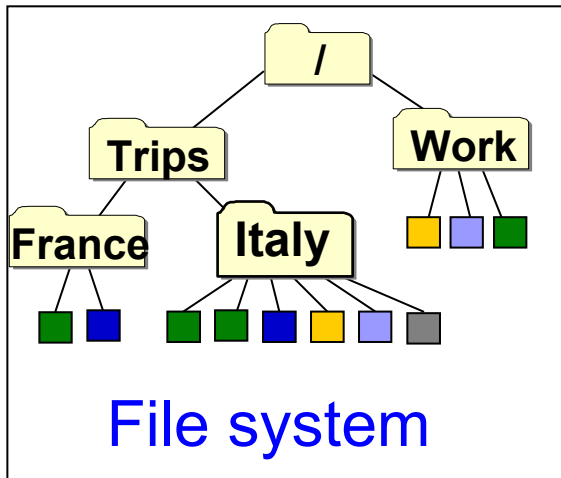
Desktop



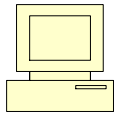
Web

Data is integrated

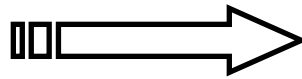
Data is scattered



# Challenge 1: Organizing personal data

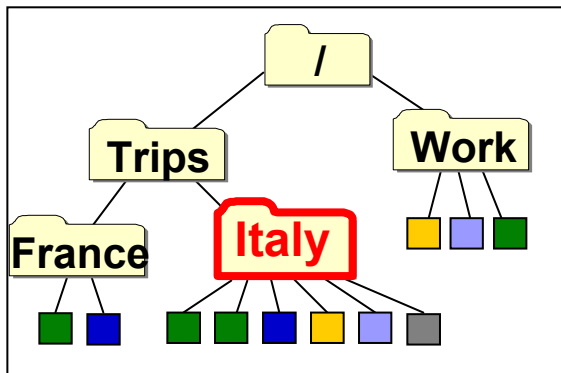


Desktop

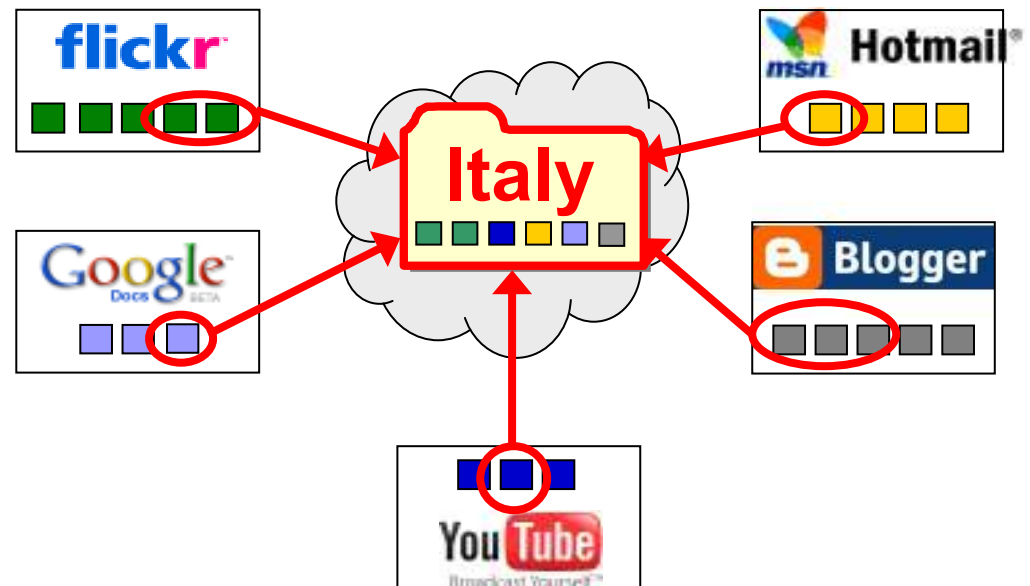


Web

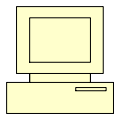
How to organize?



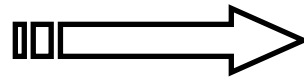
Lots of tools for organization:  
File managers, PIM systems



