



SEPARATING WEB APPLICATIONS FROM USER DATA STORAGE WITH BSTORE

Ramesh Chandra
Priya Gupta
Nickolai Zeldovich

MIT CSAIL

MOTIVATING EXAMPLE

Alice manages her photos online.

MOTIVATING EXAMPLE

Alice manages her photos online.

- Snaps family photos
- Uploads to Flickr

MOTIVATING EXAMPLE

Alice manages her photos online.

- Snaps family photos
- Uploads to Flickr
- Edits them with PhotoEditor
 - PhotoEditor uses SuperRedEyeRemover

MOTIVATING EXAMPLE

Alice manages her photos online.

- Snaps family photos
- Uploads to Flickr
- Edits them with PhotoEditor
 - PhotoEditor uses SuperRedEyeRemover
- Uses PhotoViewer to view photos as wallpaper

MOTIVATING EXAMPLE

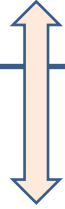
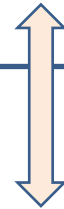
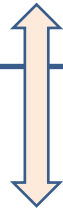
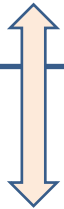
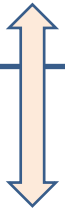
Alice manages her photos online.

- Snaps family photos
- Uploads to Flickr
- Edits them with PhotoEditor
 - PhotoEditor uses SuperRedEyeRemover
- Uses PhotoViewer to view photos as wallpaper
- Uses Shutterfly to print photos

STRAWMAN – MANUAL MANAGEMENT



BROWSER

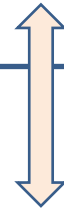
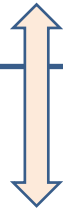
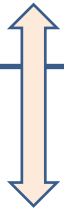
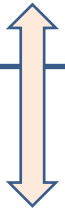


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

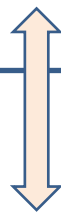
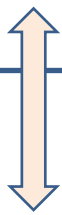
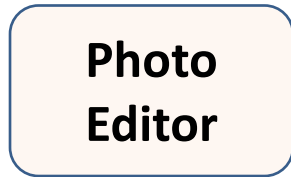


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

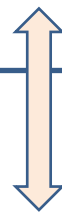
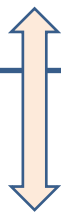
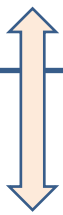


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

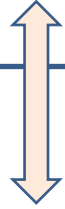
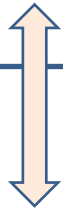
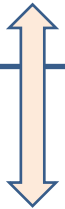
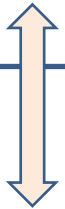


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

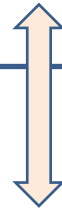
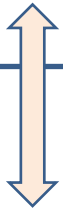


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

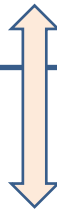


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

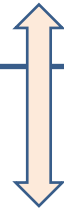
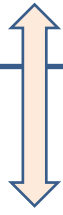
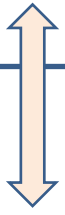


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

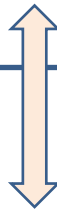


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

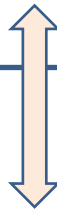


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

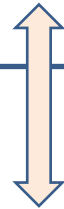
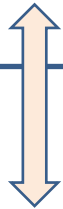
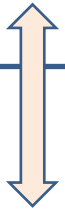
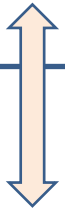


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

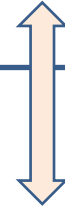
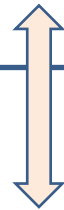
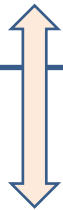
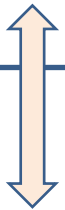


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

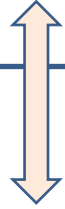
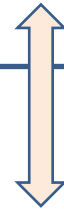
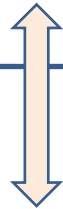
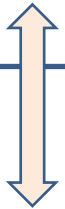


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

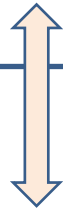
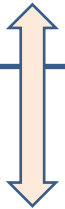


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

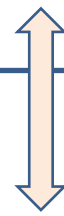
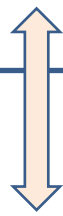


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

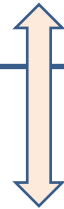
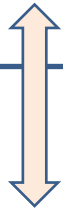
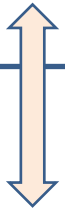
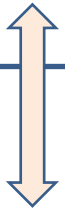


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER

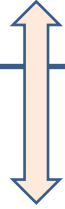
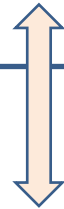
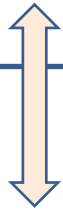
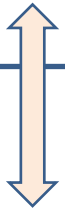
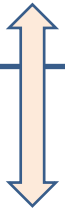


BACKEND SERVERS

STRAWMAN – MANUAL MANAGEMENT



BROWSER



BACKEND SERVERS

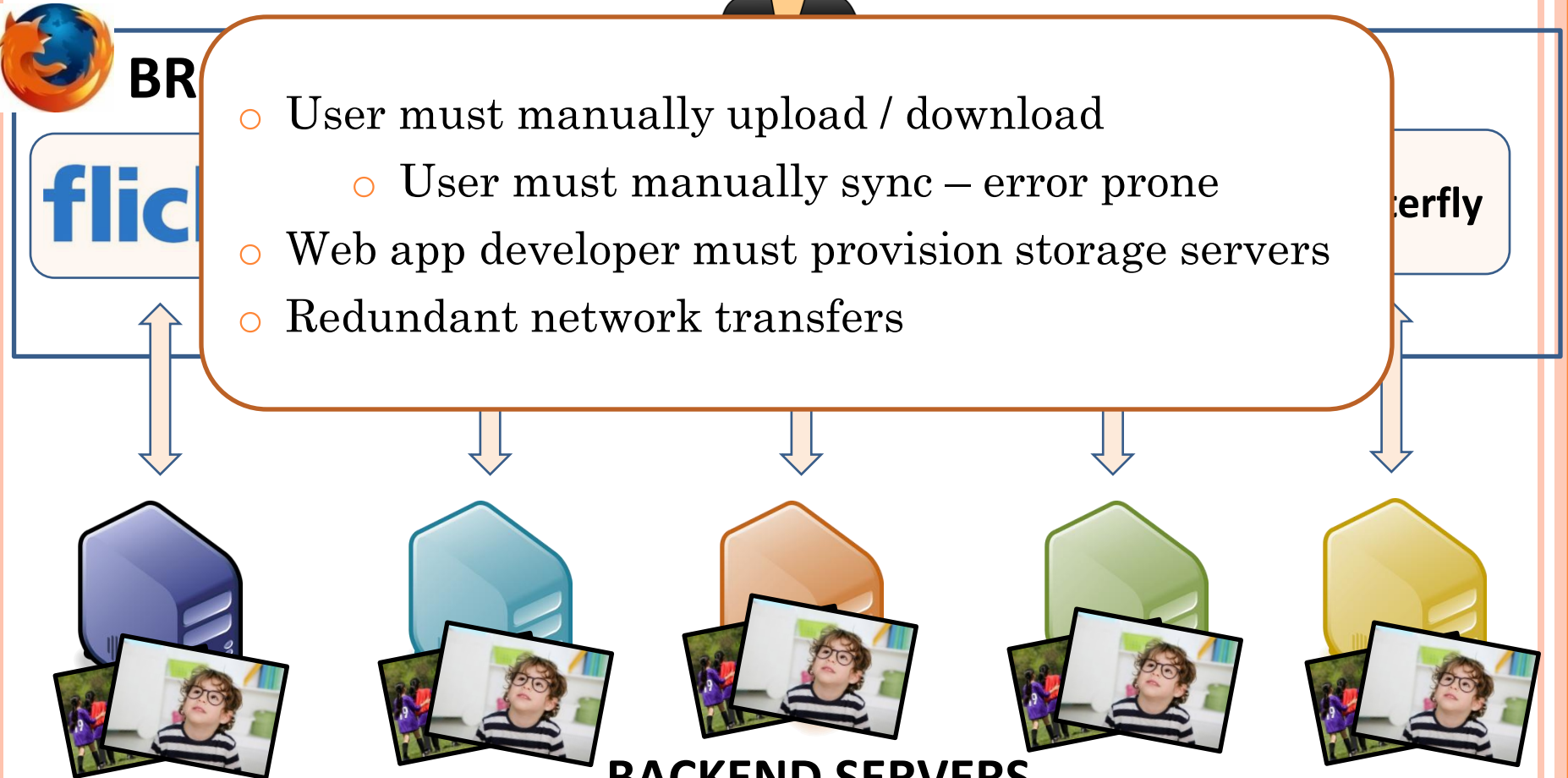
STRAWMAN – MANUAL MANAGEMENT



- User must manually upload / download
 - User must manually sync – error prone
- Web app developer must provision storage servers
- Redundant network transfers

erfly

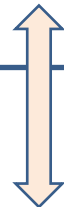
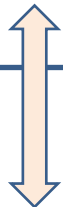
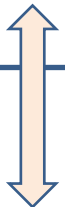
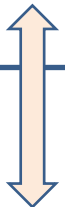
BACKEND SERVERS



CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

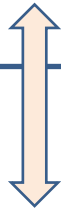
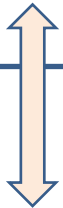
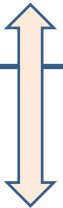


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

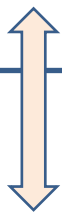


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

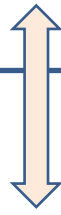
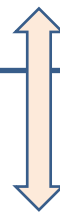
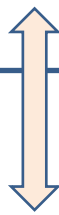
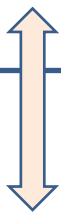


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

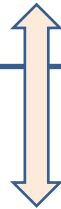
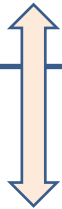
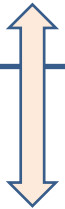


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

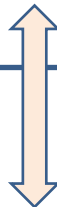
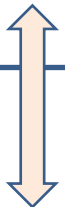


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

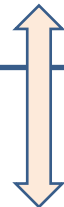
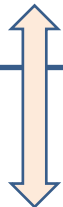
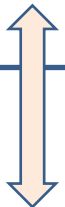


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

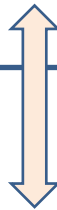
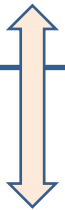


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

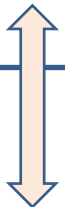


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER

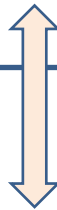
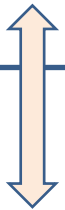


BACKEND SERVERS

CURRENT SOLN – WEB SERVICES AND OAUTH



BROWSER



BACKEND SERVERS

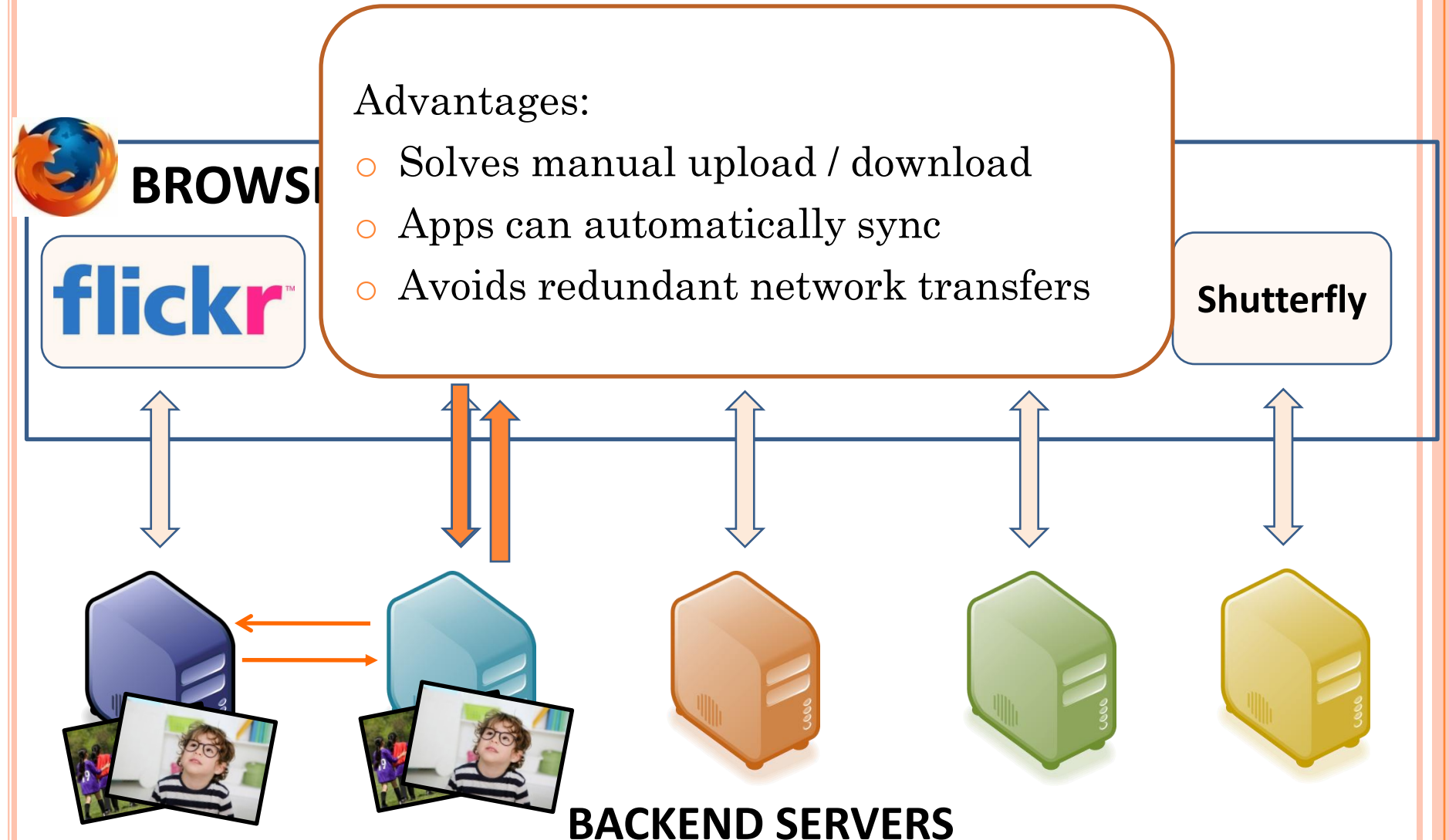
CURRENT SOLN – WEB SERVICES AND OAUTH

Advantages:

- Solves manual upload / download
- Apps can automatically sync
- Avoids redundant network transfers

Shutterfly

BACKEND SERVERS



CURRENT SOLN – WEB SERVICES AND OAUTH

Advantages:

- Solves manual upload / download
- Apps can automatically sync
- Avoids redundant network transfers

Problems:

- Needs servers to relay data
- Coarse-grained access control
- Two-party sharing
 - Apps cannot further delegate

Shutterfly

BACKEND SERVERS



flickr™



OUR SOLUTION – BSTORE

OUR SOLUTION – BSTORE

Key ideas:

- Data sharing in the browser:
 - App developers don't need servers for data access
 - Support for offline sharing
 - Avoid redundant network transfers

OUR SOLUTION – BSTORE

Key ideas:

- Data sharing in the browser:
 - App developers don't need servers for data access
 - Support for offline sharing
 - Avoid redundant network transfers
- Common storage mechanisms:
 - Unified namespace
 - Data access API
 - Delegation of access rights