# Challenge 2: Processing data



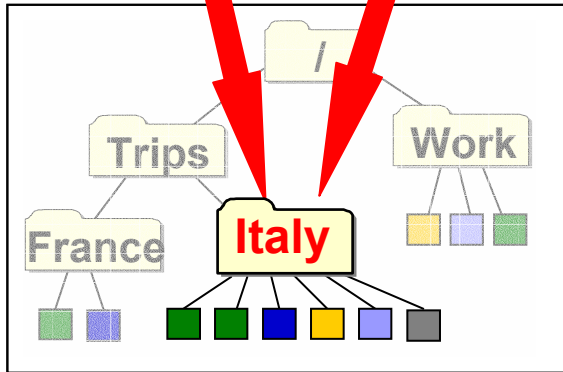
Desktop



Web

tar      grep

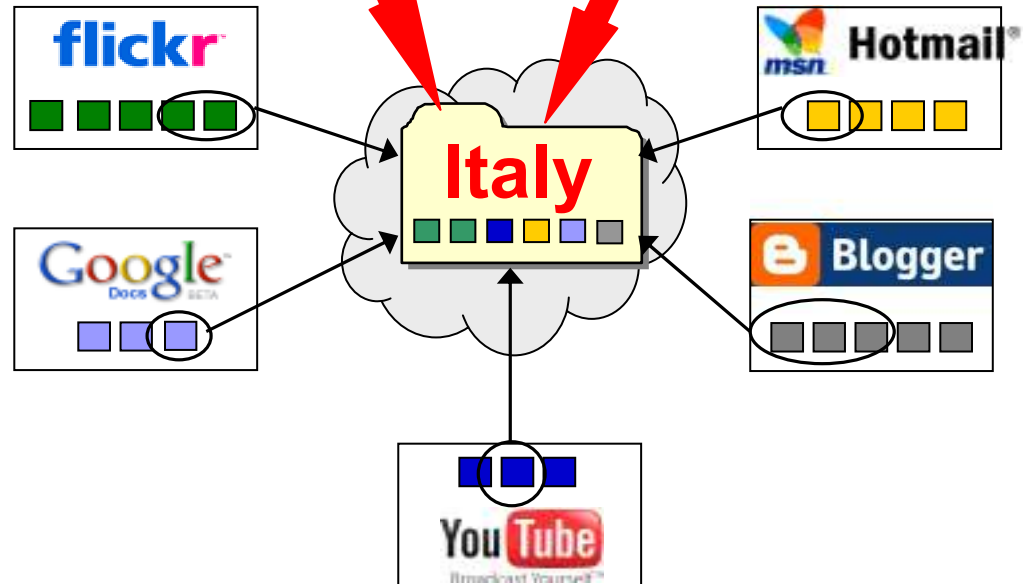
Access via FS



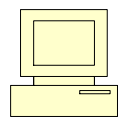
Archival service

Indexing service

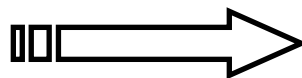
How to access?



# Challenge 3: Sharing data protectively



Desktop



Web

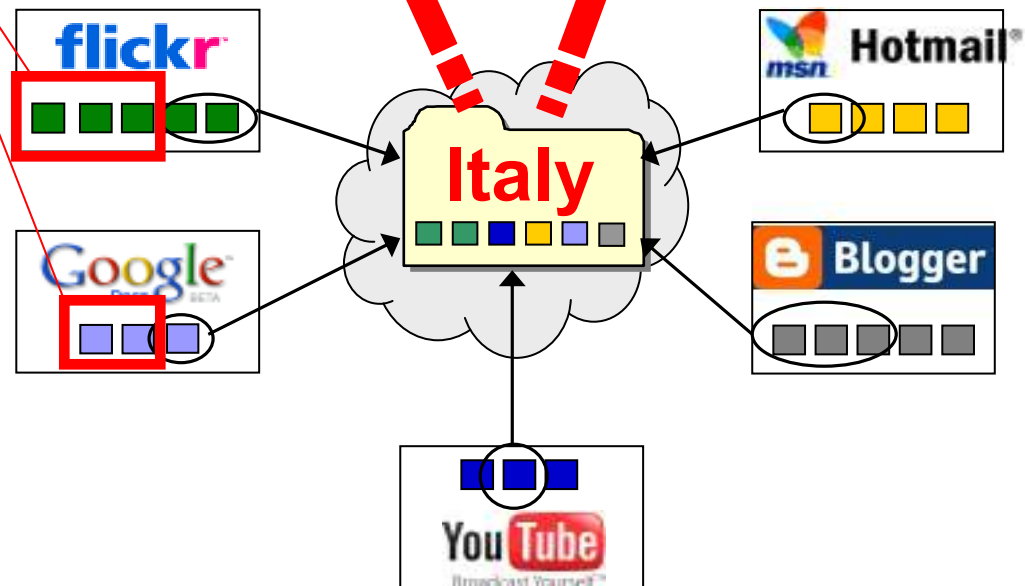
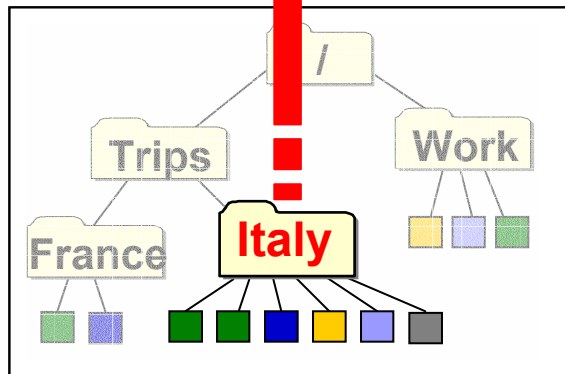


Share & protect via FS

Third-party service

How to share?