BSTORE OVERVIEW

BSTORE OVERVIEW

flickr™



BACKEND SERVERS

BSTORE OVERVIEW



BROWSER



Photo Editor

SuperRedEye Remover

Photo Viewer

Shutterfly

flickr

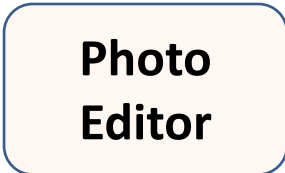


BACKEND SERVERS

BSTORE OVERVIEW

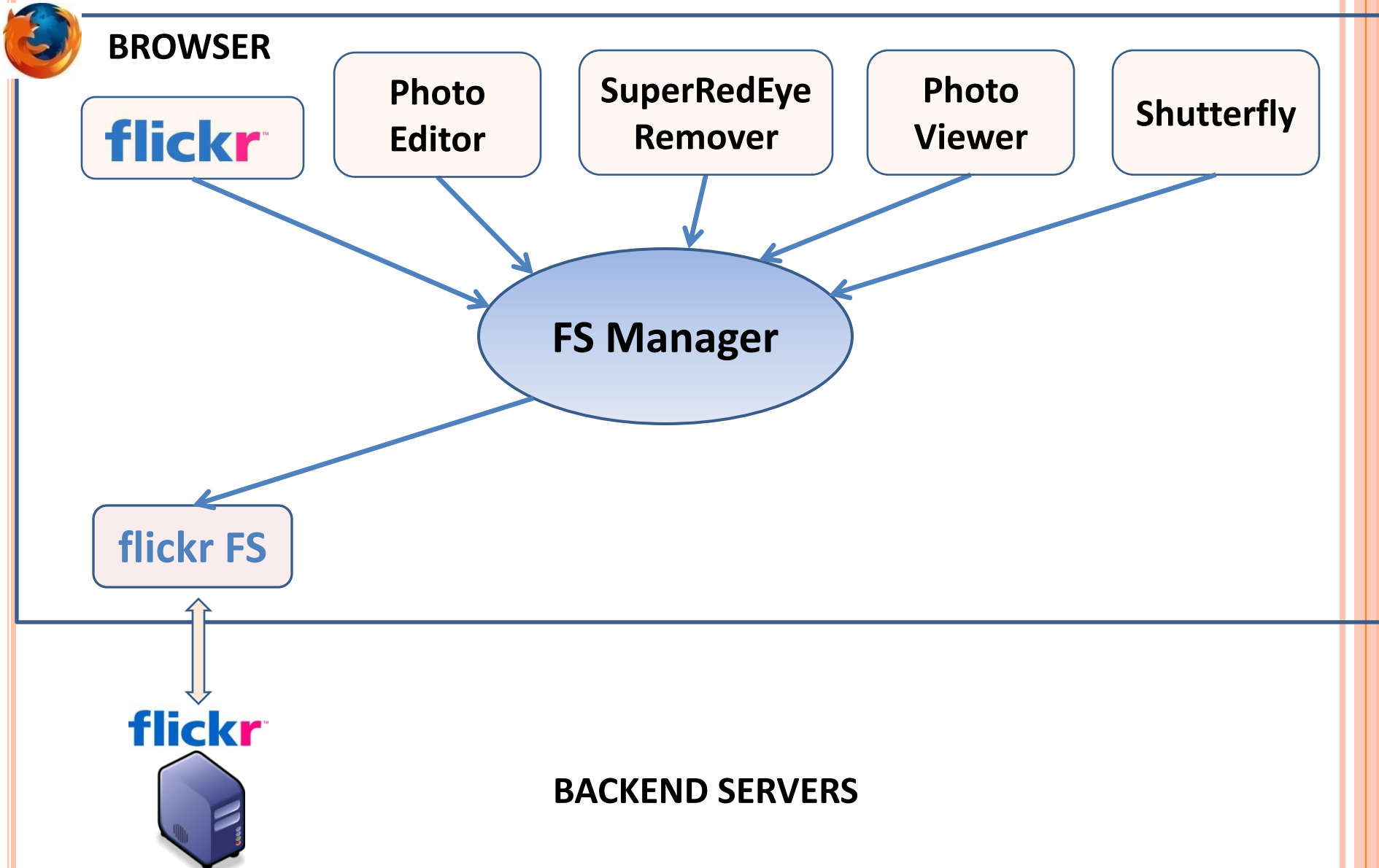


BROWSER

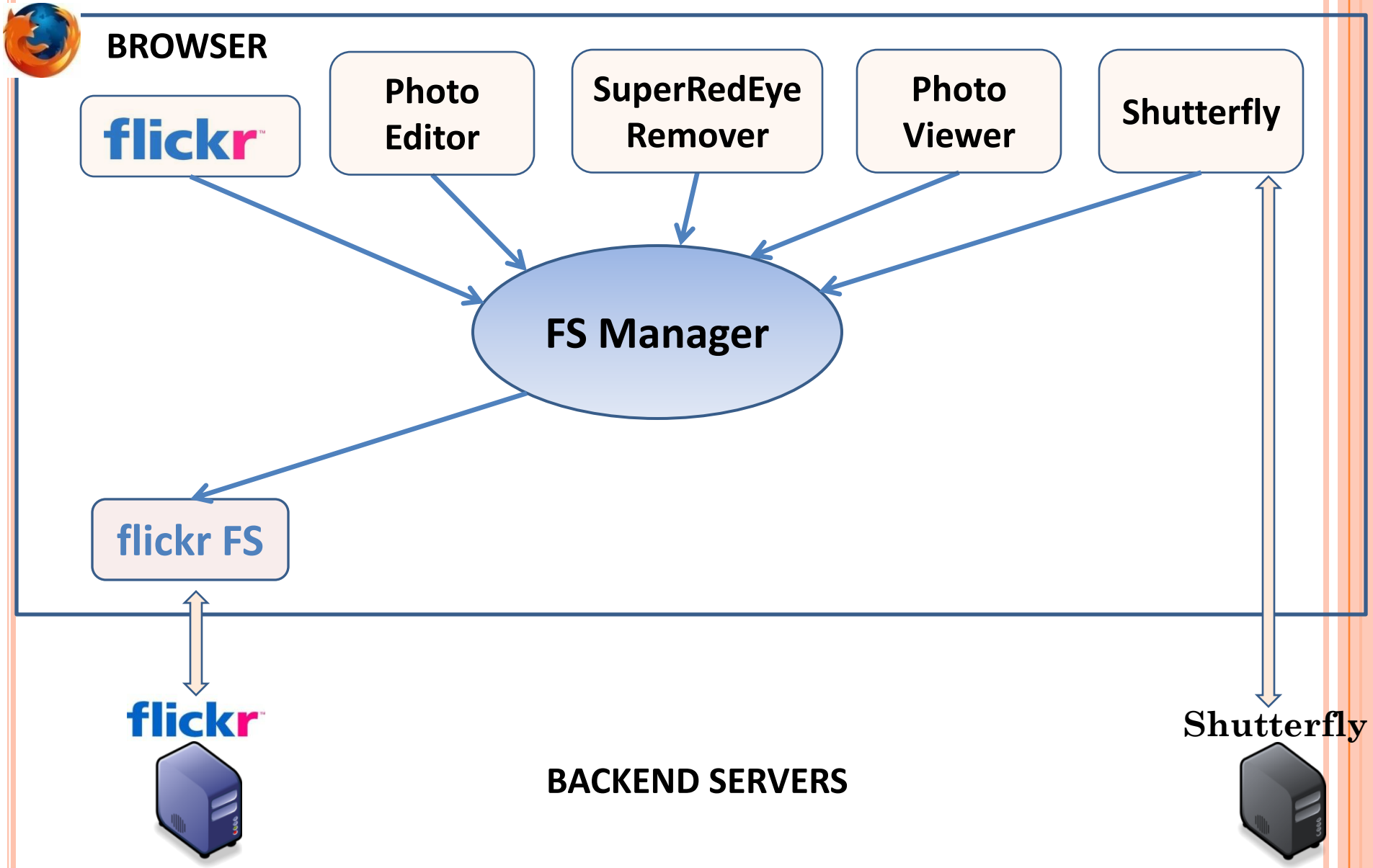


BACKEND SERVERS

BSTORE OVERVIEW



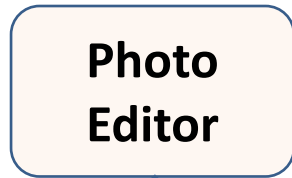
BSTORE OVERVIEW



BSTORE OVERVIEW



BROWSER

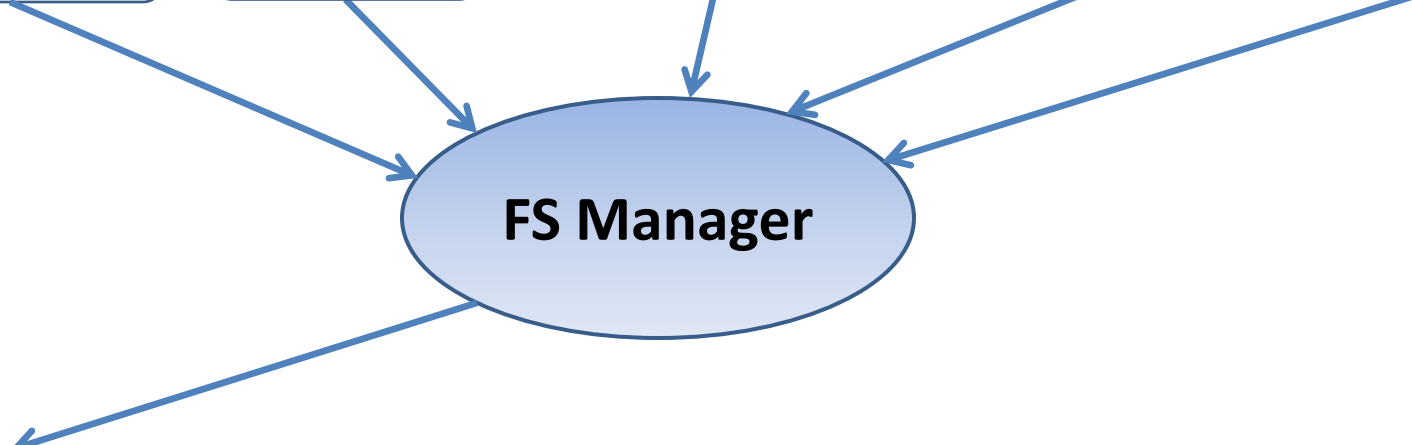
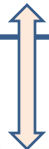


flickr



BACKEND SERVERS

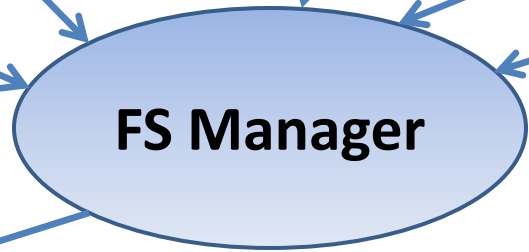
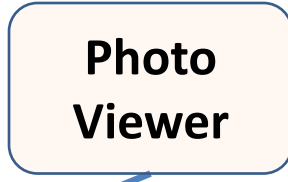
Shutterfly



BSTORE OVERVIEW



BROWSER

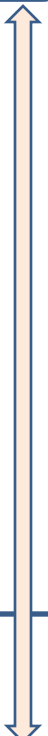
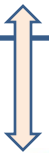
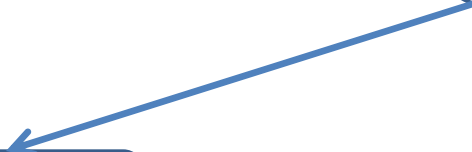
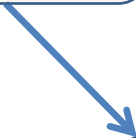
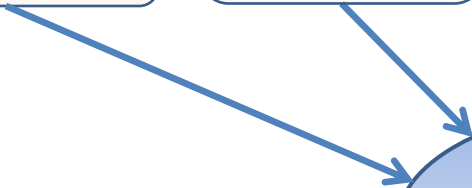
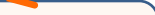
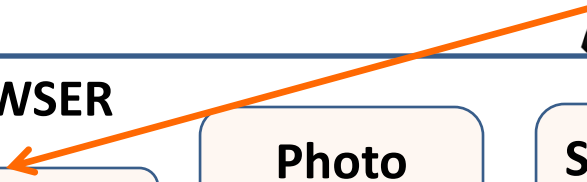


flickr



BACKEND SERVERS

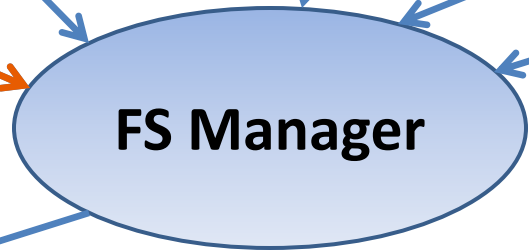
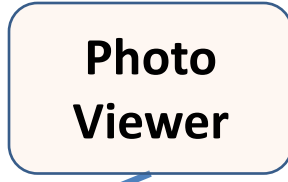
Shutterfly



BSTORE OVERVIEW



BROWSER

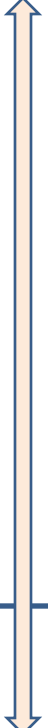
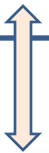
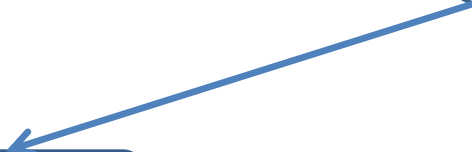
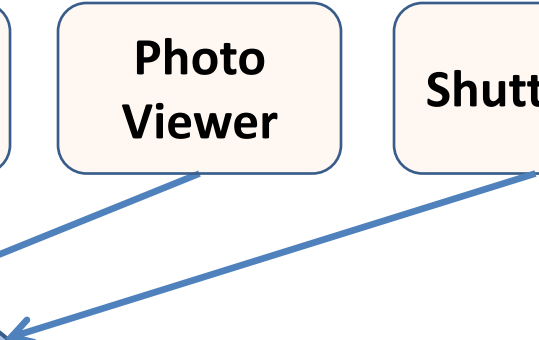
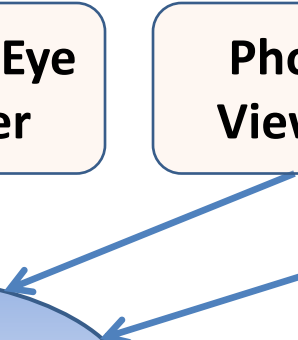


flickr



BACKEND SERVERS

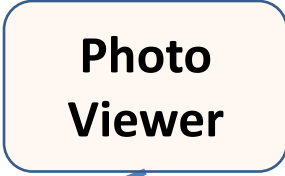
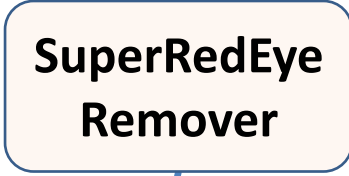
Shutterfly



BSTORE OVERVIEW

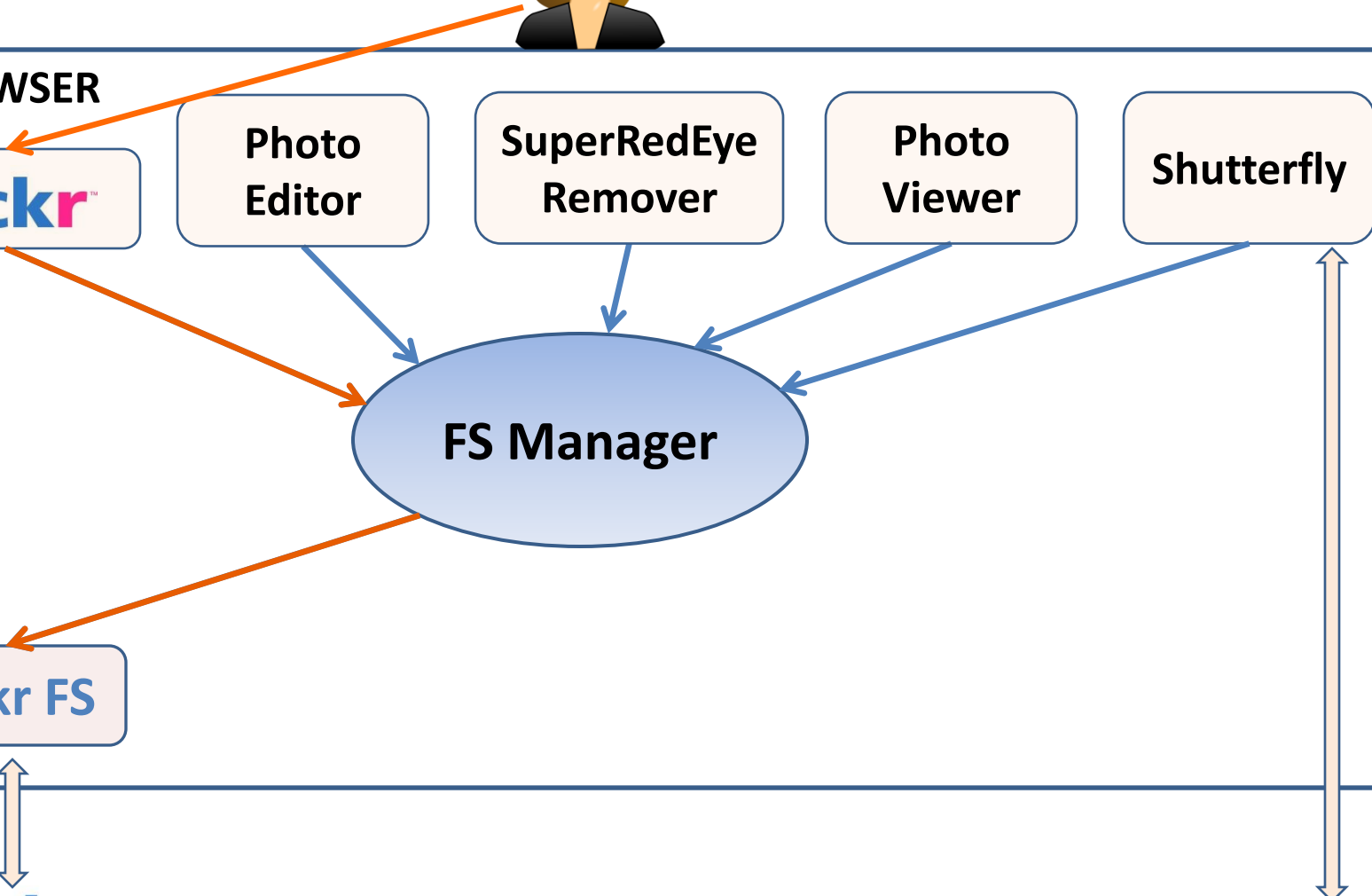


BROWSER



BACKEND SERVERS

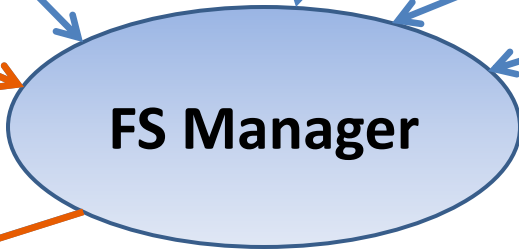
Shutterfly



BSTORE OVERVIEW

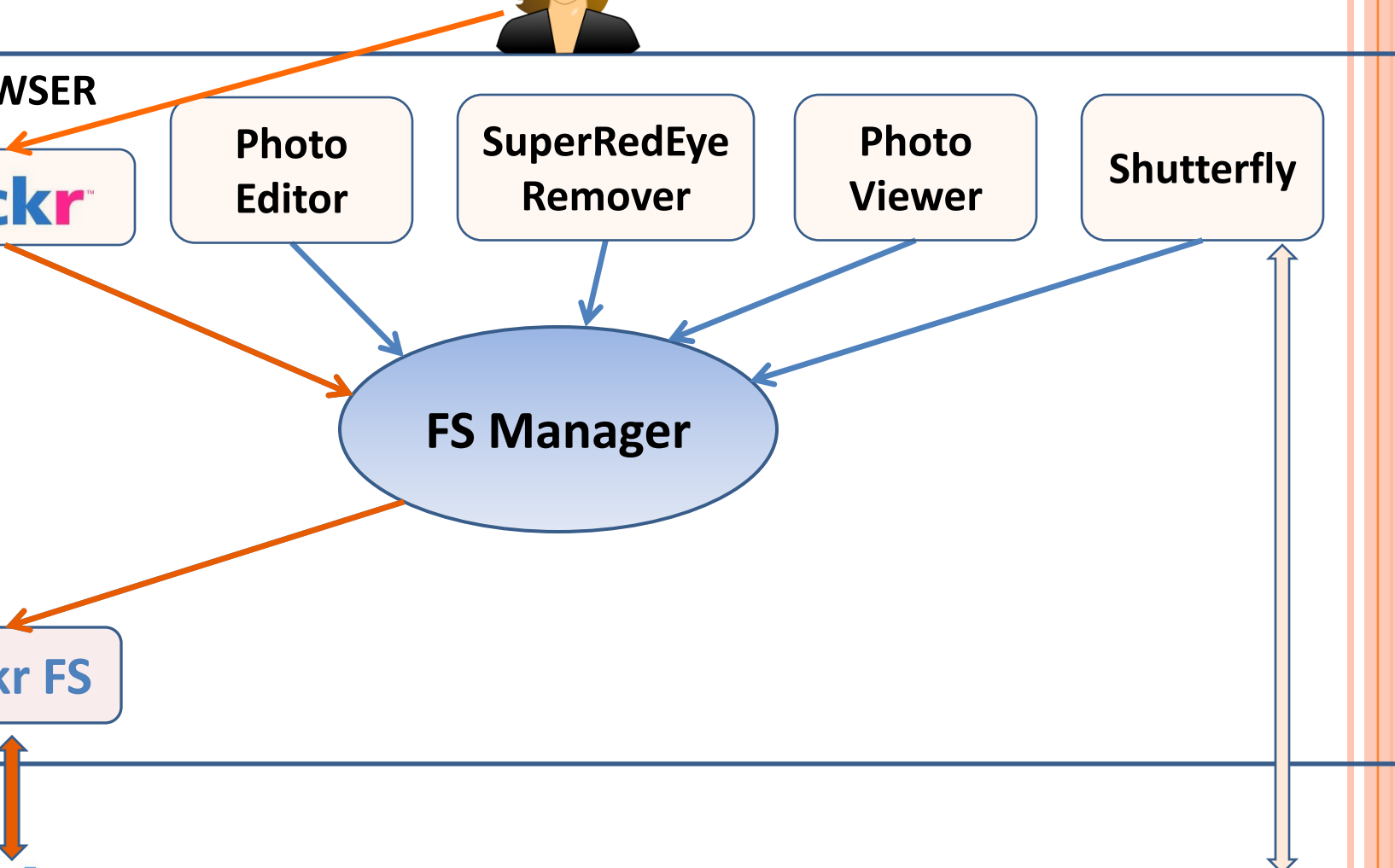


BROWSER



BACKEND SERVERS

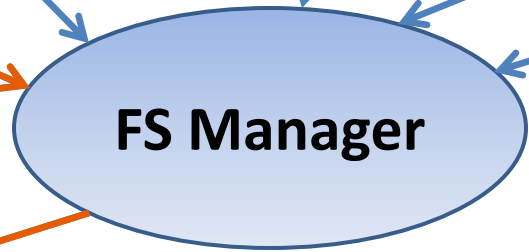
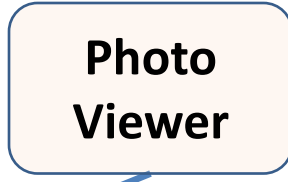
Shutterfly



BSTORE OVERVIEW



BROWSER

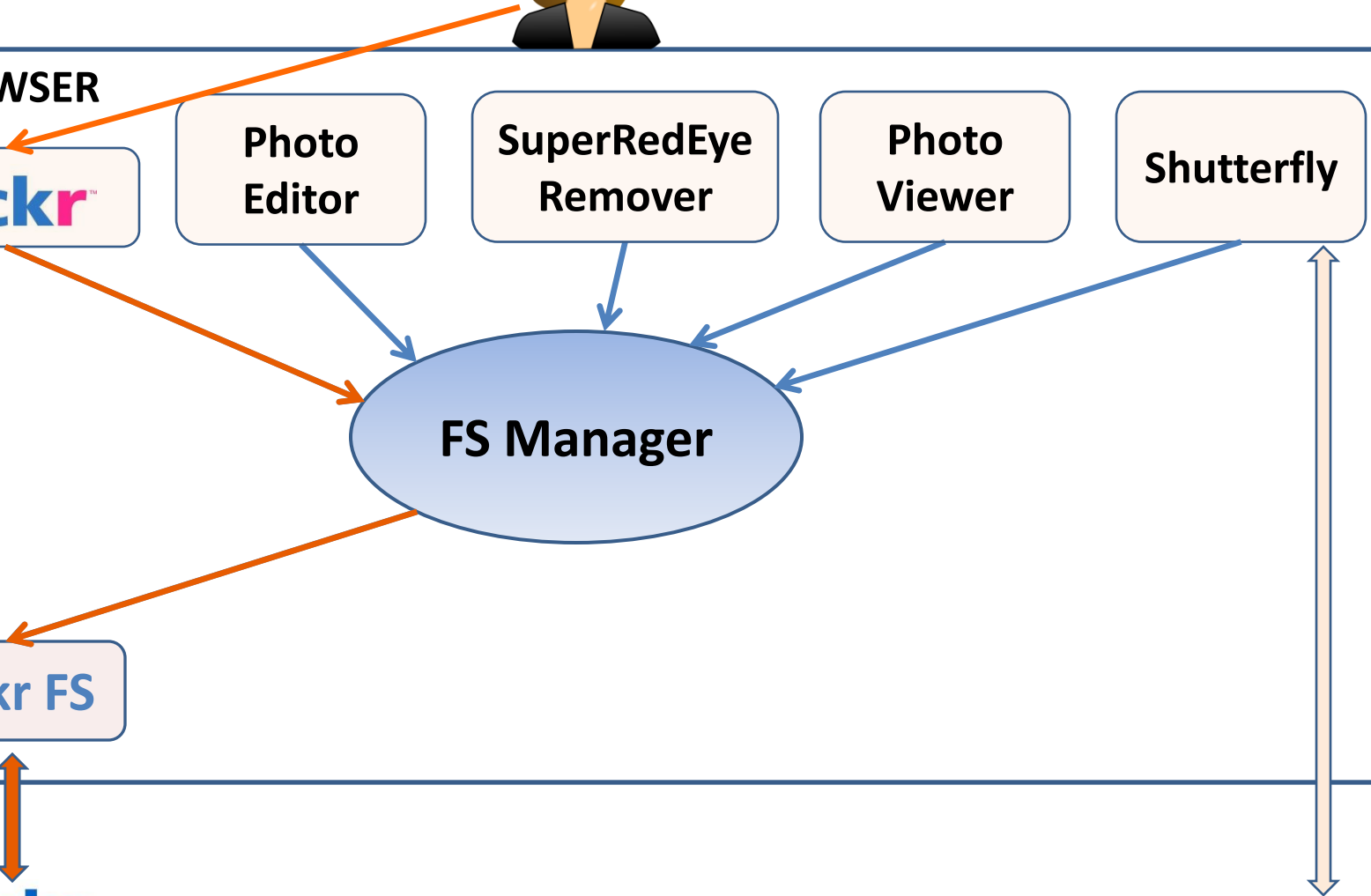


flickr



BACKEND SERVERS

Shutterfly



BSTORE OVERVIEW



BROWSER

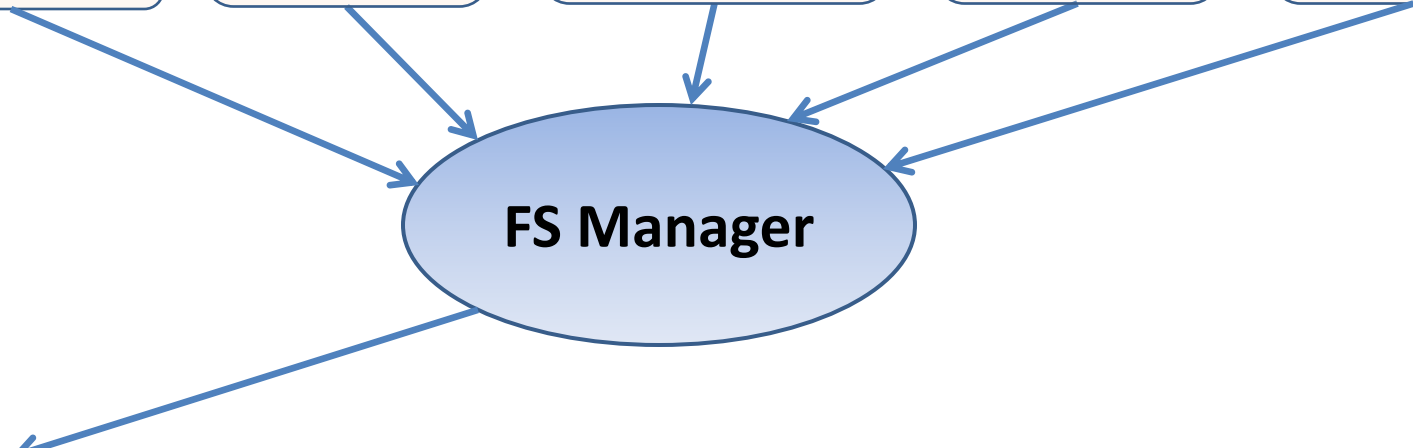
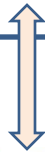


flickr



BACKEND SERVERS

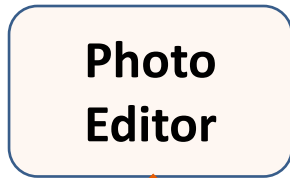
Shutterfly



BSTORE OVERVIEW



BROWSER

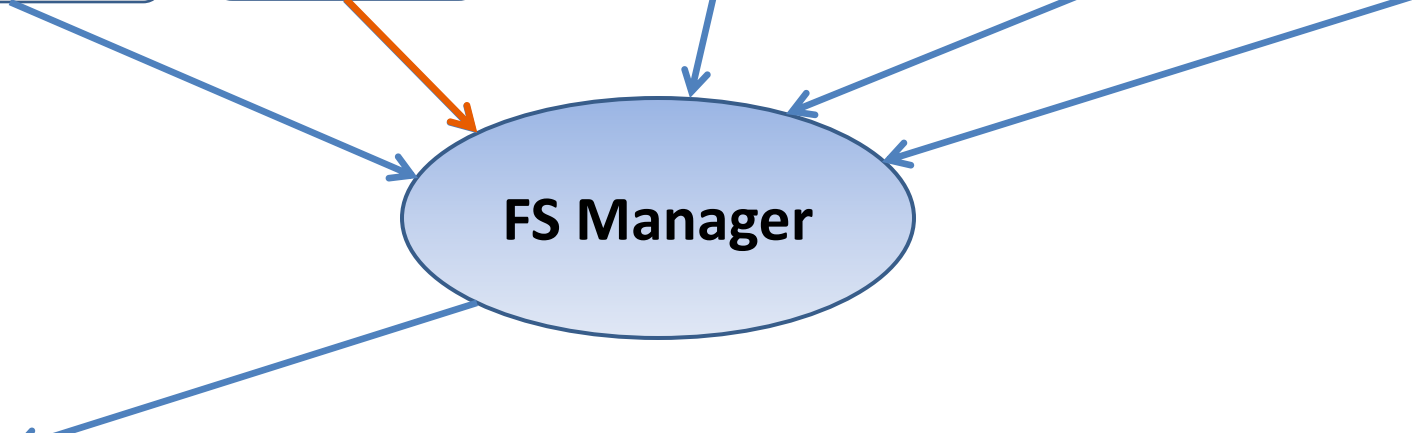
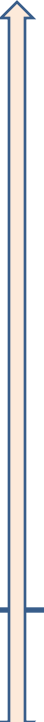


flickr



BACKEND SERVERS

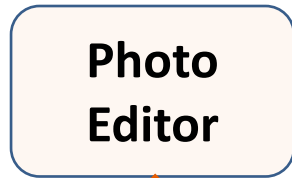
Shutterfly



BSTORE OVERVIEW



BROWSER

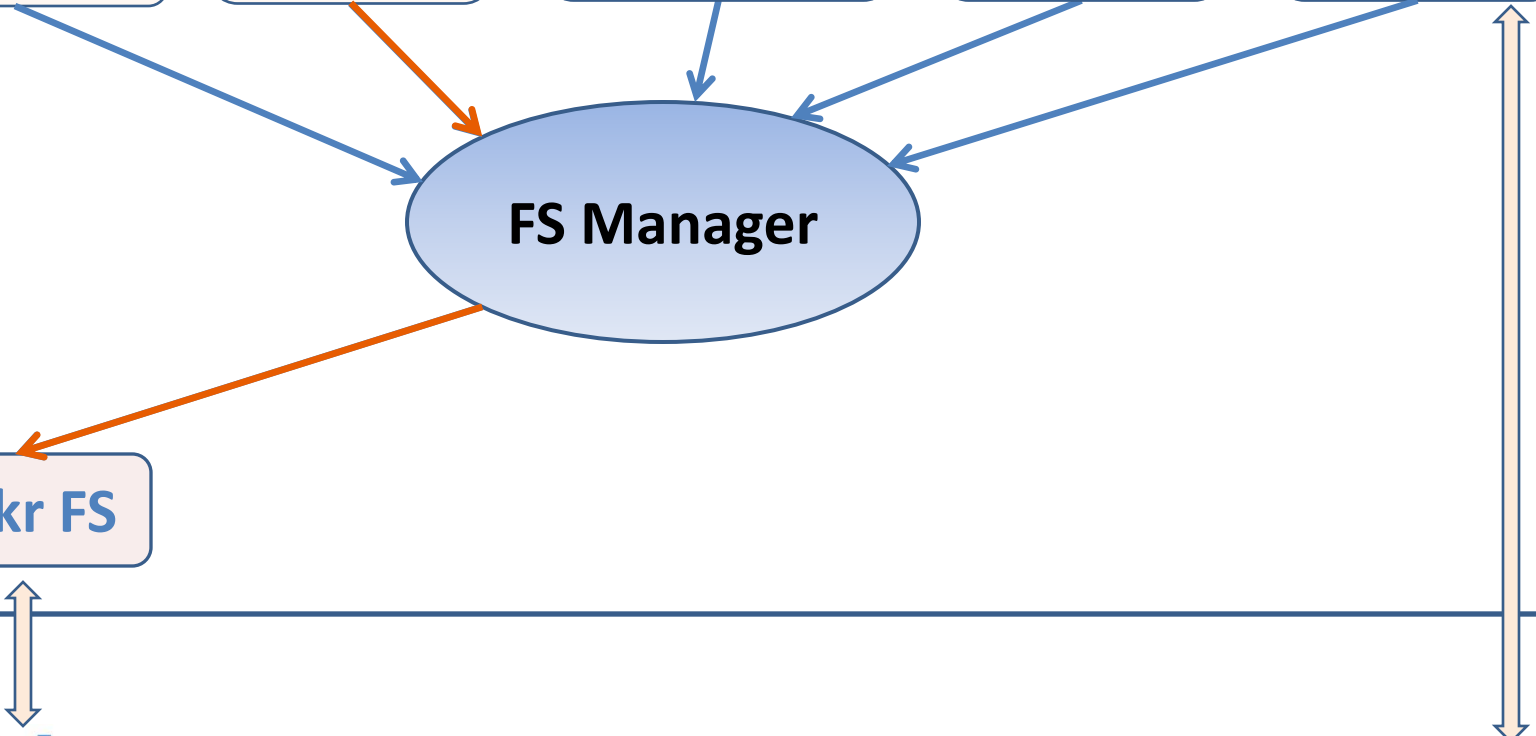


flickr



BACKEND SERVERS

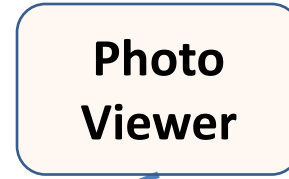
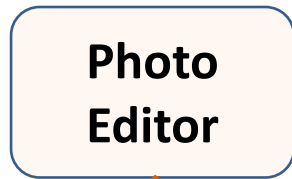
Shutterfly



BSTORE OVERVIEW



BROWSER

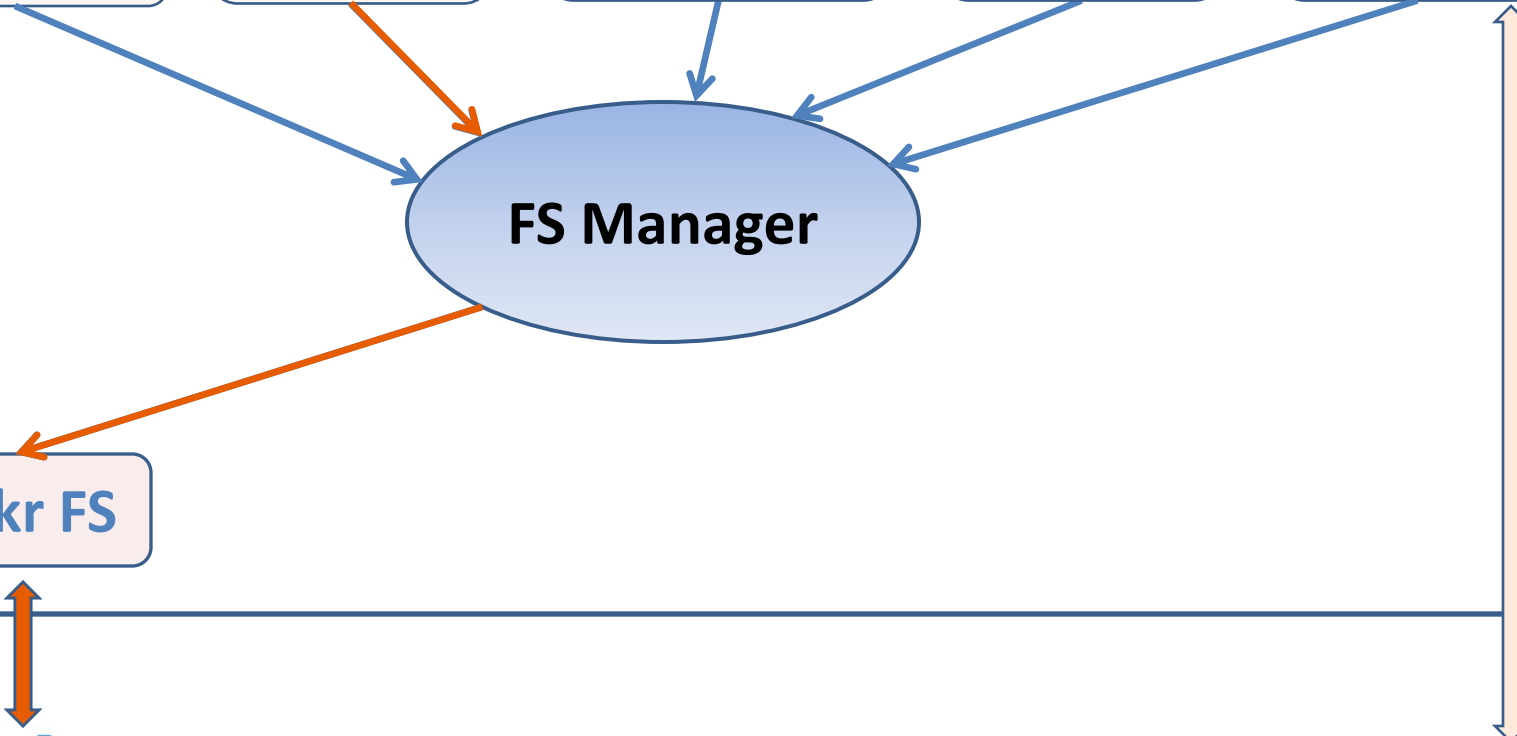


flickr

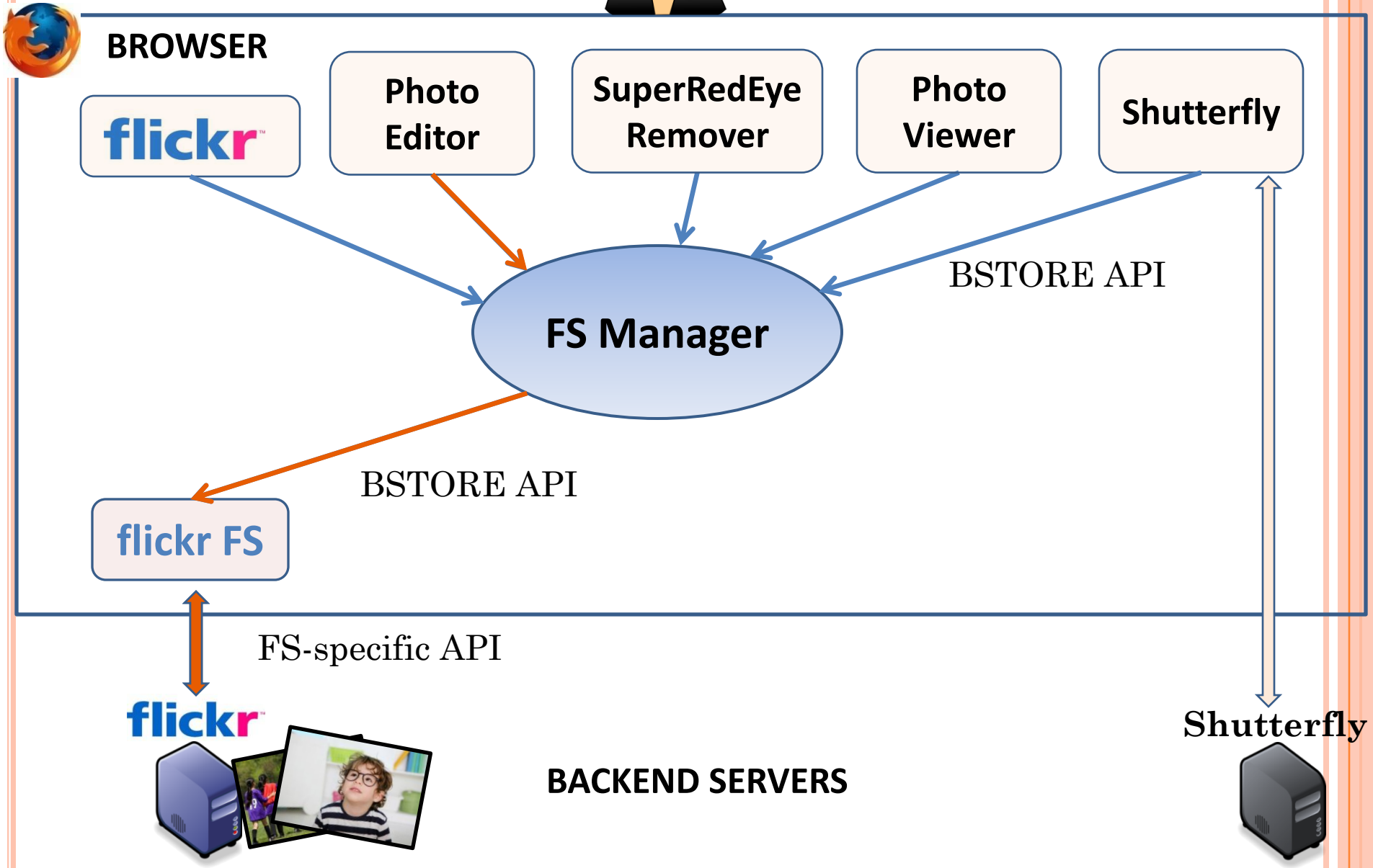


BACKEND SERVERS

Shutterfly



BSTORE OVERVIEW



BSTORE OVERVIEW



BROWSER

flickr™

Benefits to User: User-centric storage

- User's data available to all her apps
- User controls which apps get access
- Doesn't require browser changes

Shutterfly

API

FS Manager

BSTORE API

flickr FS

FS-specific API

flickr™



BACKEND SERVERS

Shutterfly



BSTORE OVERVIEW



BROWSER

flickr™

Benefits to User: User-centric storage

- User's data available to all her apps
- User controls which apps get access
- Doesn't require browser changes

Shutterfly

API

FS Manager

Benefits to Apps:

- Can share data using a common API
- Do not need server for storing or processing data
 - Enables offline sharing
- Egalitarian: Apps are first-class citizens
 - Can export existing data via BSTORE
 - Can delegate rights to other apps

flick

flic



Shutterfly



TALK OUTLINE

- Motivation
- Current solutions
- BSTORE overview
- Design details:
 - Filesystem API
 - FS organization with tags
 - Tag-based access control
 - Filesystem setup
- Implementation
- Evaluation
- Related work

FILESYSTEM API

- Each FS has a flat file structure
- BSTORE FS API:
 - `create(fs, init_tags)` -> handle
 - `set (handle, data)` ->
 - `get (handle)` -> data
 - `delete(handle)` ->
- Fits data model for most web apps

VERSIONING

- All files are versioned
- Most API calls return current version

VERSIONING

- All files are versioned
- Most API calls return current version
 - `create(fs, init_tags)` -> `ver`, handle
 - `set (handle, data)` -> `ver`
 - `get (handle)` -> `ver`, data
 - `delete(handle)` ->

VERSIONING

- All files are versioned
- Most API calls return current version
 - `create(fs, init_tags)` -> `ver, handle`
 - `set (handle, data)` -> `ver`
 - `get (handle)` -> `ver, data`
 - `delete(handle)` ->
- `set` increments version number

VERSIONING

- All files are versioned
- Most API calls return current version
 - `create(fs, init_tags)` -> `ver`, `handle`
 - `set (handle, data, [match_ver])` -> `ver`
 - `get (handle)` -> `ver`, `data`
 - `delete(handle, [match_ver])` ->
- `set` increments version number
- Compare and swap-like functionality to detect concurrent modifications
 - `set` and `delete` fail if current `ver` \neq `match_ver`

PRINCIPALS



BROWSER

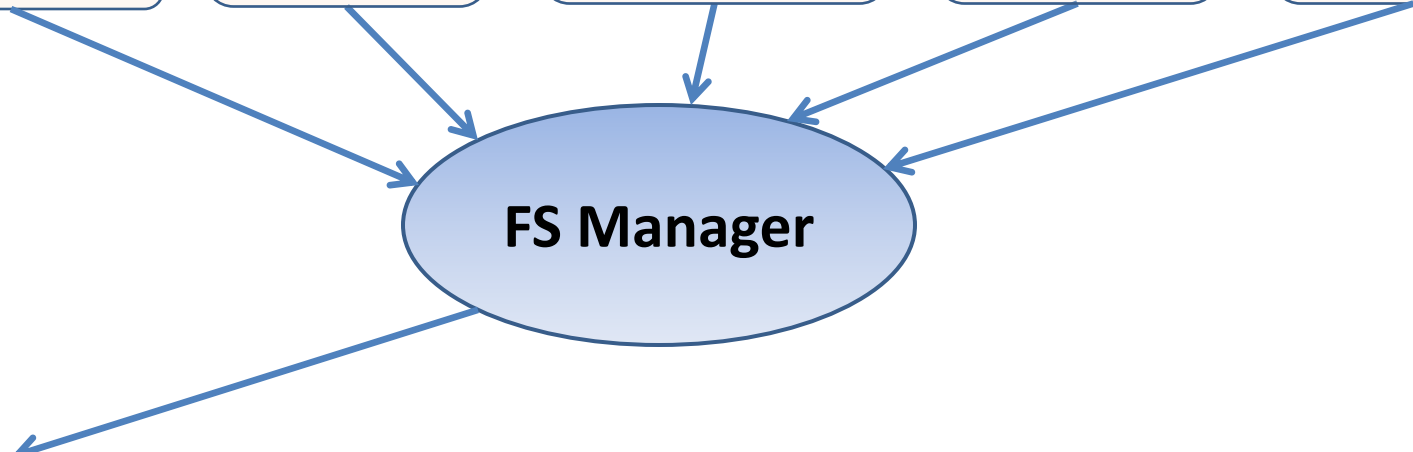
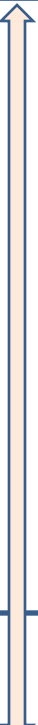
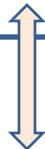