Protect





# 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

The image displays two side-by-side browser windows illustrating the Menagerie Web Object Manager interface.

**Menagerie Virtual Desktop (Left Window):** The address bar shows <http://chouette.cs.washington.edu/>. The title bar reads "Menagerie Virtual Desktop". The main content area features a hierarchical tree structure:

- Root: /
- Children: Trips, Work
- Trips Children: France, Italy
- France Children: [Green icon], [Blue icon]
- Italy Children: [Green icon], [Green icon], [Blue icon], [Yellow icon], [Purple icon], [Grey icon]
- Work Children: [Yellow icon], [Blue icon], [Green icon]

Below the tree, several web objects are displayed as thumbnails with associated metadata:

- Duck in Venice** (www.youtube.com): Includes a thumbnail and a small video player.
- Expenditures**: Includes a folder icon and a thumbnail.
- Italy trip** (mail.google.com): Includes a thumbnail and an email preview (From: raxxon@cs.washington.edu, Subject: Italy trip, Content: Hi guys, I had so much fun in Italy! I ...).
- Leaving for Italy** (mail.google.com): Includes a thumbnail and an email preview (From: raxxon@cs.washington.edu, Subject: Leaving for Italy, Content: Do we need to do any other preparations...).
- Rialto-Bridge** (flickr.com): Includes a thumbnail.

**Web Service Objects (Right Window):** The address bar shows "Google". The title bar reads "Web Service Objects". The main content area displays a list of web service objects:

- flickr
  - Private\_photos
  - trip\_photos
- Gmail
- YouTube

Below the list, a grid of photo thumbnails is shown, each with a caption:

- Notre-Dame-Paris
- Paris-Versailles
- Rialto-Bridge
- San-Misico
- Tour-Eiffel
- Venice-Mas.

A red dashed arrow points from the "Italy trip" email object in the left window to the "trip\_photos" object in the right window, indicating a link between the two.

# The Menagerie Web Object Manager

The image displays two side-by-side browser windows illustrating the Menagerie Web Object Manager interface.

**Left Window: Menagerie Virtual Desktop**  
The address bar shows <http://chouette.cs.washington.edu/>. The desktop features a file explorer showing a tree structure:

- Home - Virtual Menagerie
  - Trips
    - France
    - Italy**
    - Expenditures
  - Work

At the bottom, there are icons for Mail, News, and Calendar. The desktop contains several objects:

- A video thumbnail titled "Duck in Venice" with a link to [www.youtube.com](http://www.youtube.com).
- A folder icon titled "Expenditures".
- An email icon titled "Italy trip" with a link to [mail.google.com](mailto:mail.google.com). The email preview shows: "From: rexona@cs.washington.edu", "Subject: Italy trip", "Content: Hi guys, I had so much fun in Italy! I ...".
- An email icon titled "Leaving for Italy" with a link to [mail.google.com](mailto:mail.google.com). The email preview shows: "From: rexona@cs.washington.edu", "Subject: Leaving for Italy", "Content: Do we need to do any other preparations...".
- A photo thumbnail titled "Rialto-Bridge" with a link to [flickr.com](http://flickr.com).
- A photo thumbnail of a cityscape.

**Right Window: Web Service Objects**  
The address bar shows "Google". The interface displays a list of web service objects:

- flickr
  - Private\_photos
  - trip\_photos**
- Gmail
- YouTube

Below this list, several service icons are shown with progress indicators:

- flickr: 5 green progress bars.
- Google Docs BETA: 3 blue progress bars.
- YouTube (Powered by Windows): 3 blue progress bars.
- Gmail: 5 grey progress bars.
- Hotmail (msn): 4 yellow progress bars.

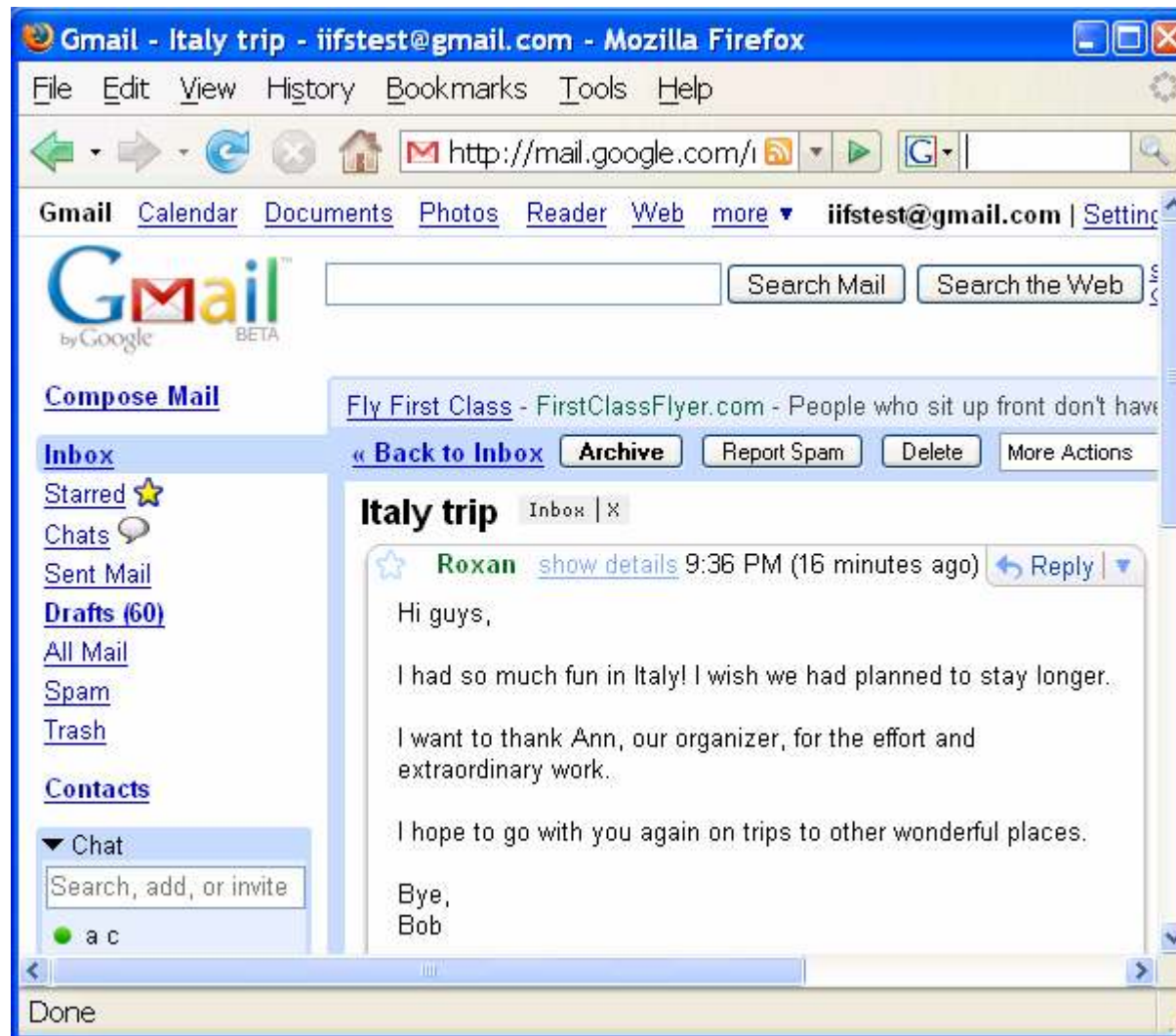
At the bottom, there are thumbnails for "San-Misco", "Tons-Eiffel", and "Venice-Mad". A red dashed arrow points from the cityscape thumbnail in the left window to the "San-Misco" thumbnail in the right window.

# The Menagerie Web Object Manager

The screenshot displays two browser windows side-by-side. The left window, titled "Menagerie Virtual Desktop", shows a file explorer view with a tree structure: "Trips" (containing "France" and "Italy"), "Expenditures", and "Work". Below the explorer are icons for "Duck in Venice (www.youtube.com)", "Expenditures", "Leaving for Italy (mail.google.com)", and "Rialto-Bridge (flickr.com)". A red box highlights an email object with the following text: "From: rexana@cs.washington.edu", "Subject: Italy trip", "Content: Hi guys, I had so much fun in Italy! I ...". Below this is the link "Italy trip (mail.google.com)". A red dashed arrow points from this email object to the right window.

The right window, titled "Web Service Objects", shows a file explorer view with a tree structure: "flickr" (containing "Private\_photos" and "Trip\_photos"), "Gmail", and "YouTube". Below the explorer are icons for "Notre-Dame-Paris", "Paris-Versailles", "Rialto-Bridge", "San-Marco", "Tour-Eiffel", and "Venice-Mask".

# The Menagerie Web Object Manager



# The Menagerie Web Object Manager – Sharing

The image displays two side-by-side browser windows. The left window, titled "Menagerie Virtual Desktop", shows a file explorer view with folders for "Trips", "France", "Italy", "Expenditures", and "Work". The "Italy" folder is highlighted with a red box and labeled "Share". Below the file explorer, a toolbar contains icons for "Share", "Share", and "Delete". A red arrow points from the "Share" label to the "Italy" folder. Below the toolbar, a grid of six objects is shown, each in a red-bordered box. These objects include a video thumbnail "Duck in Venice", a folder icon "Expenditures", an email icon "Italy trip", another email icon "Leaving for Italy", a photo thumbnail "Rialto-Bridge", and a photo thumbnail "San-Marco". A red dashed arrow points from the "Rialto-Bridge" object to the "Web Service Objects" window. The right window, titled "Web Service Objects", shows a folder view with "flickr", "Private\_photos", "Trip\_photos", "Gmail", and "YouTube". Below this, a grid of nine photo thumbnails is displayed, each with a caption: "Notre-Dame-Paris", "Paris-Versailles", "Rialto-Bridge", "San-Marco", "Tour-Eiffel", and "Venice-Mask".