flickr



BACKEND SERVERS

Shutterfly



PRINCIPALS



BROWSER

flickr.com

editor.com

redeye.com

viewer.com

sfly.com

fsmgr.com

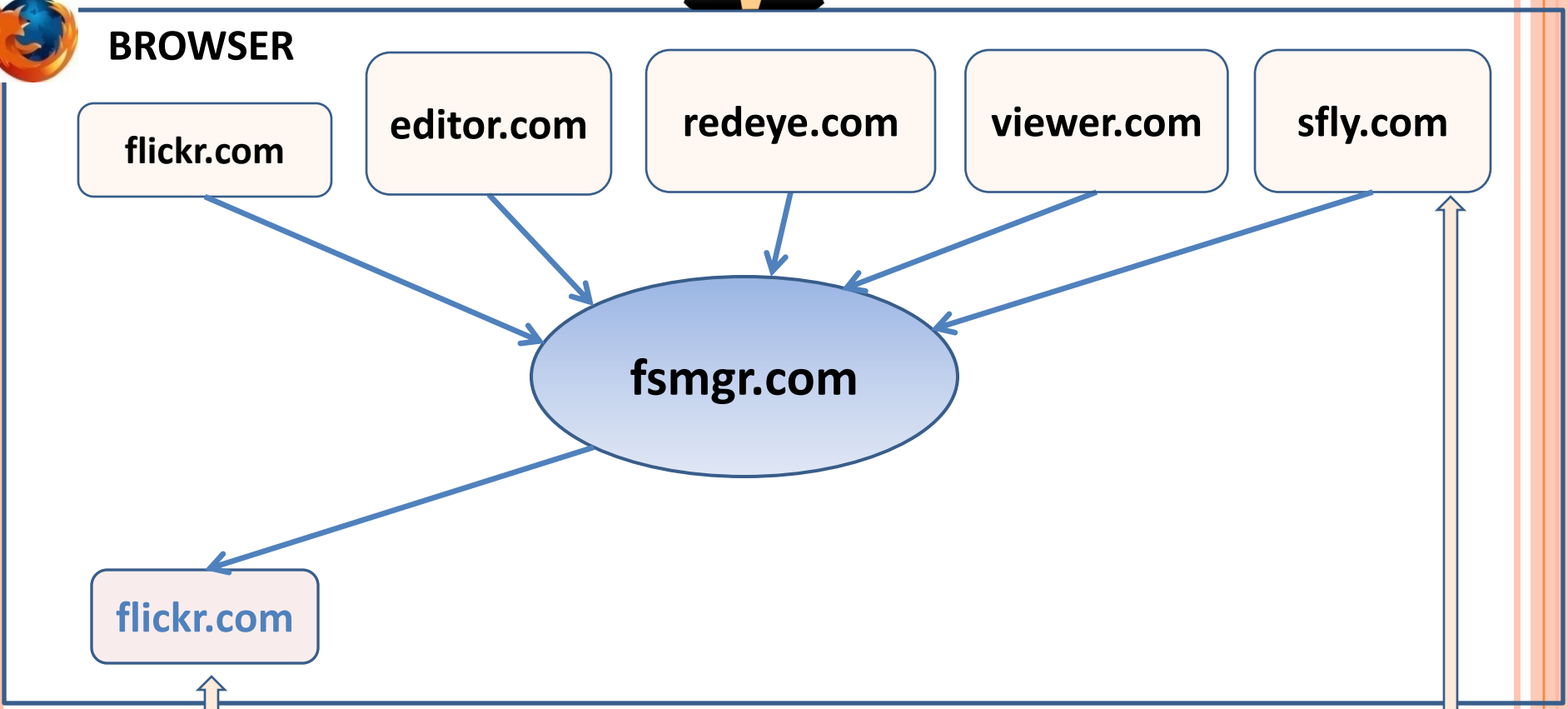
flickr.com

flickr

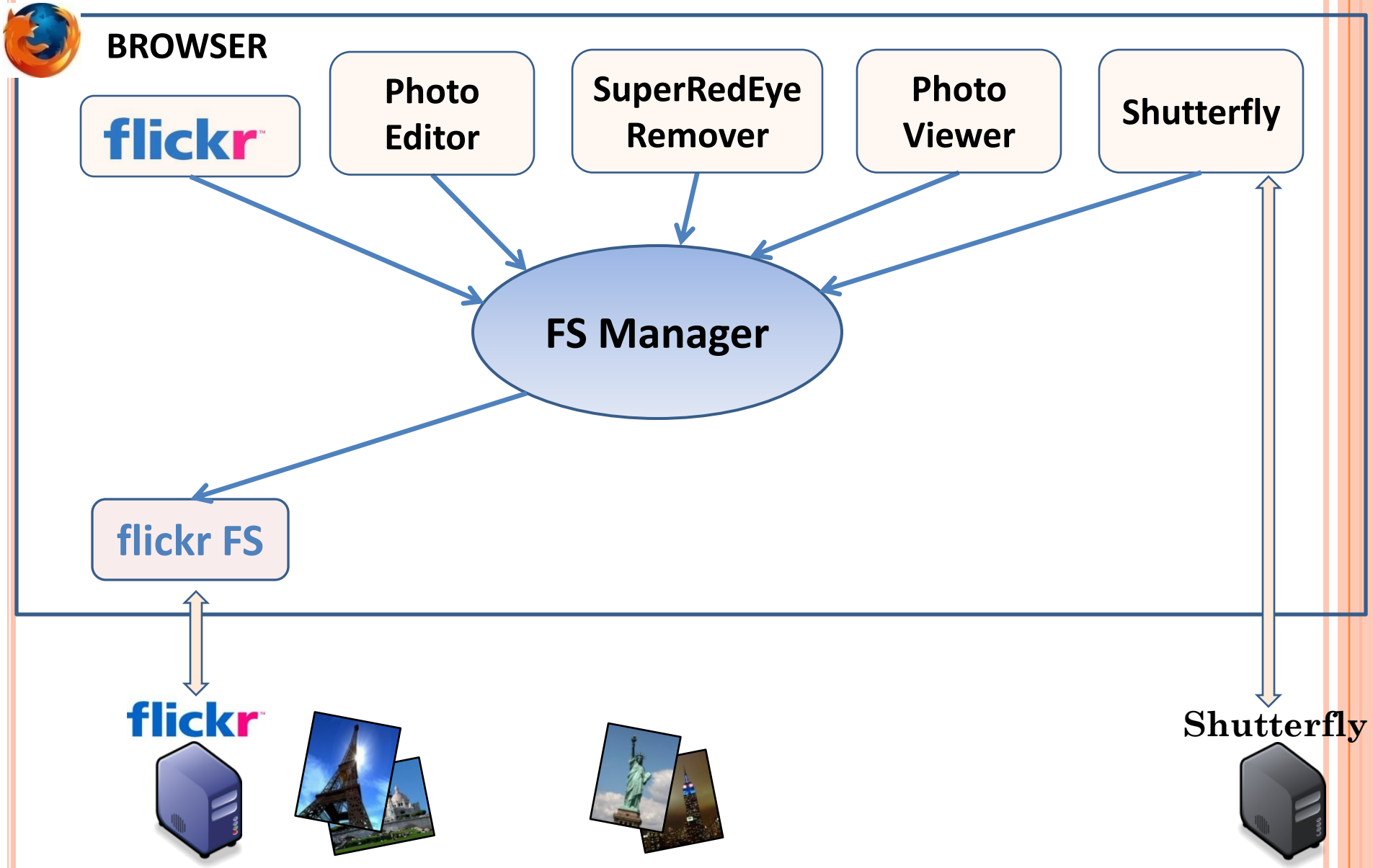


BACKEND SERVERS

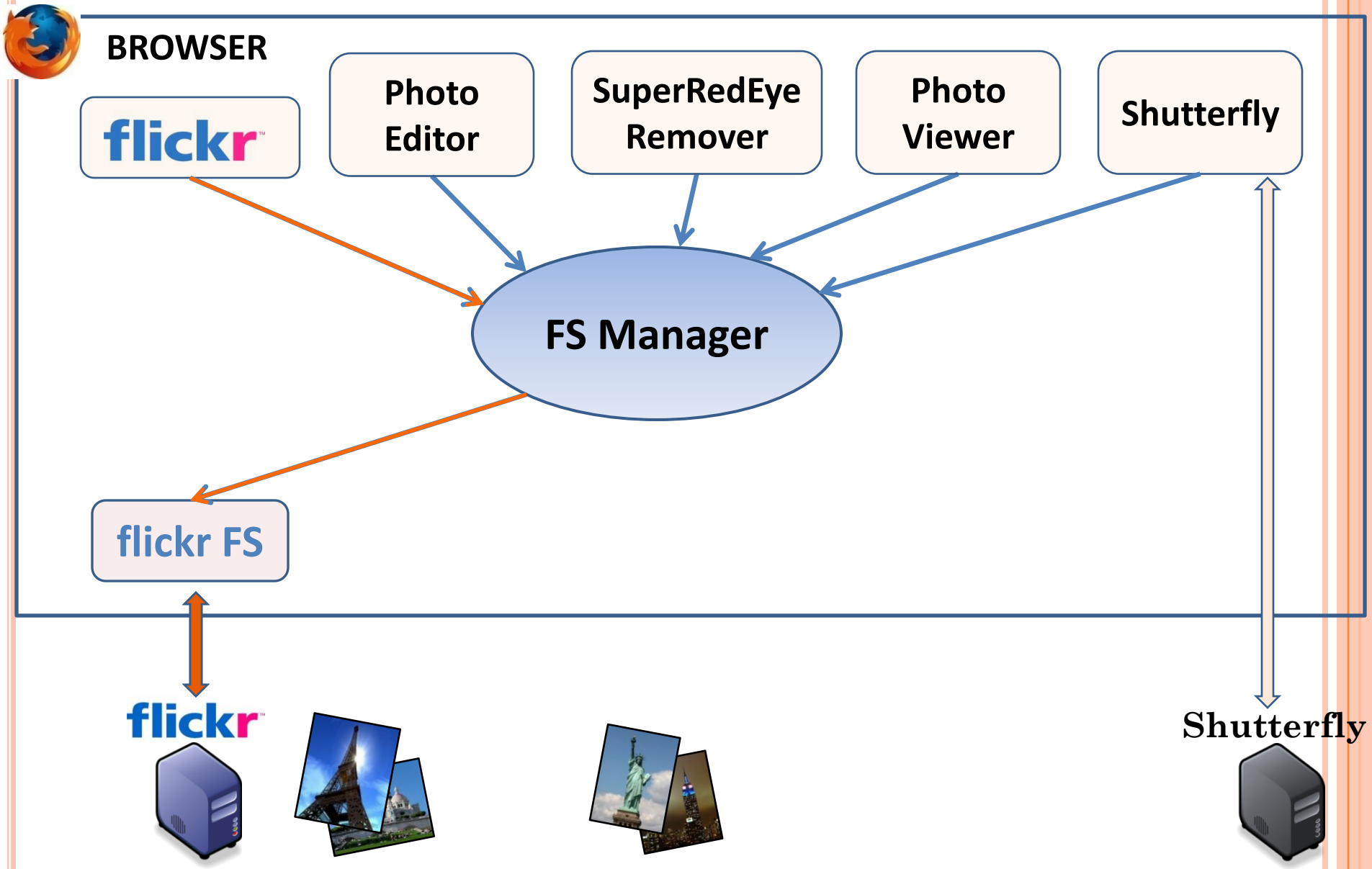
Shutterfly



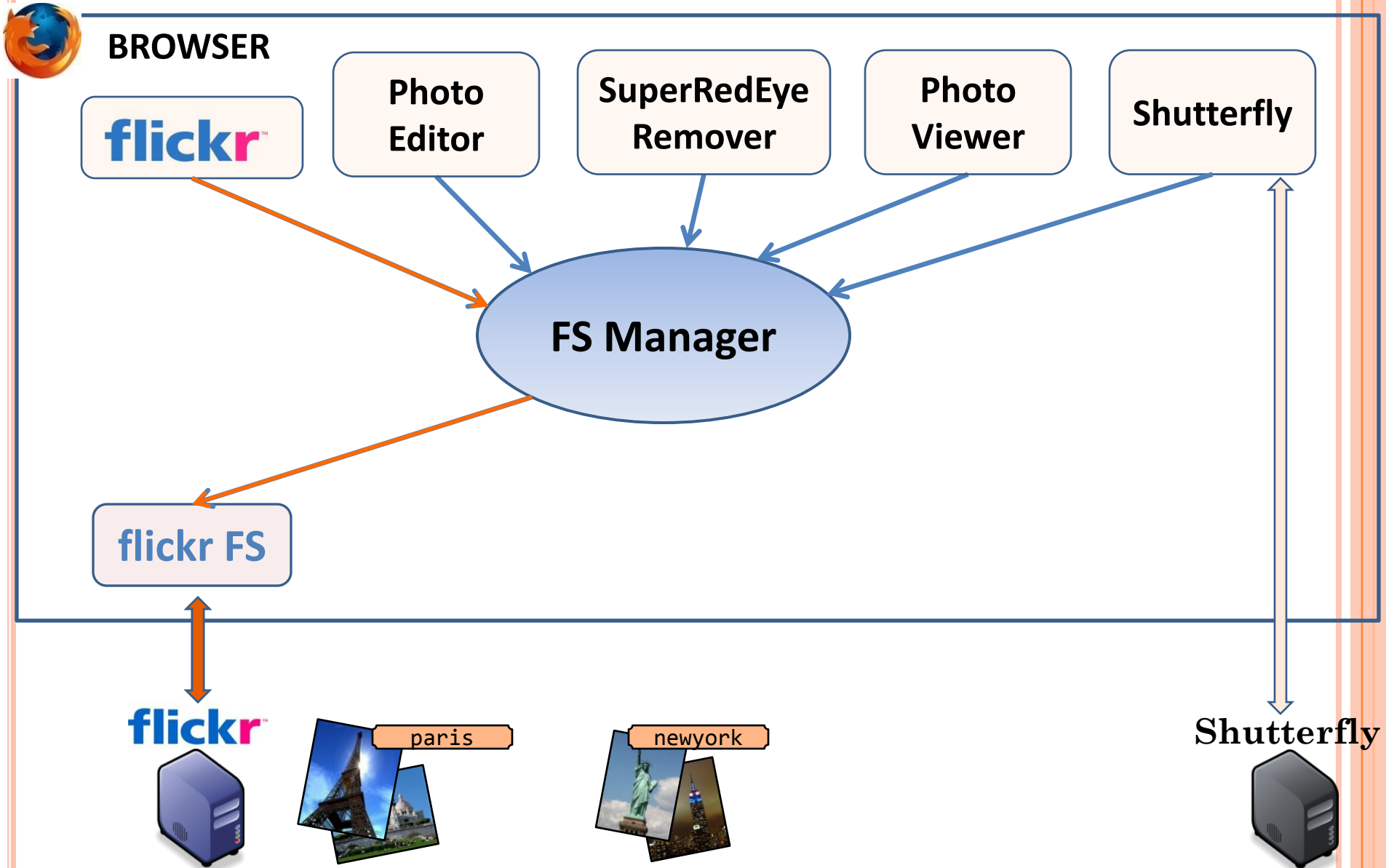
FS ORGANIZATION WITH TAGS



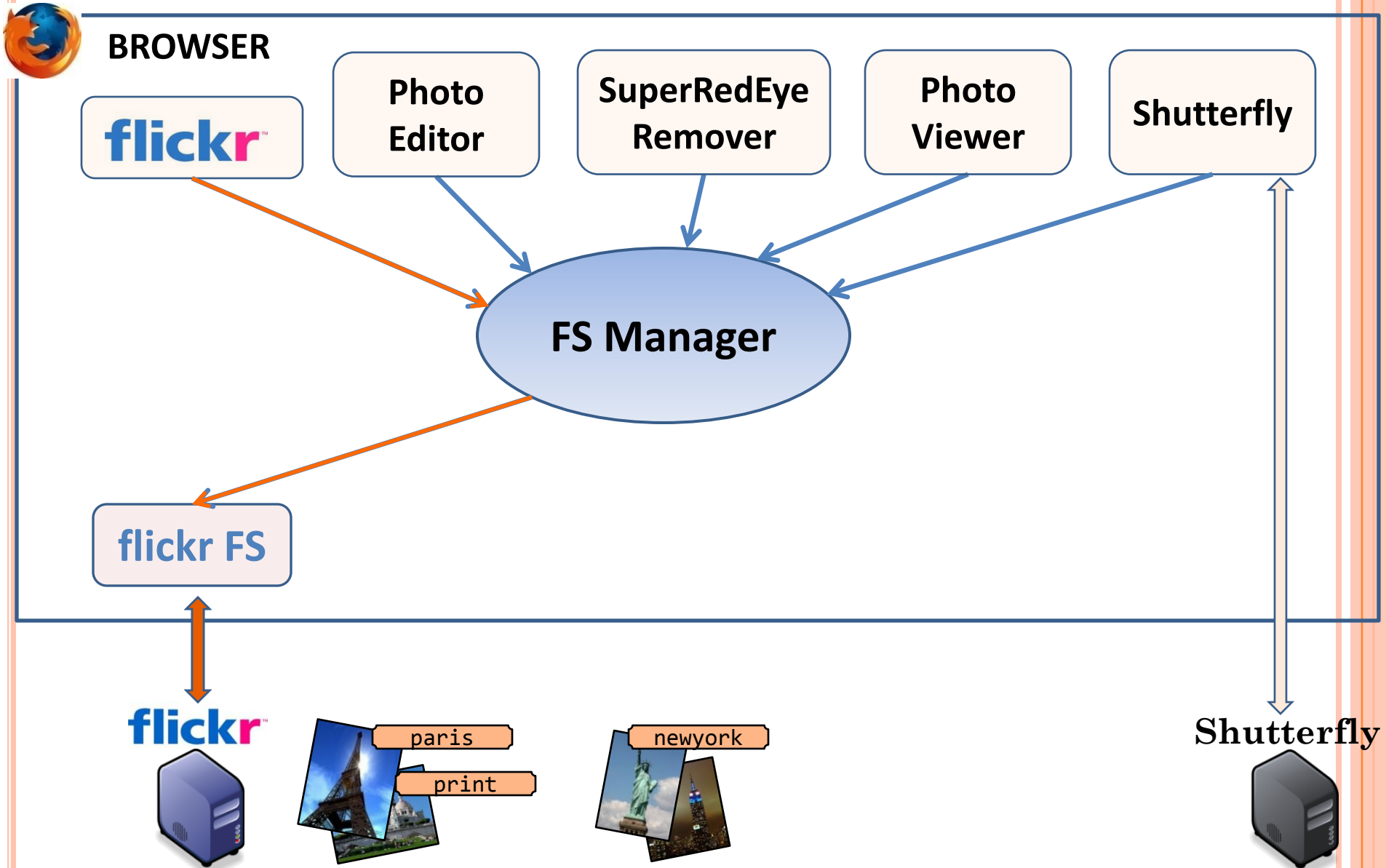
FS ORGANIZATION WITH TAGS



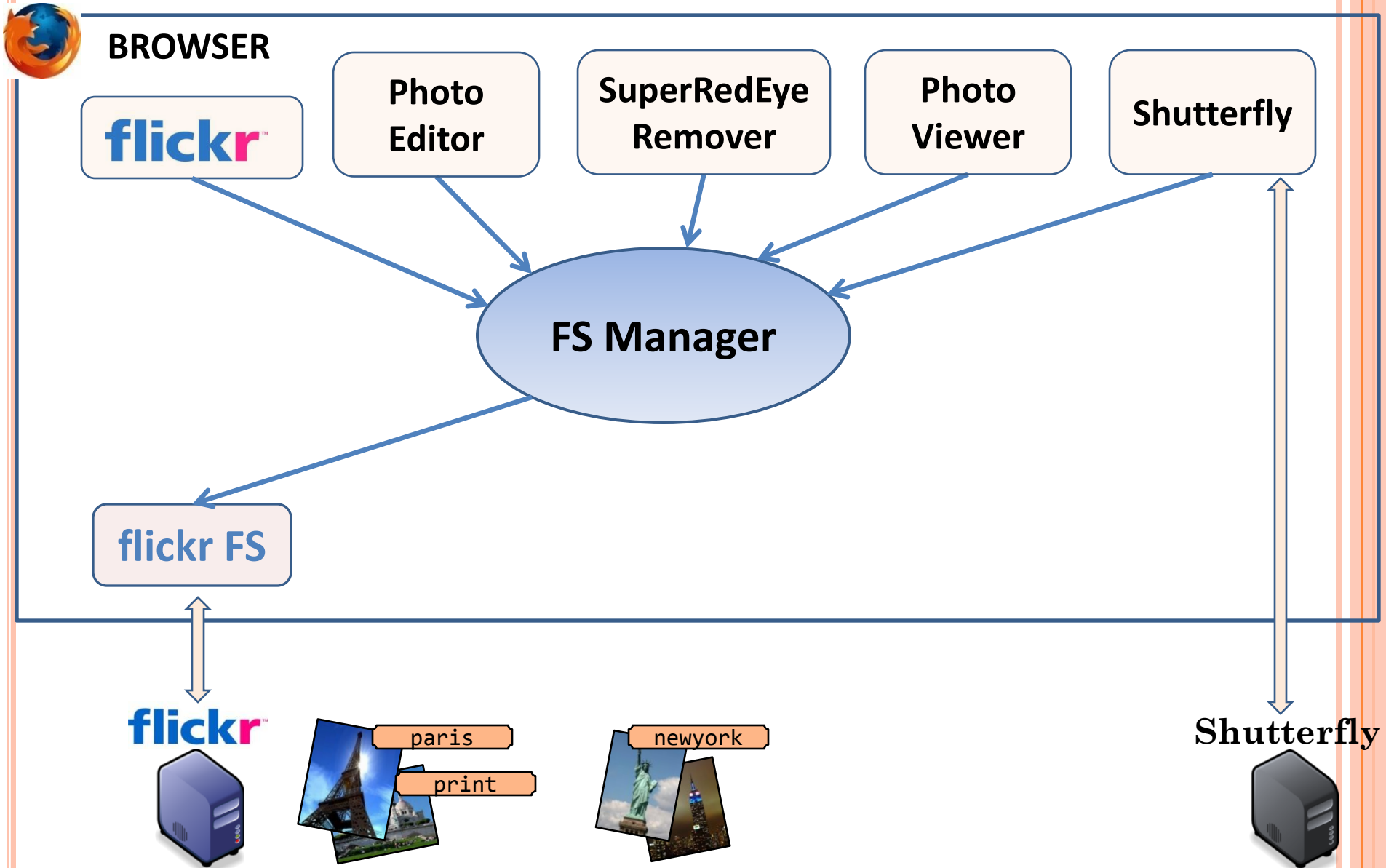
FS ORGANIZATION WITH TAGS



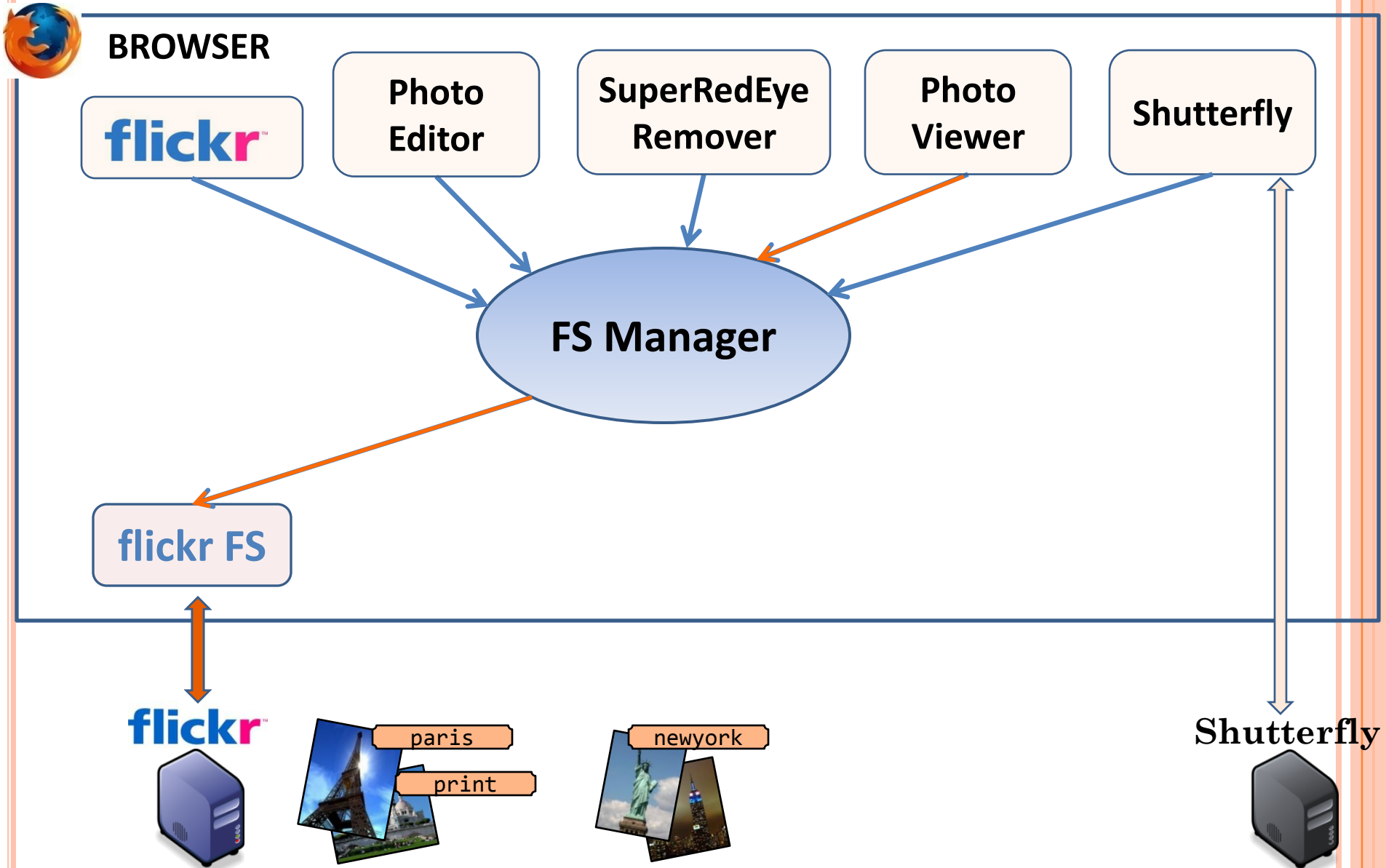
FS ORGANIZATION WITH TAGS



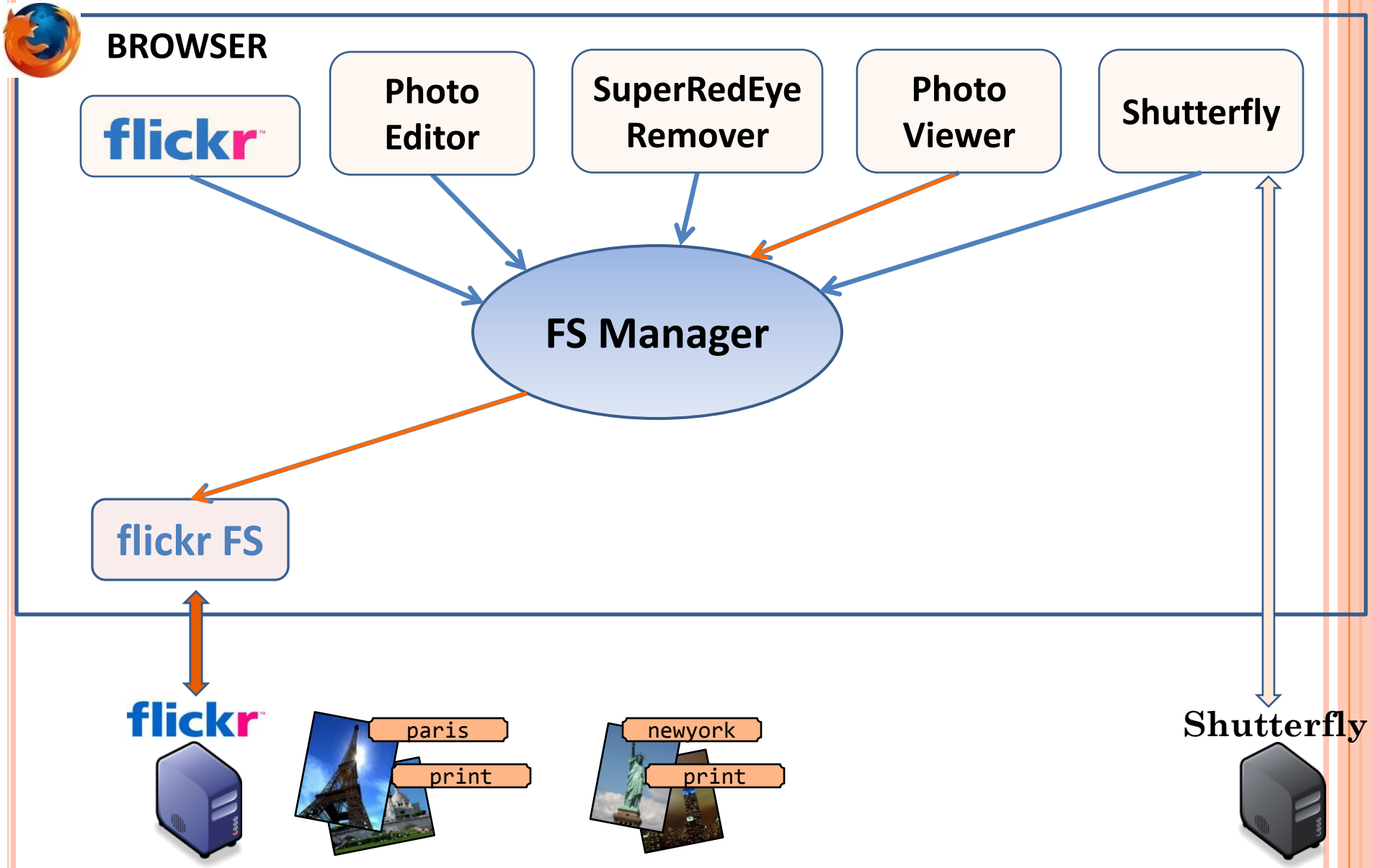
FS ORGANIZATION WITH TAGS



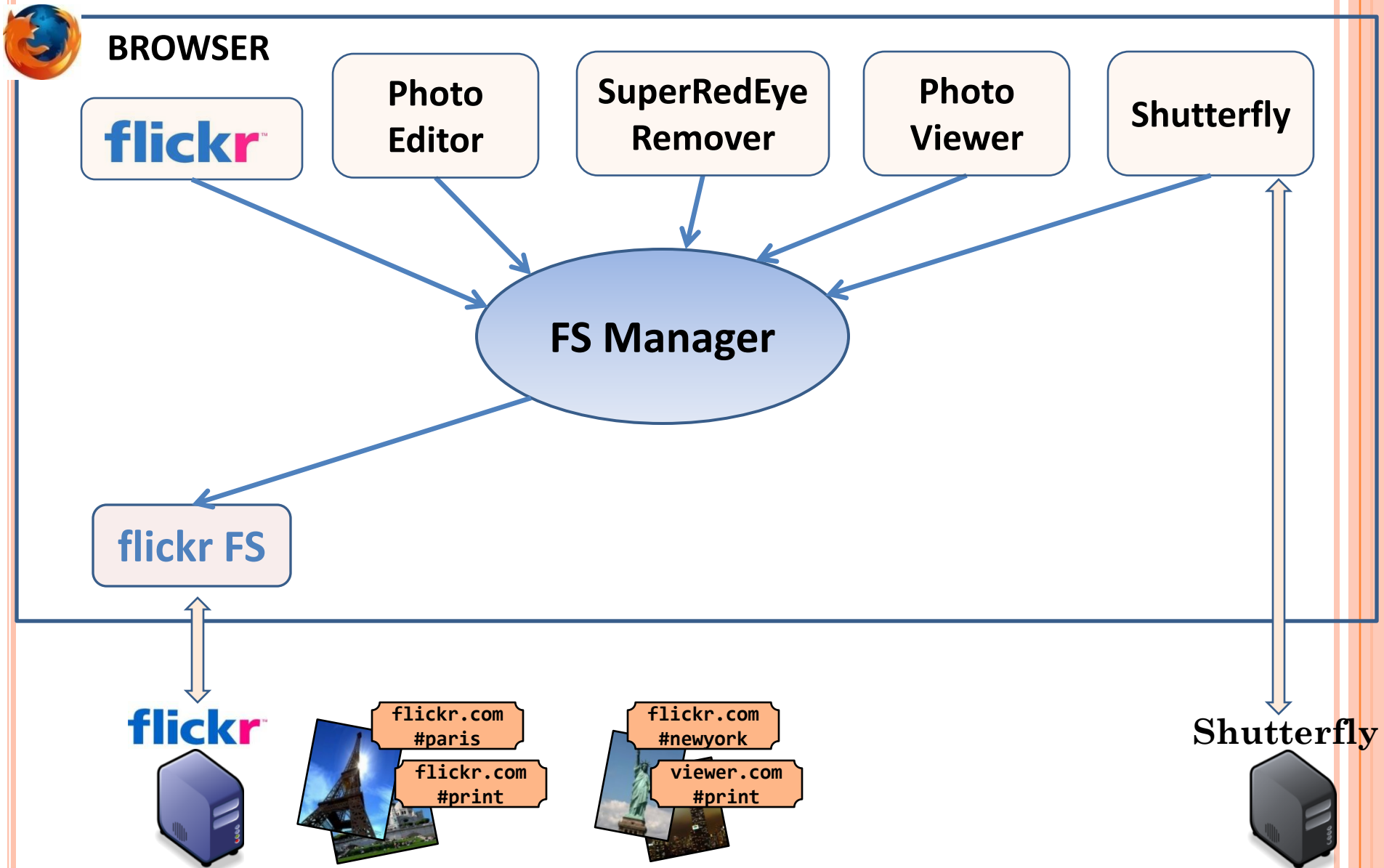
FS ORGANIZATION WITH TAGS



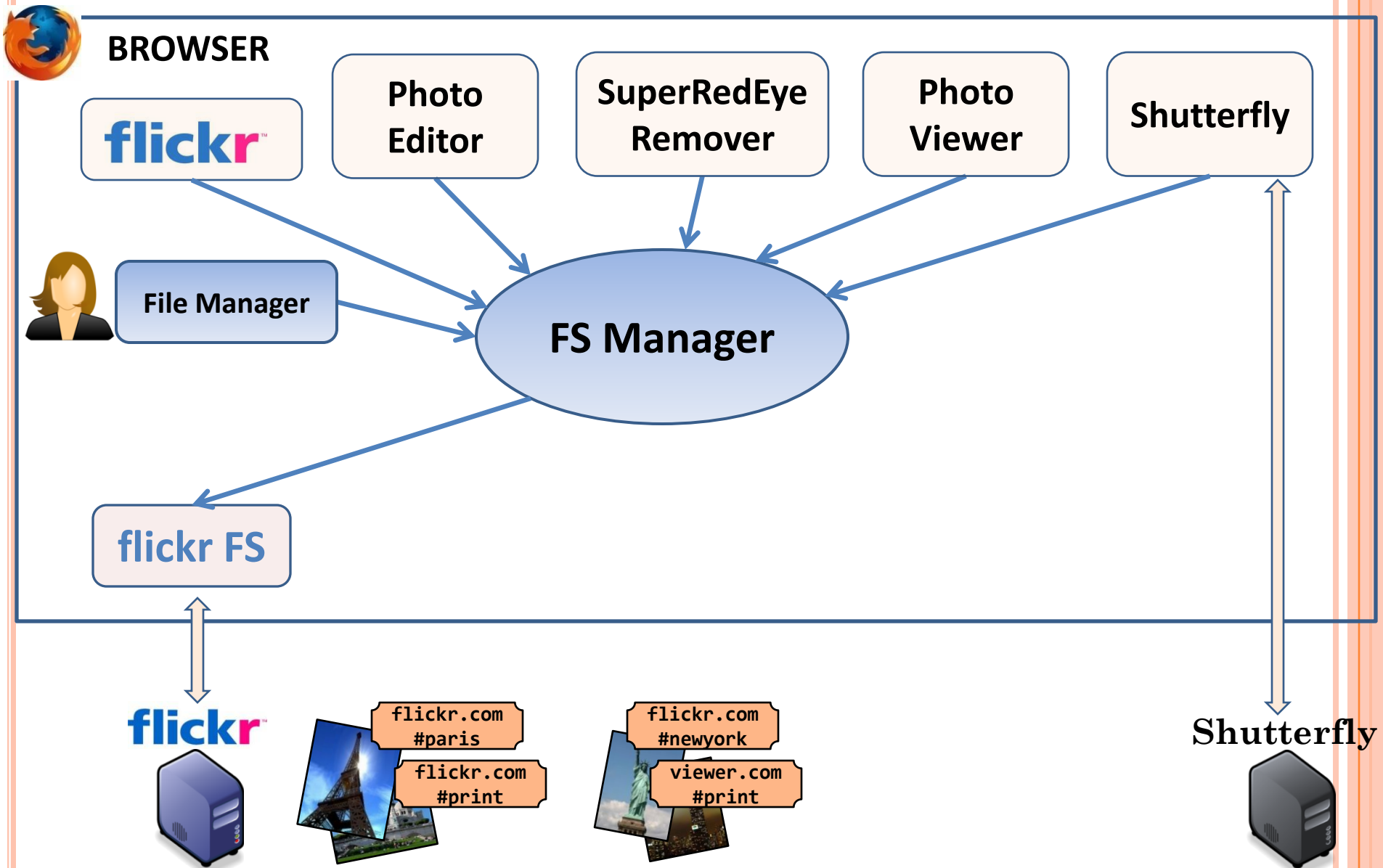
FS ORGANIZATION WITH TAGS



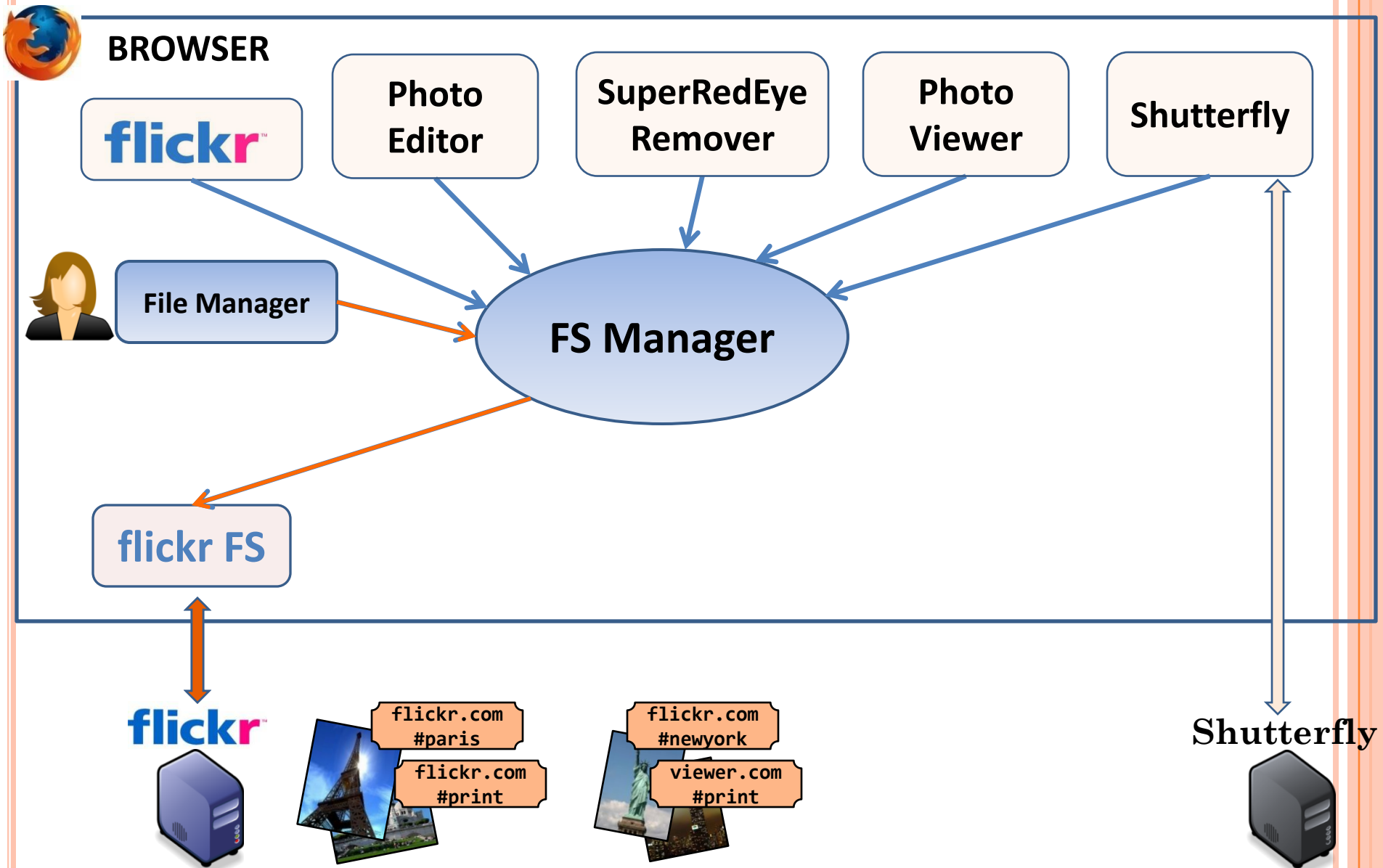
FS ORGANIZATION WITH TAGS



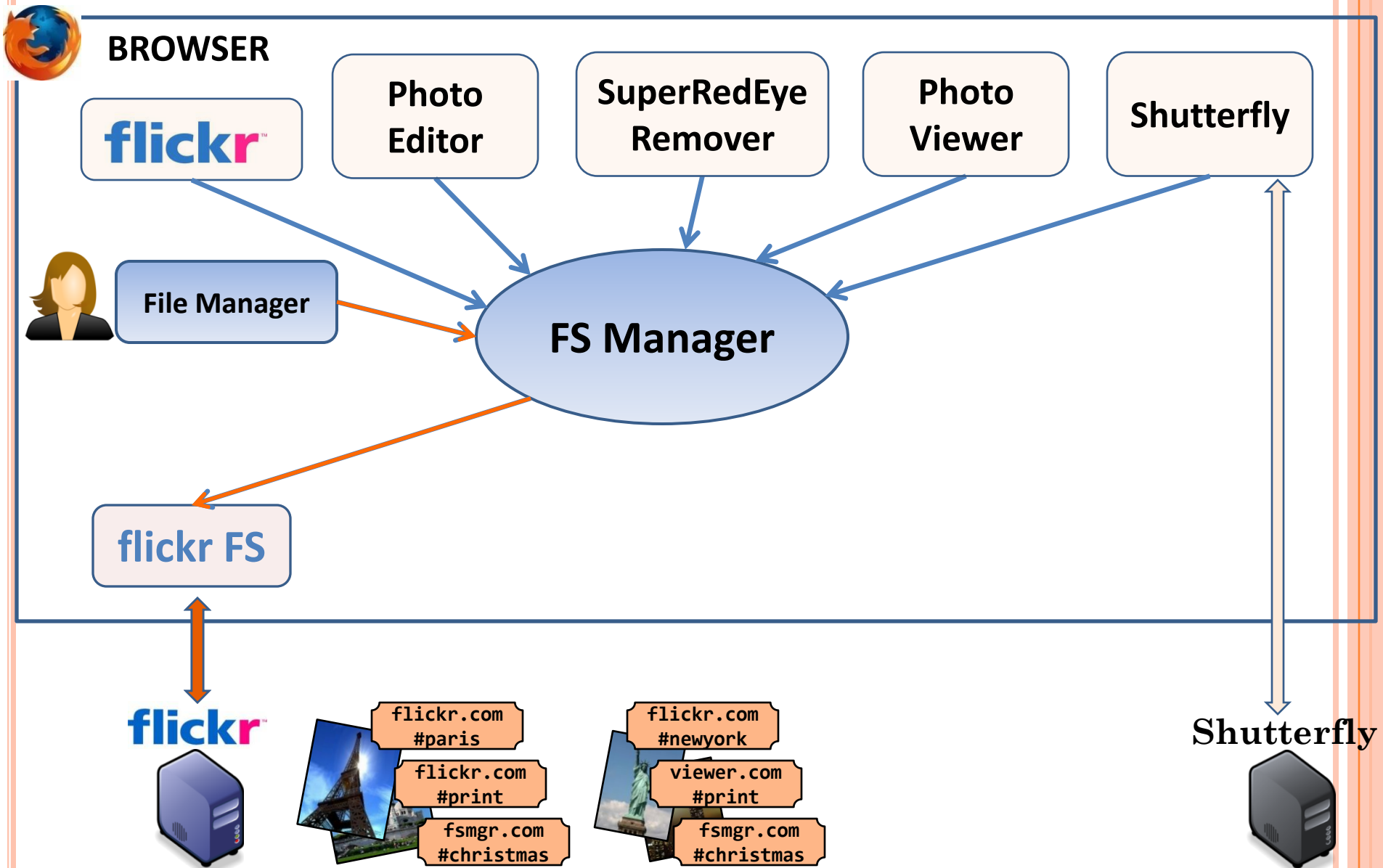
FS ORGANIZATION WITH TAGS



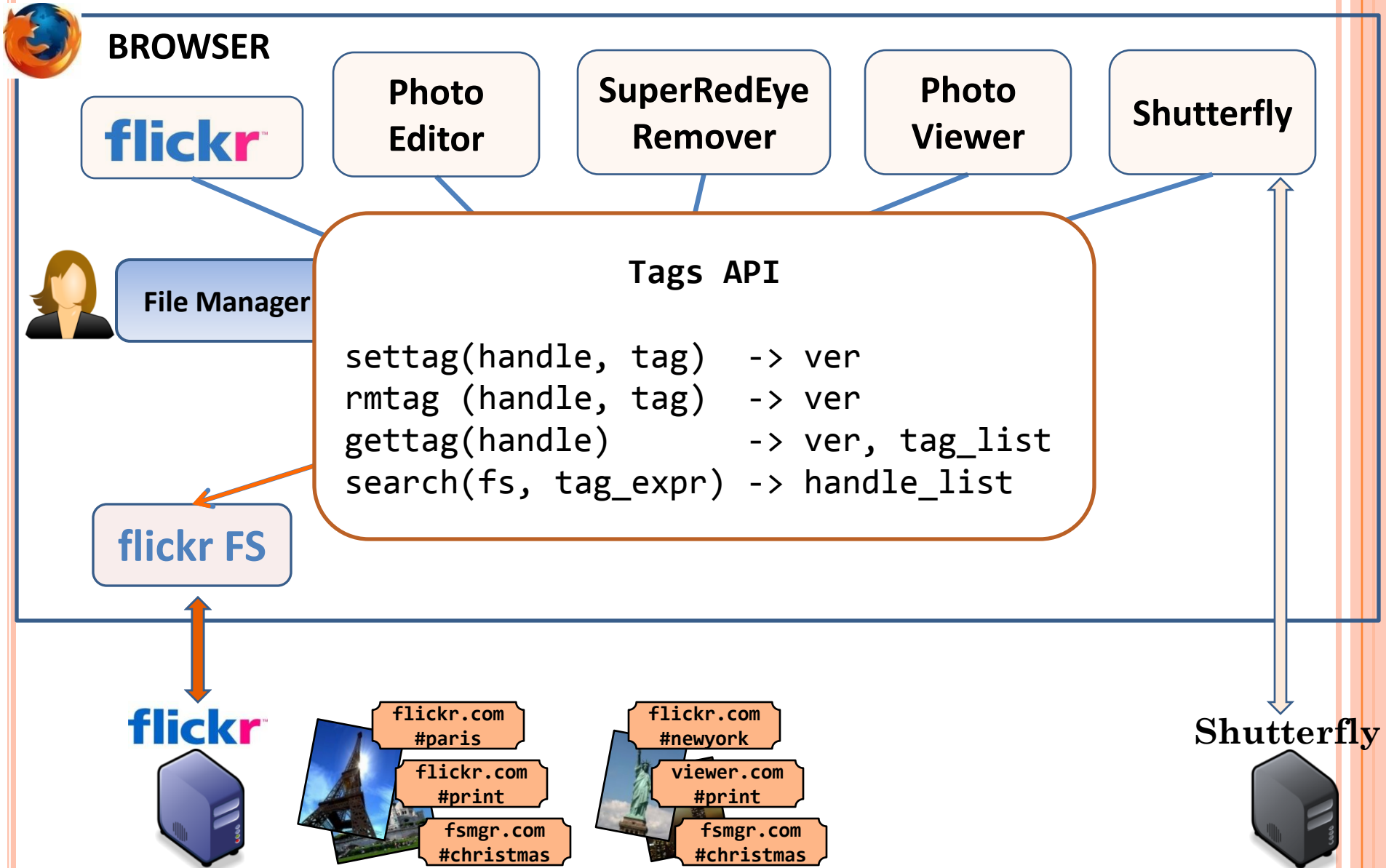