**Menagerie Virtual Desktop**

Trips  
France  
**Italy**  
Expenditures  
Work

**Share**

**Sharing is deep**

Share Share Delete

Duck in Venice (www.youtube.com)

Expenditures

Italy trip (mail.google.com)

Leaving for Italy (mail.google.com)

Rialto-Bridge (flickr.com)

San-Marco

Tour-Eiffel

Venice-Mask

**Web Service Objects**

flickr  
Private\_photos  
Trip\_photos  
Gmail  
YouTube

Notre-Dame-Paris  
Paris-Versailles  
Rialto-Bridge  
San-Marco  
Tour-Eiffel  
Venice-Mask



# 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

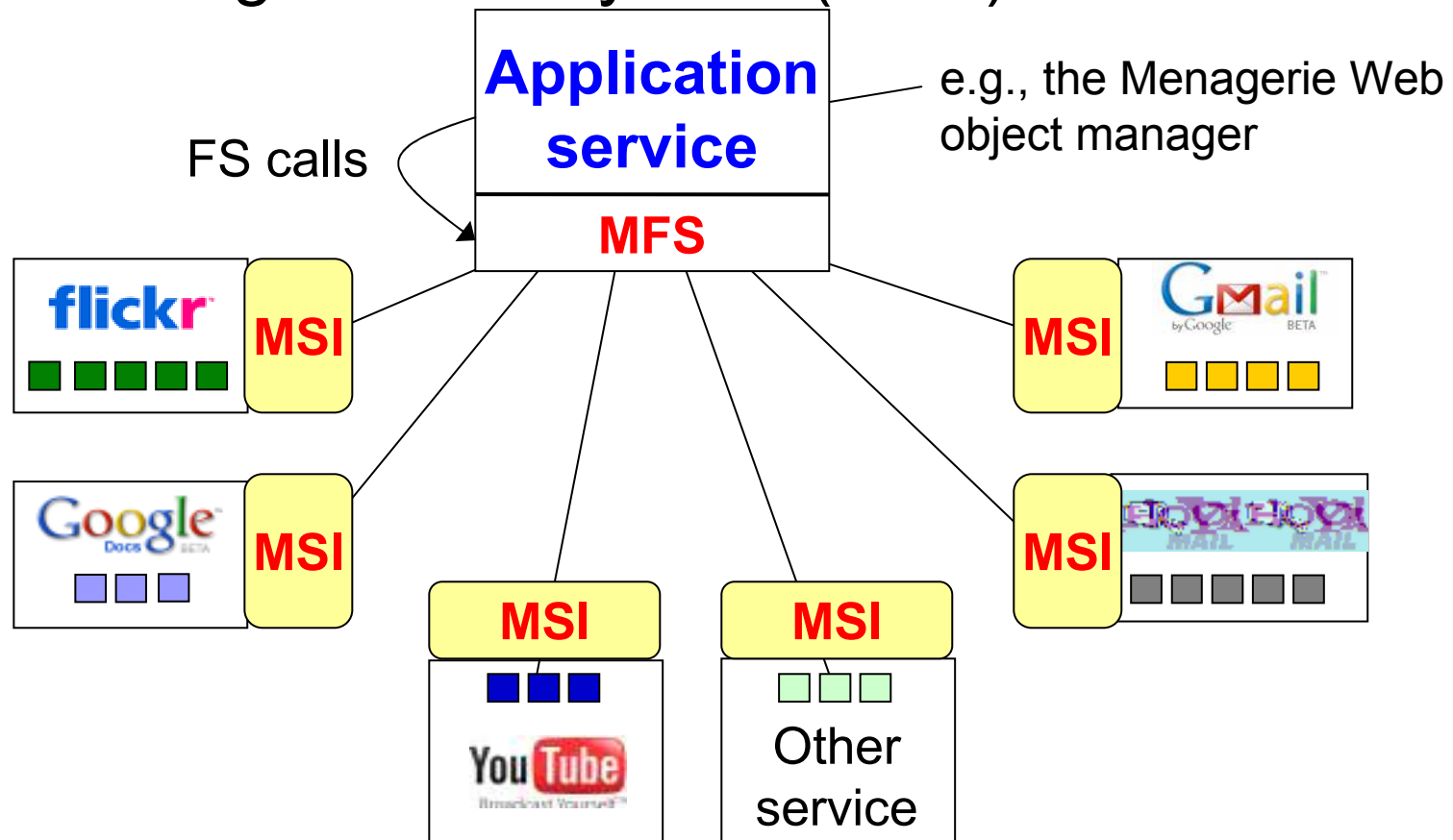
- Framework for building applications for **personal Web-data management and sharing**
- Provides a set of unified infrastructure functions
  - On the desktop, the **FS** provided these functions