FS ORGANIZATION WITH TAGS



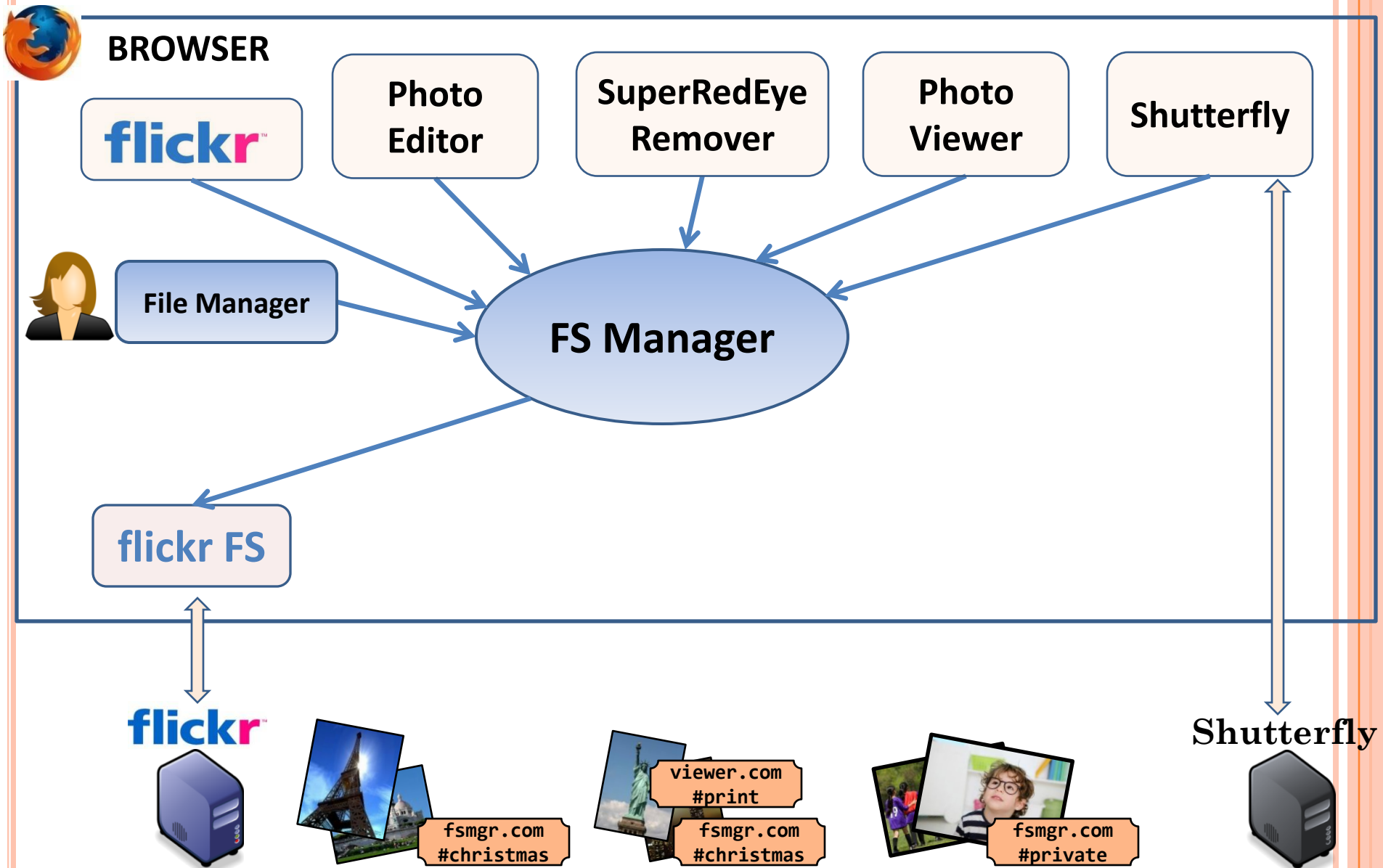
FS ORGANIZATION WITH TAGS



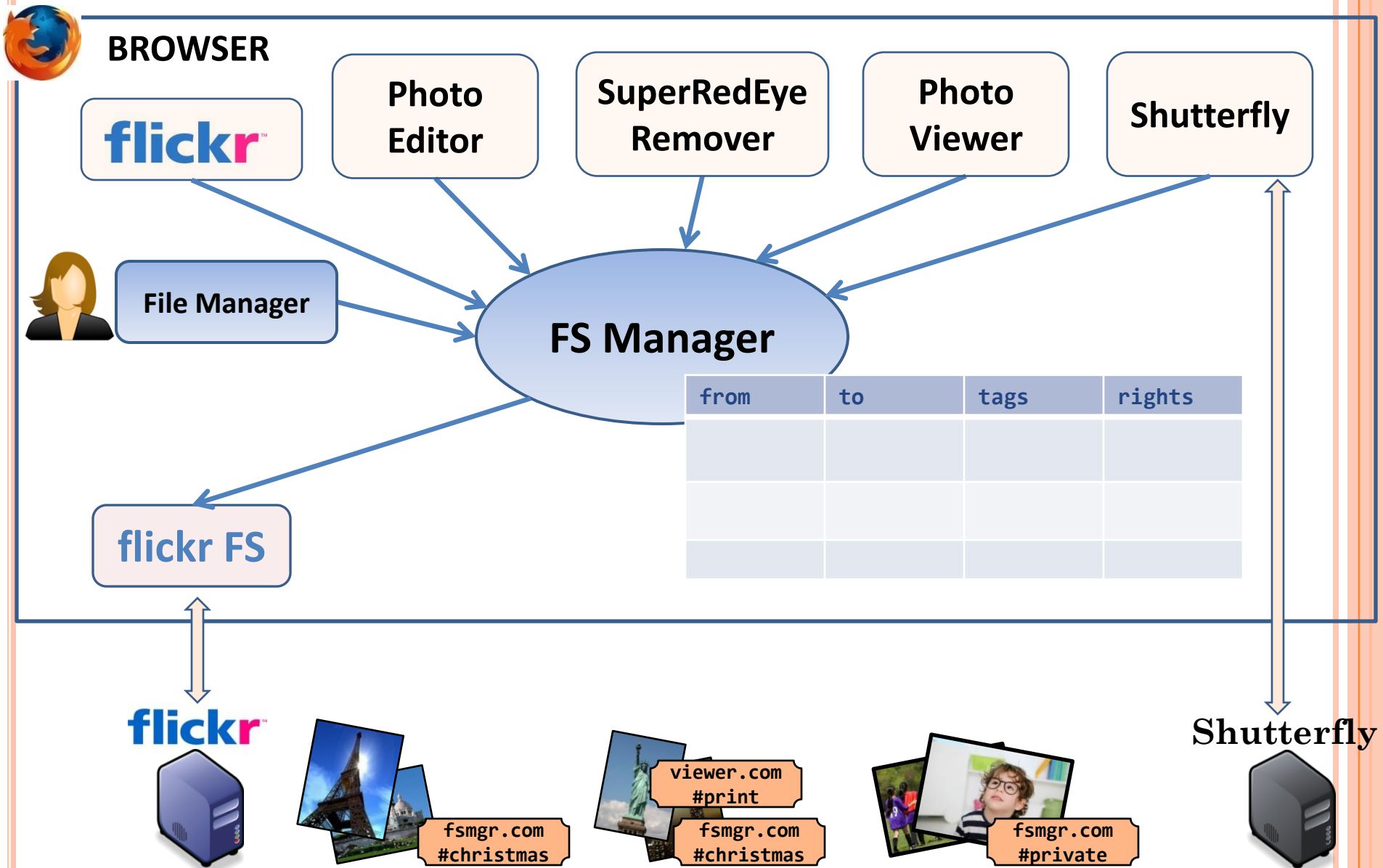
FS ORGANIZATION WITH TAGS



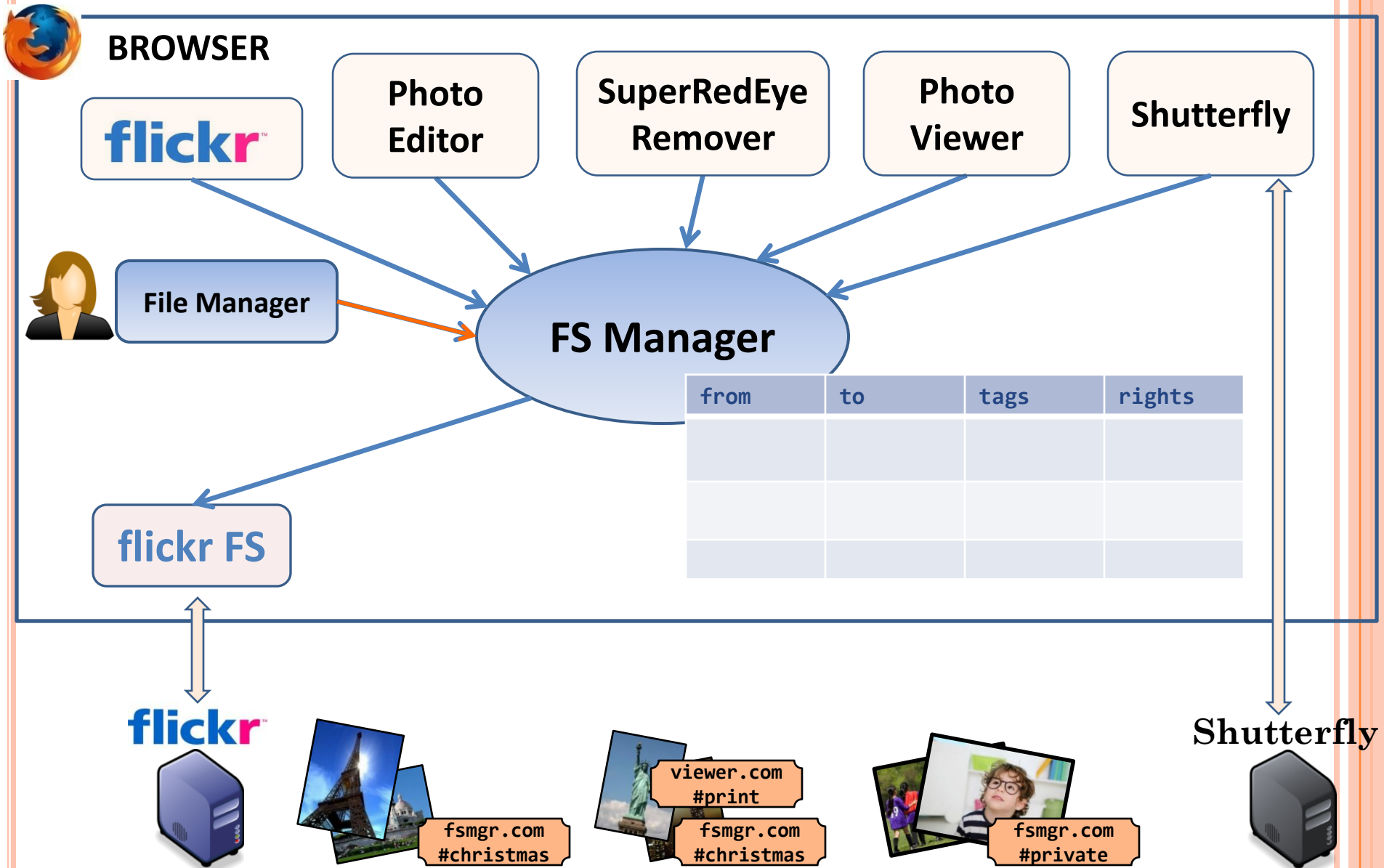
ACCESS CONTROL



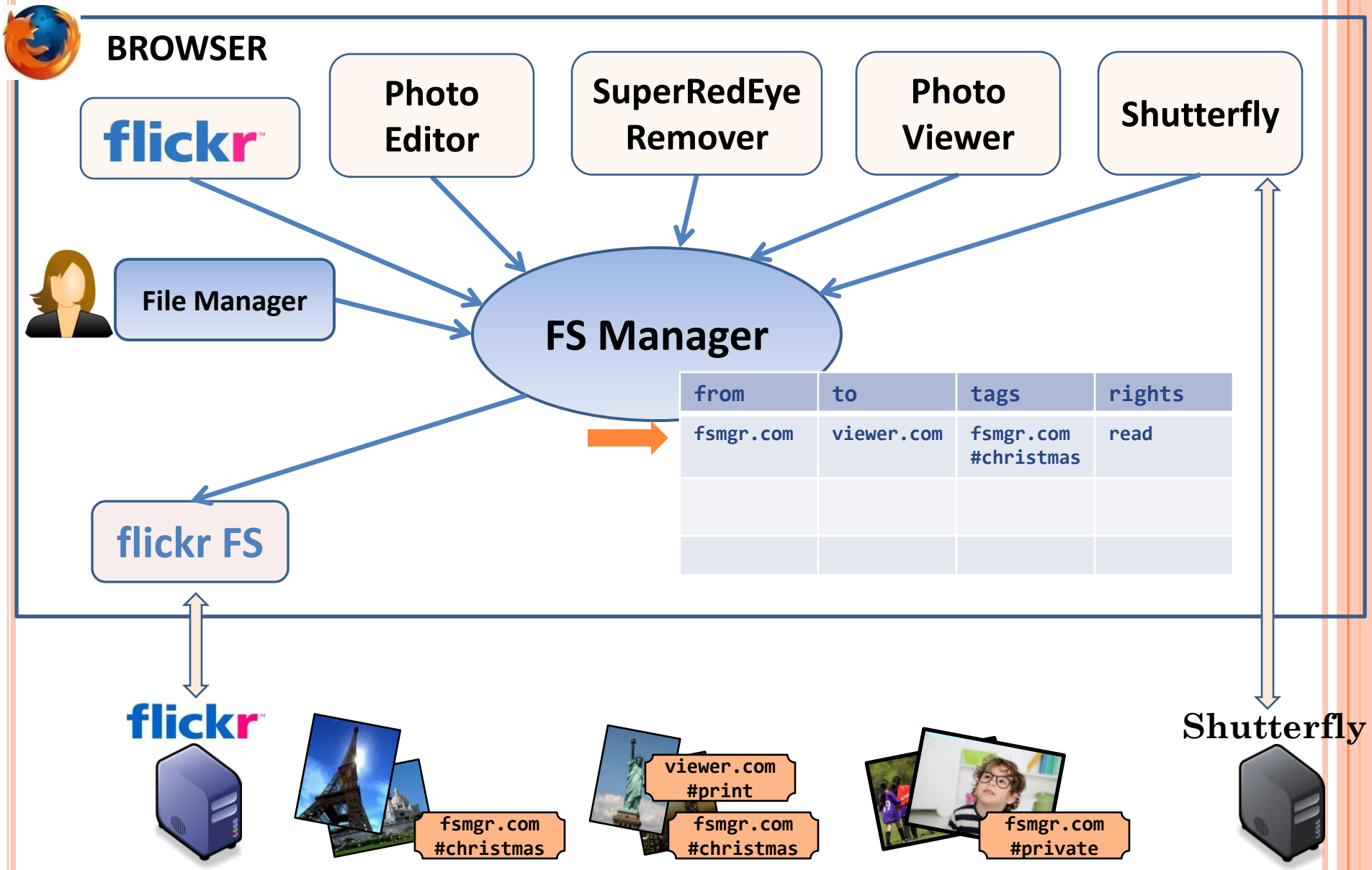
ACCESS CONTROL



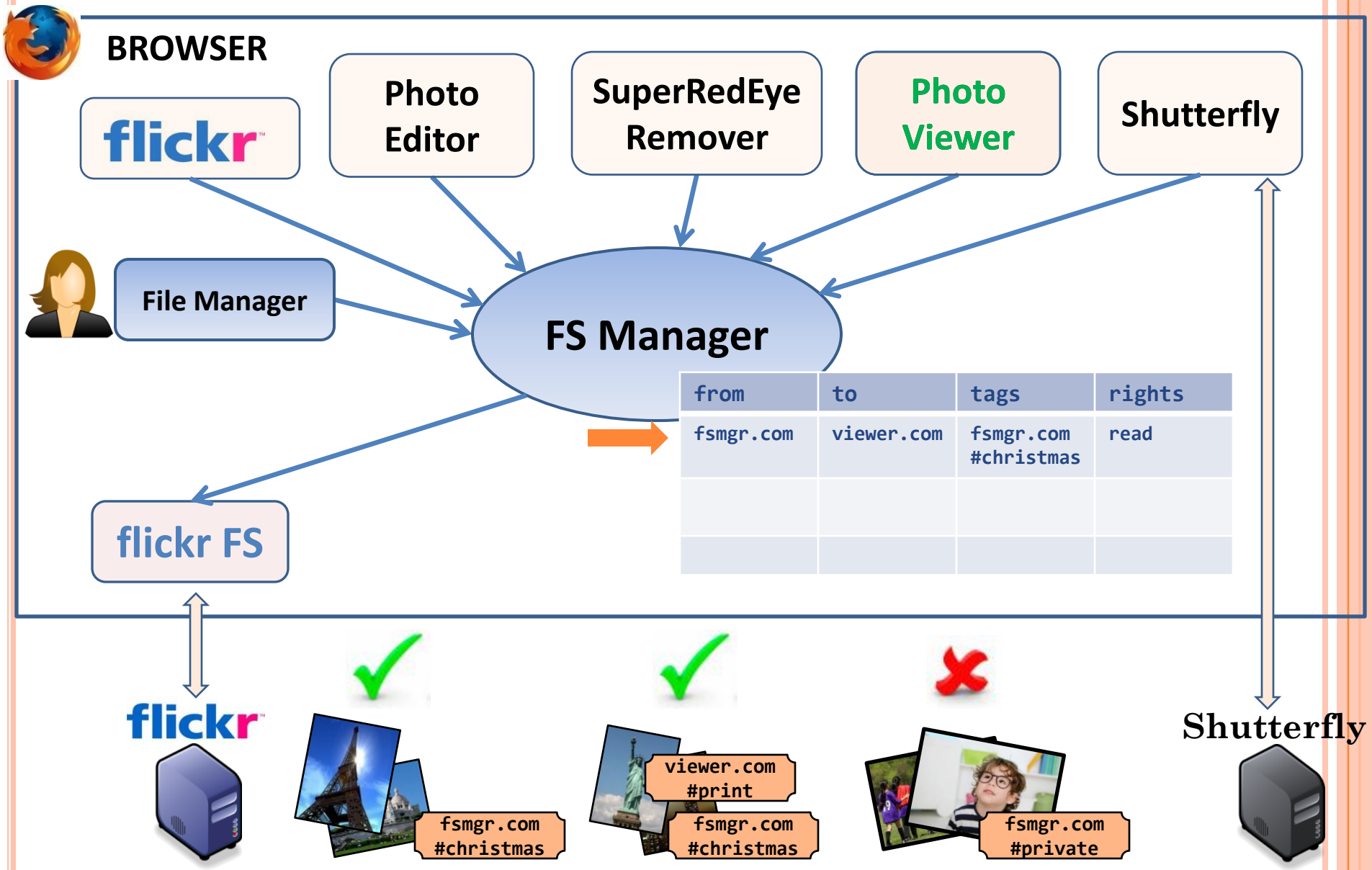
ACCESS CONTROL



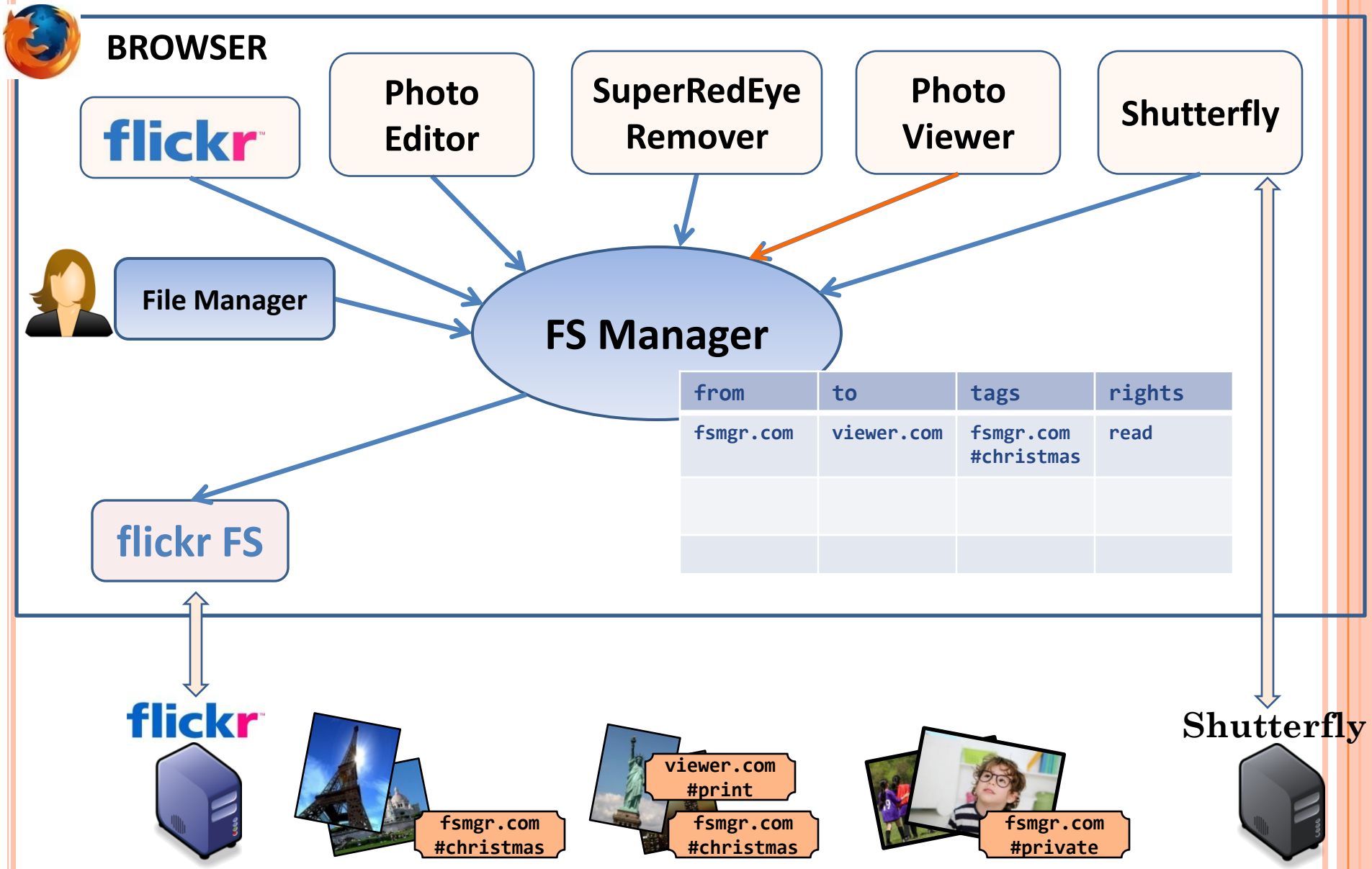
ACCESS CONTROL



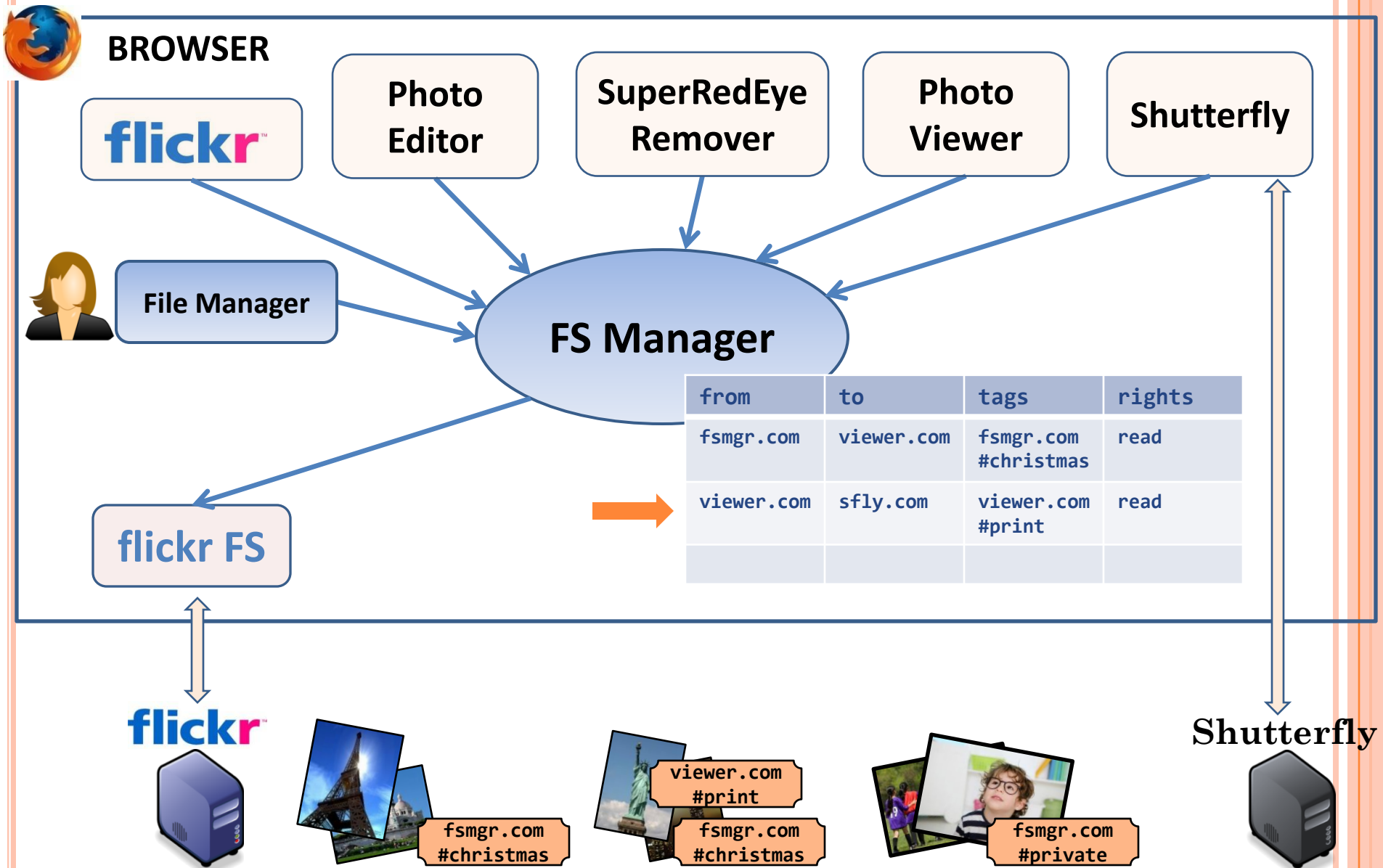
ACCESS CONTROL



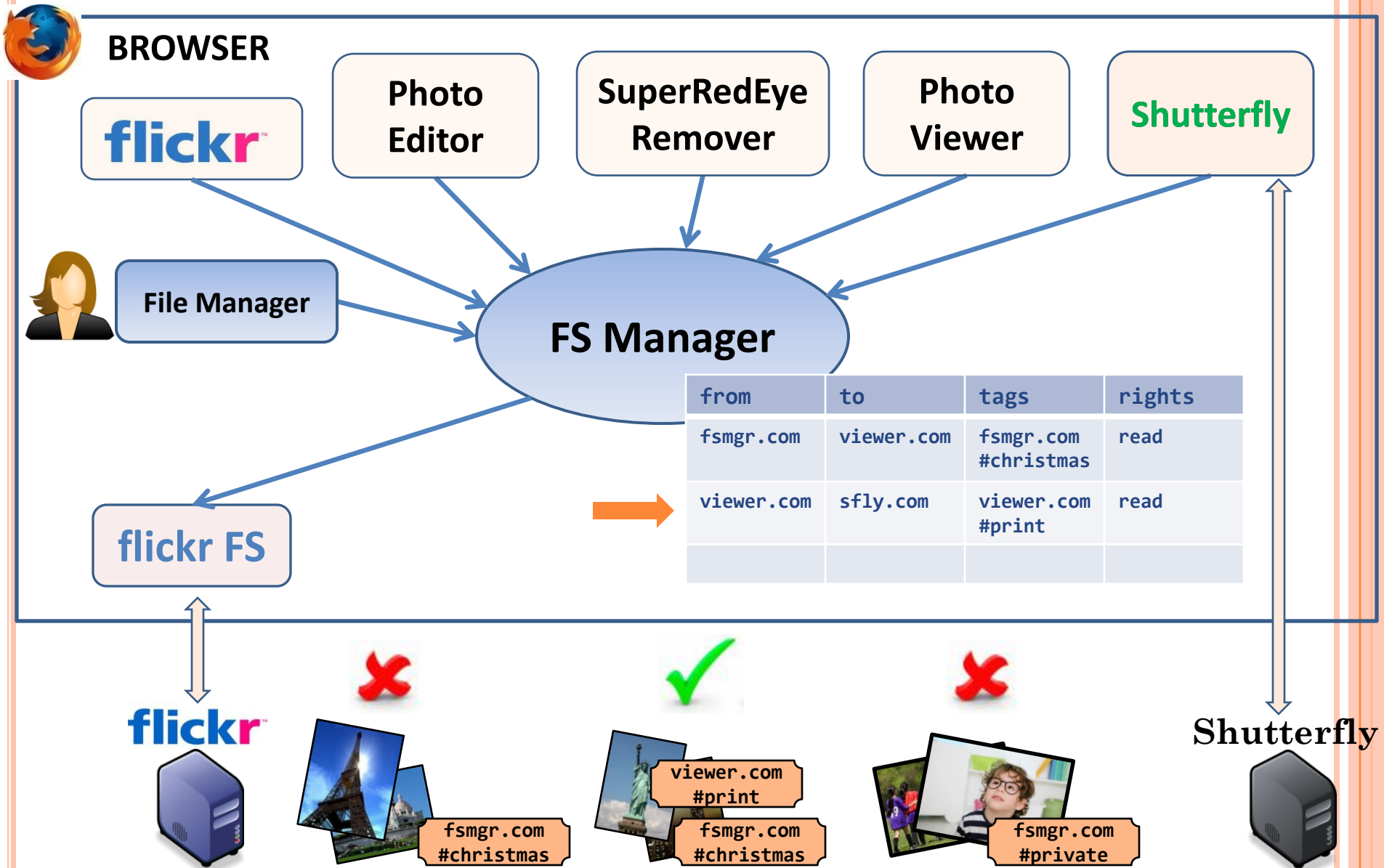
ACCESS CONTROL



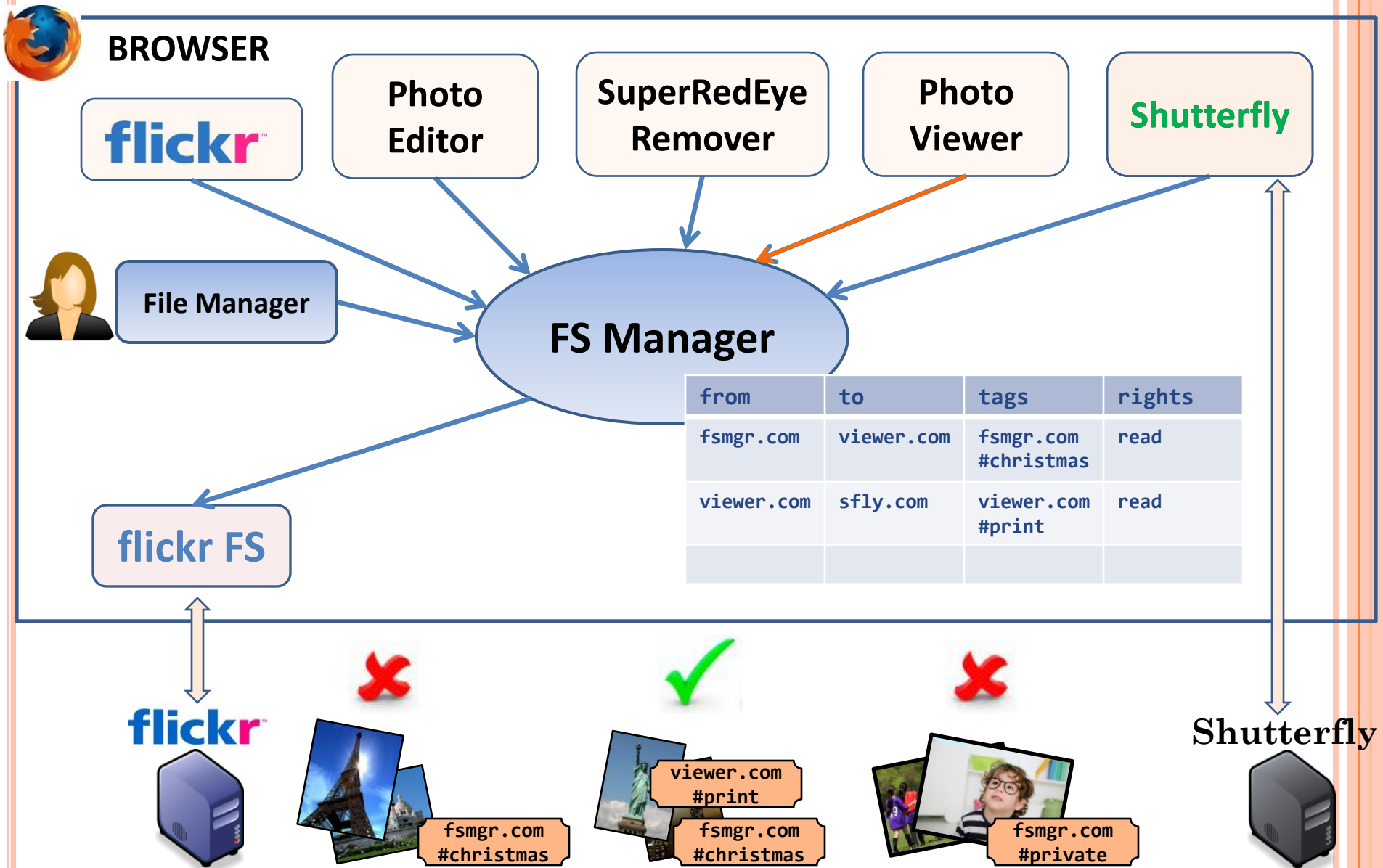
ACCESS CONTROL



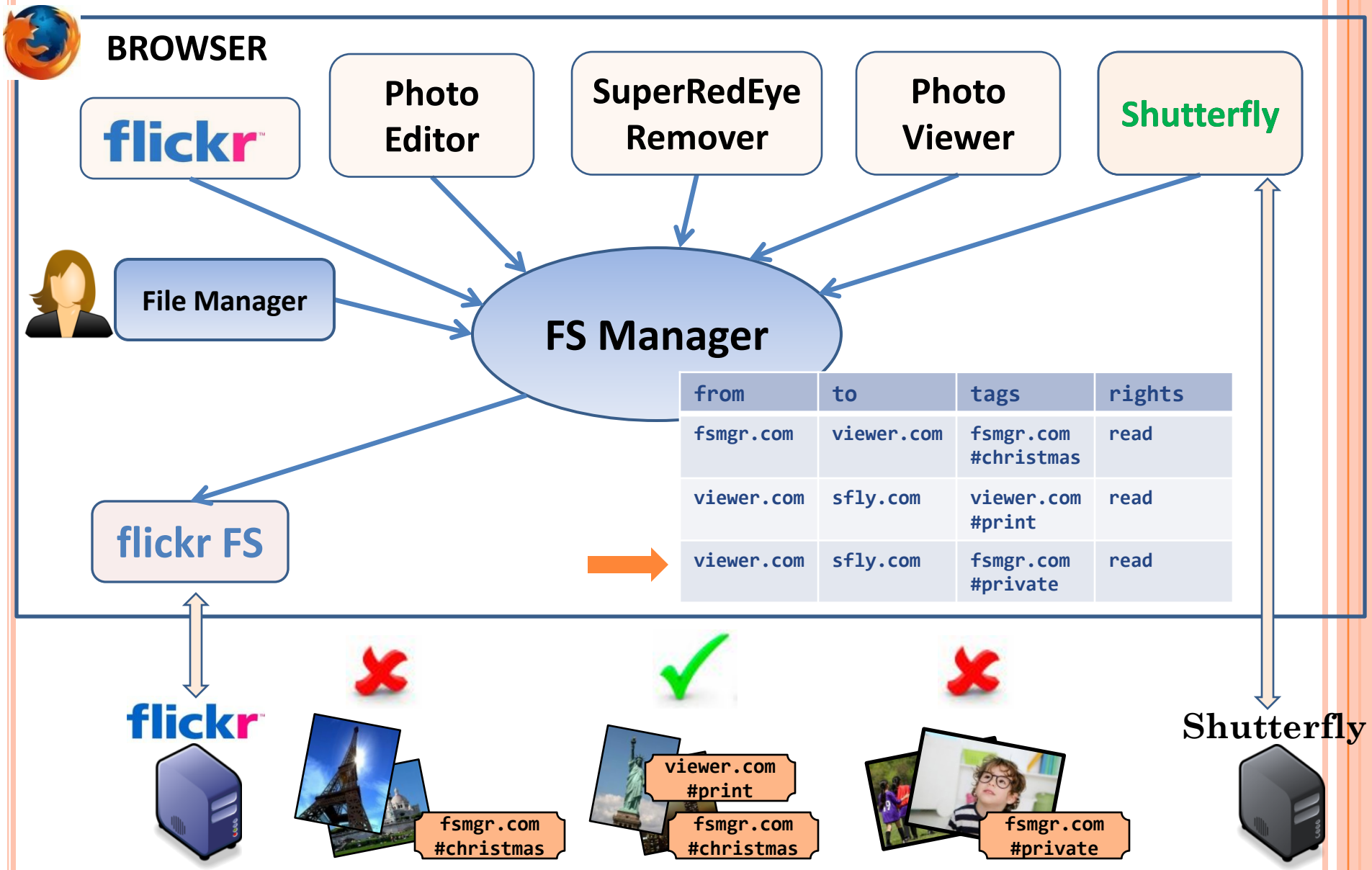
ACCESS CONTROL



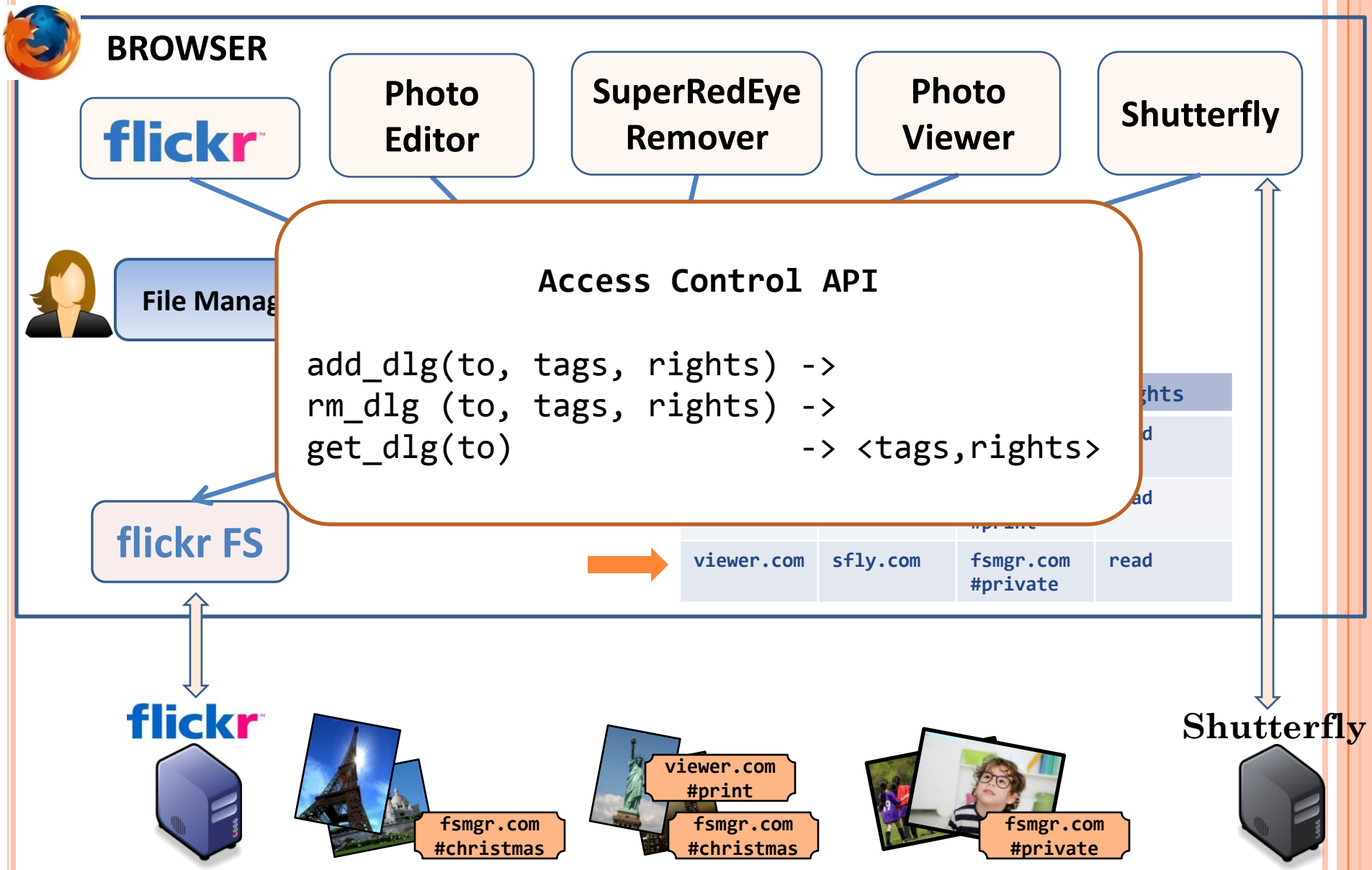
ACCESS CONTROL



ACCESS CONTROL



ACCESS CONTROL

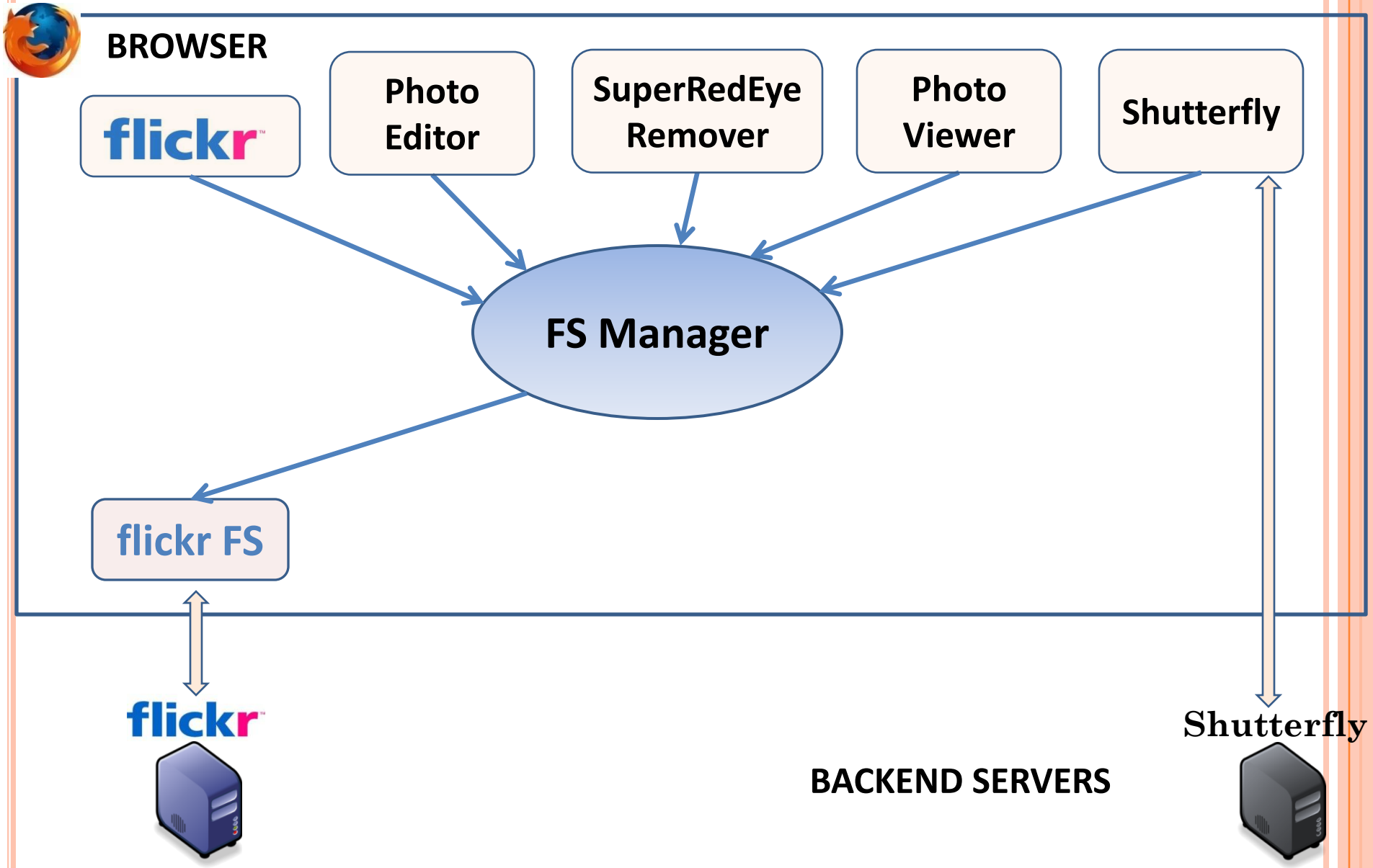


FILESYSTEM SETUP

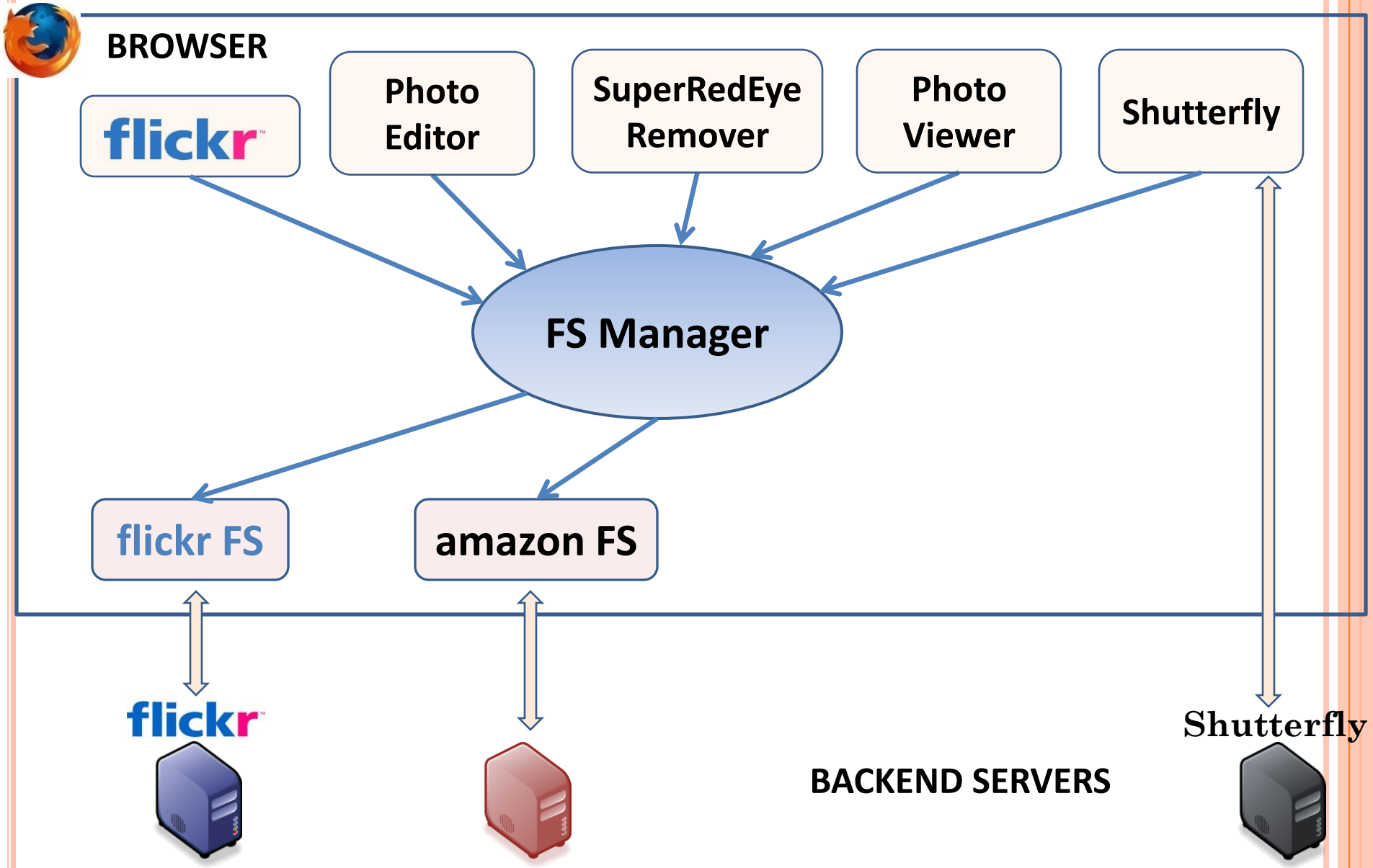
- FS manager needs FS info to mount the FS
- FS info stored in a mountpoint file
 - Example:

```
<mountpoint>  
  <url src="http://flickr.com/fs.html"/>  