### 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

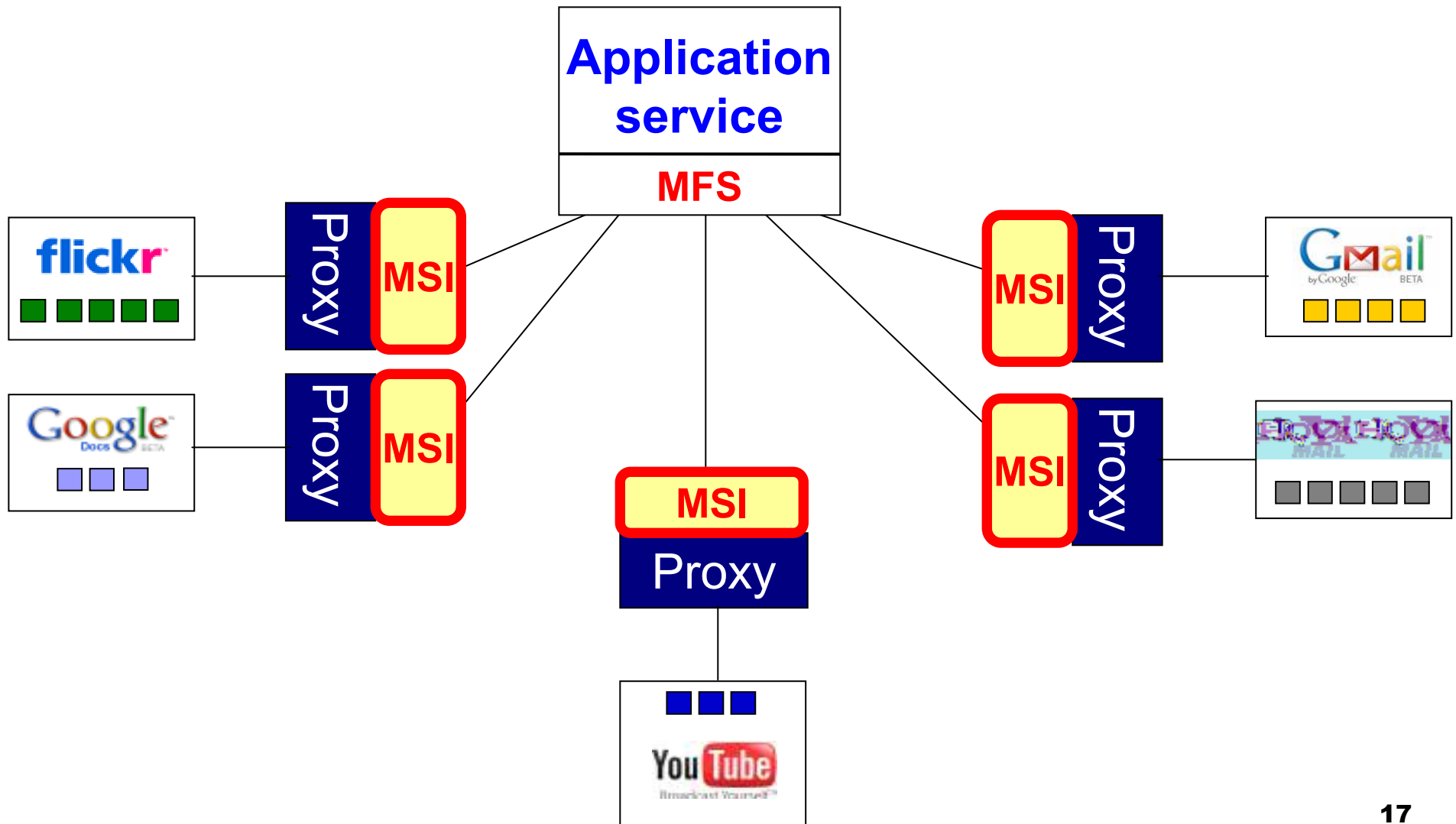
# Menagerie architecture

- 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)

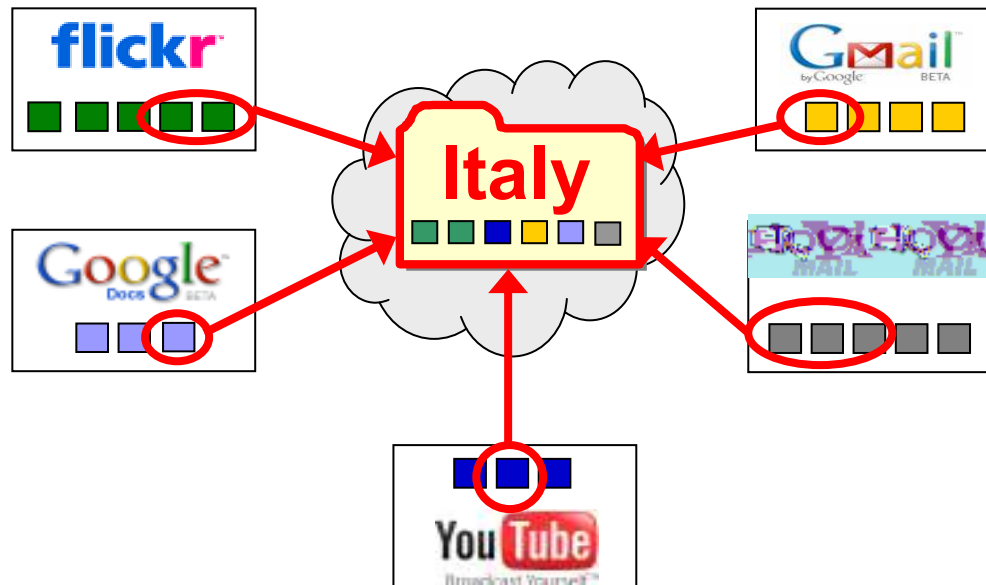
- Common service API

Contains operations for:

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

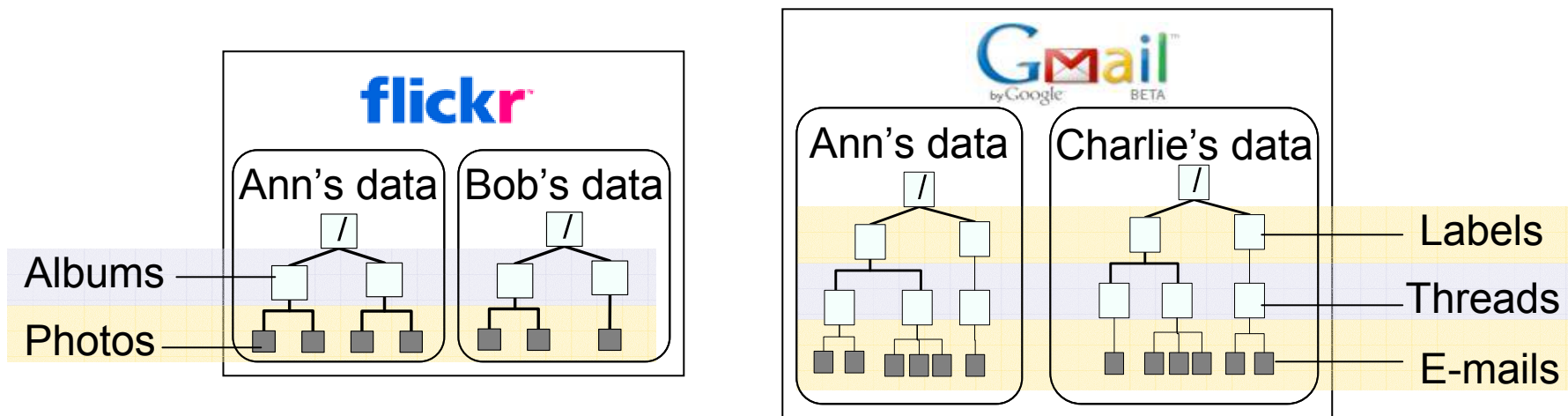
# 1. Common object naming

- Supports creation of heterogeneous folders



# 1. Common object naming

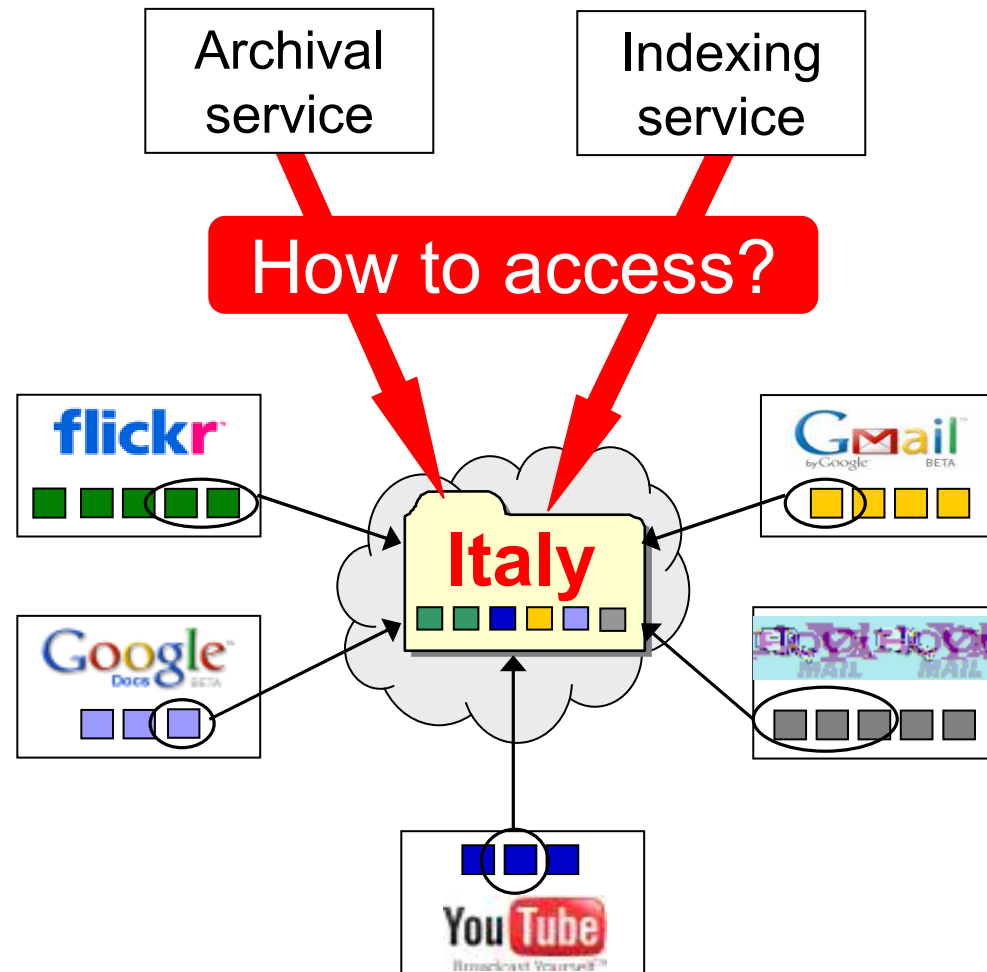
- Supports creation of heterogeneous folders
- Each service **exports** a hierarchical namespace of each user's objects



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

## 2. Common access to object content

- Supports processing of objects and collections



## 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



Email  
(HTML snippet)



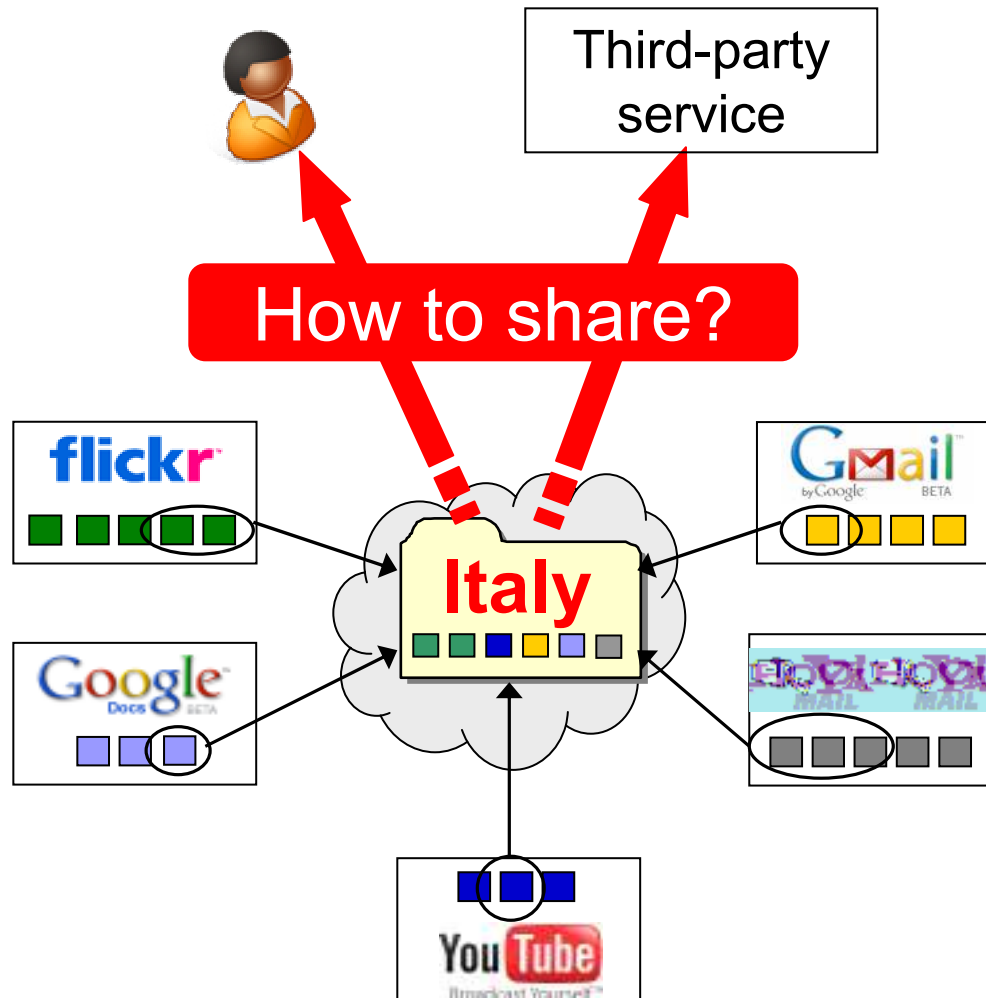
Youtube  
(`<object>` tag)



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

- Managerie 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 <i>any</i> type with a group (uses Gallery)	167 (php)
Web-data backup	Back up heterogeneous collections of web objects (uses <b>tar</b> )	10 (bash)
Contact synchronizer	Uses <b>unison</b> 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



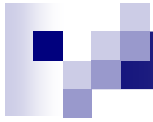
## Related work

- 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