  <params username="ramesh" password="pass"/>  
</mountpoint>
```
- Any principal can add a mountpoint as long as it can create the mountpoint file
- Mountpoint file contents encrypted by FS manager
 - `encrypt(mntpoint_plaintext) -> mntpoint_ciphertext`

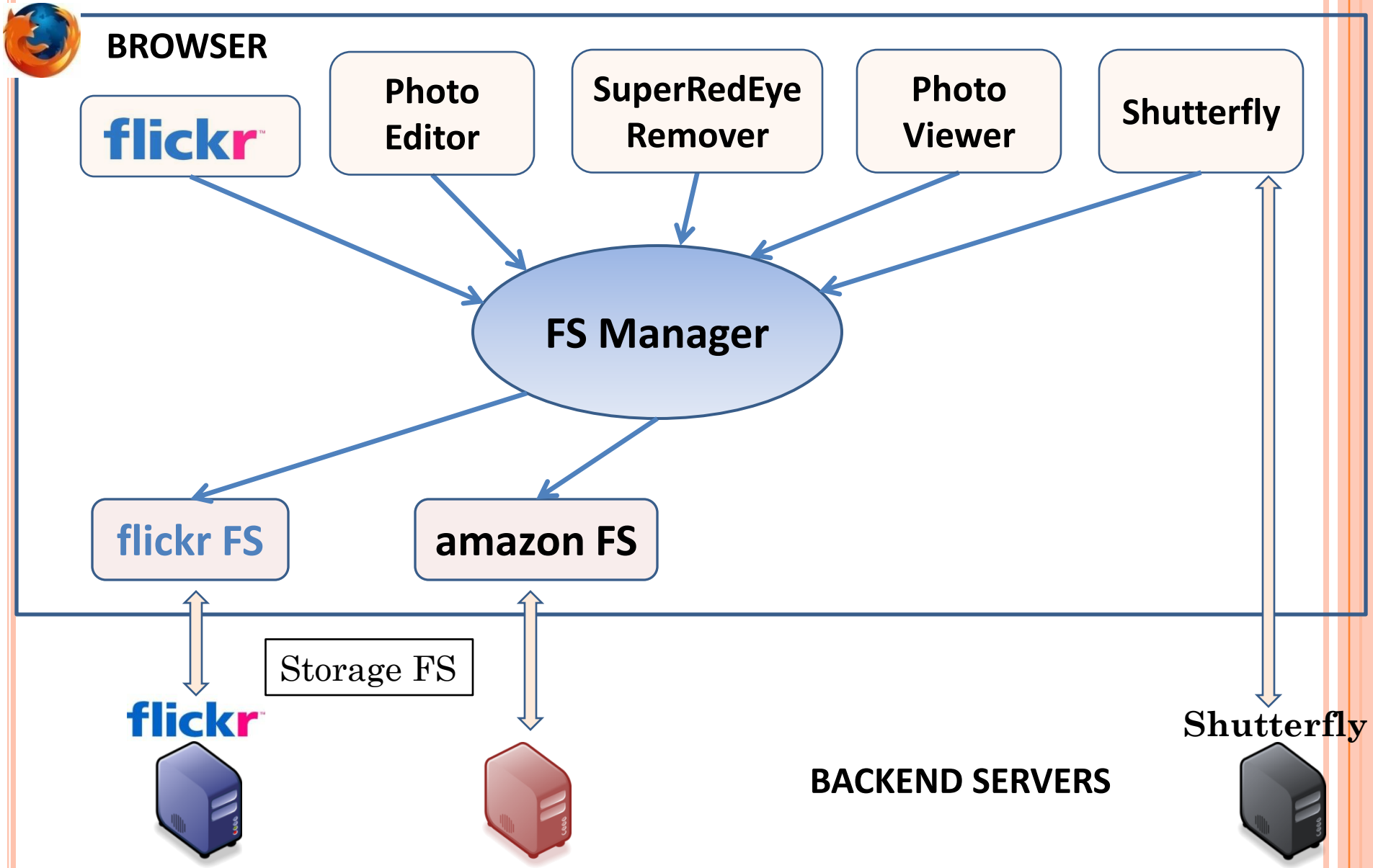
FILESYSTEM EXAMPLES



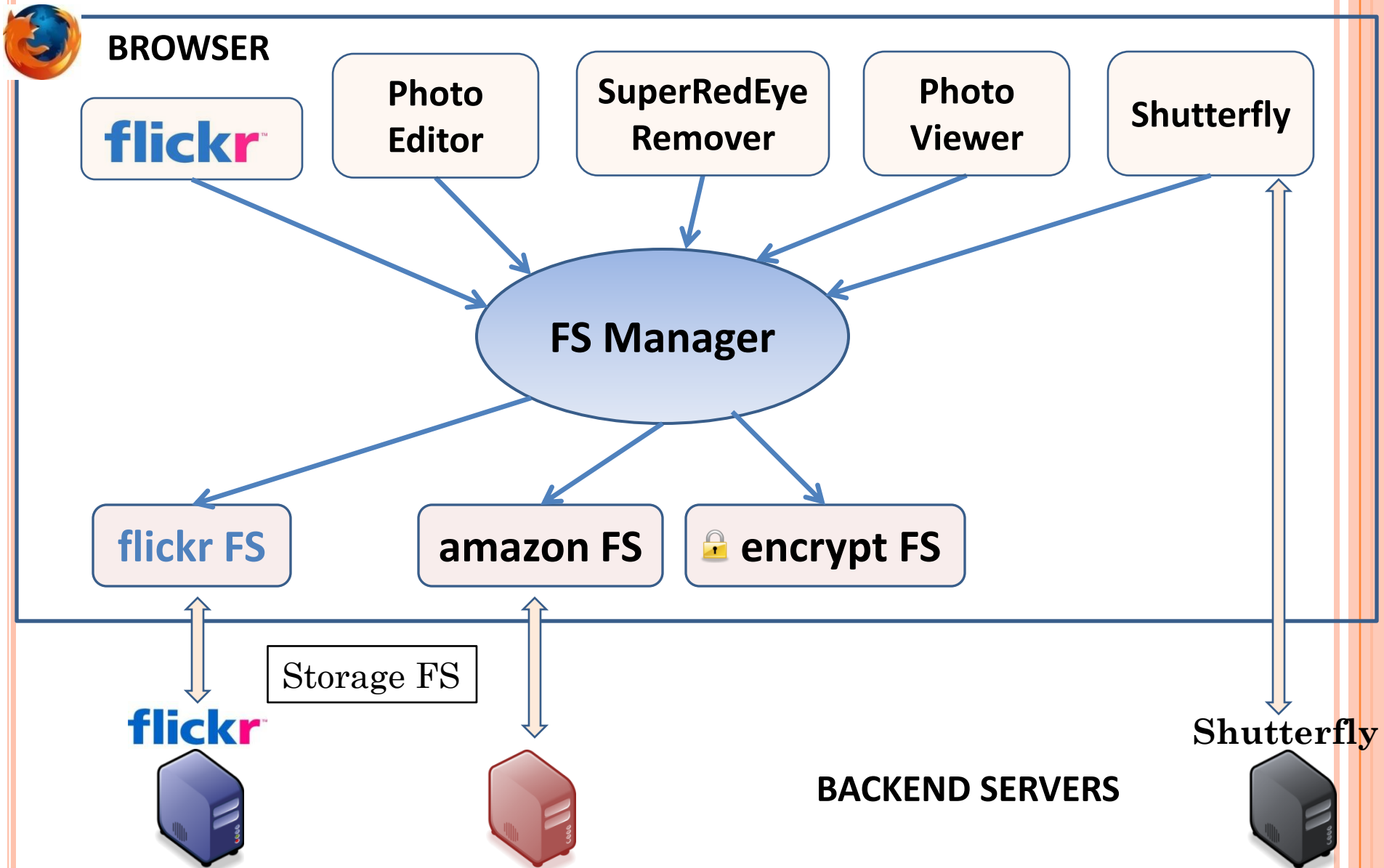
FILESYSTEM EXAMPLES



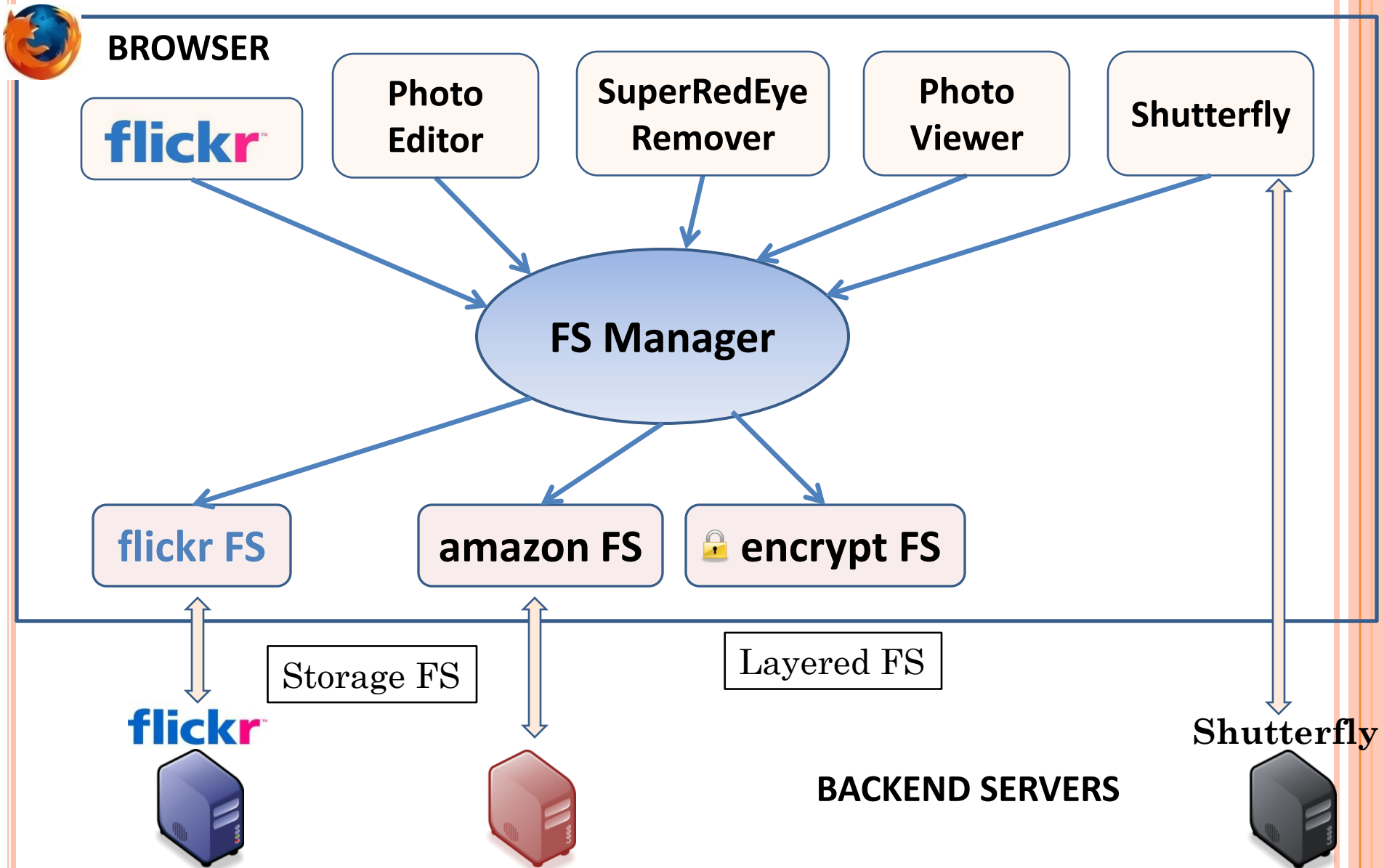
FILESYSTEM EXAMPLES



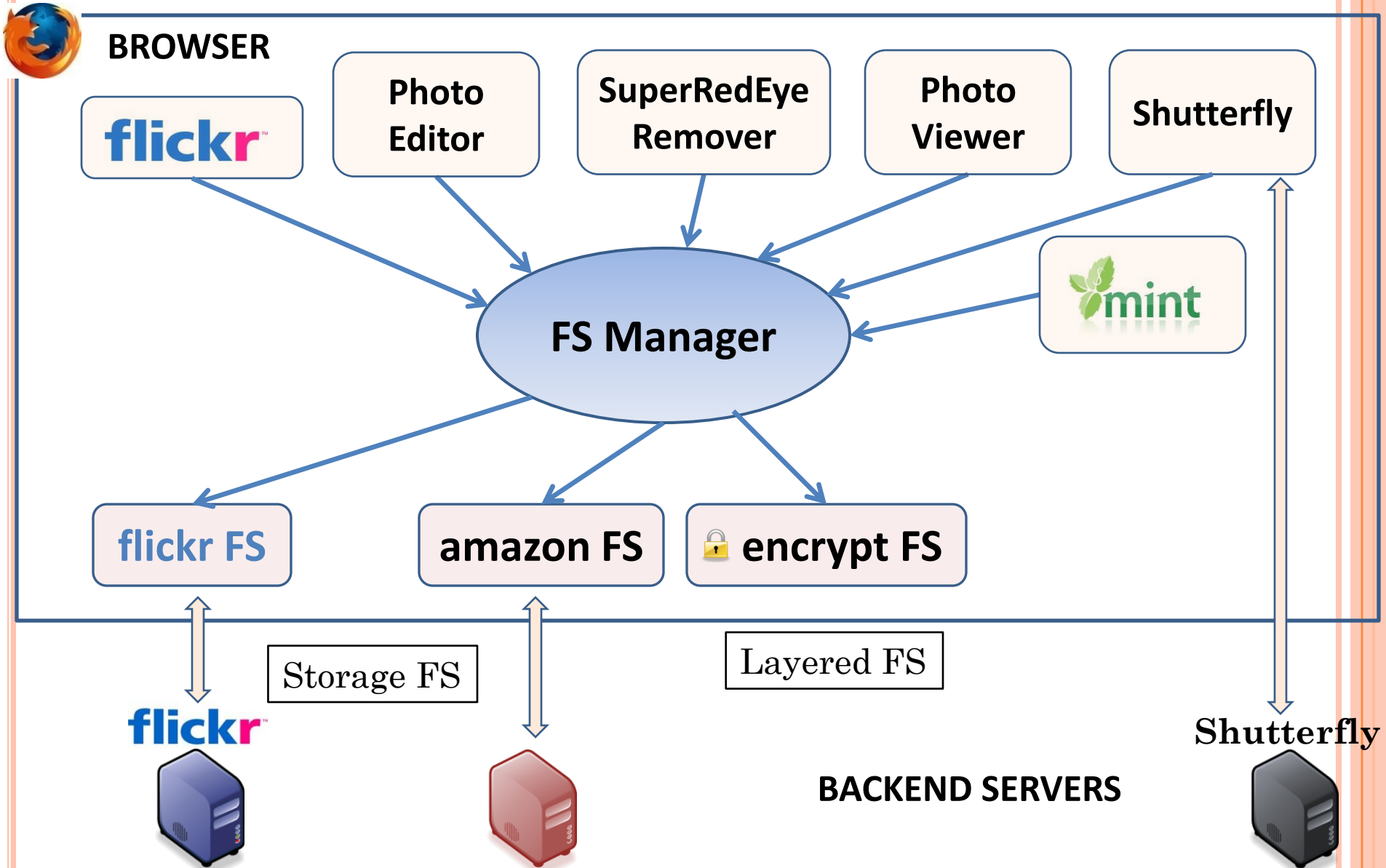
FILESYSTEM EXAMPLES



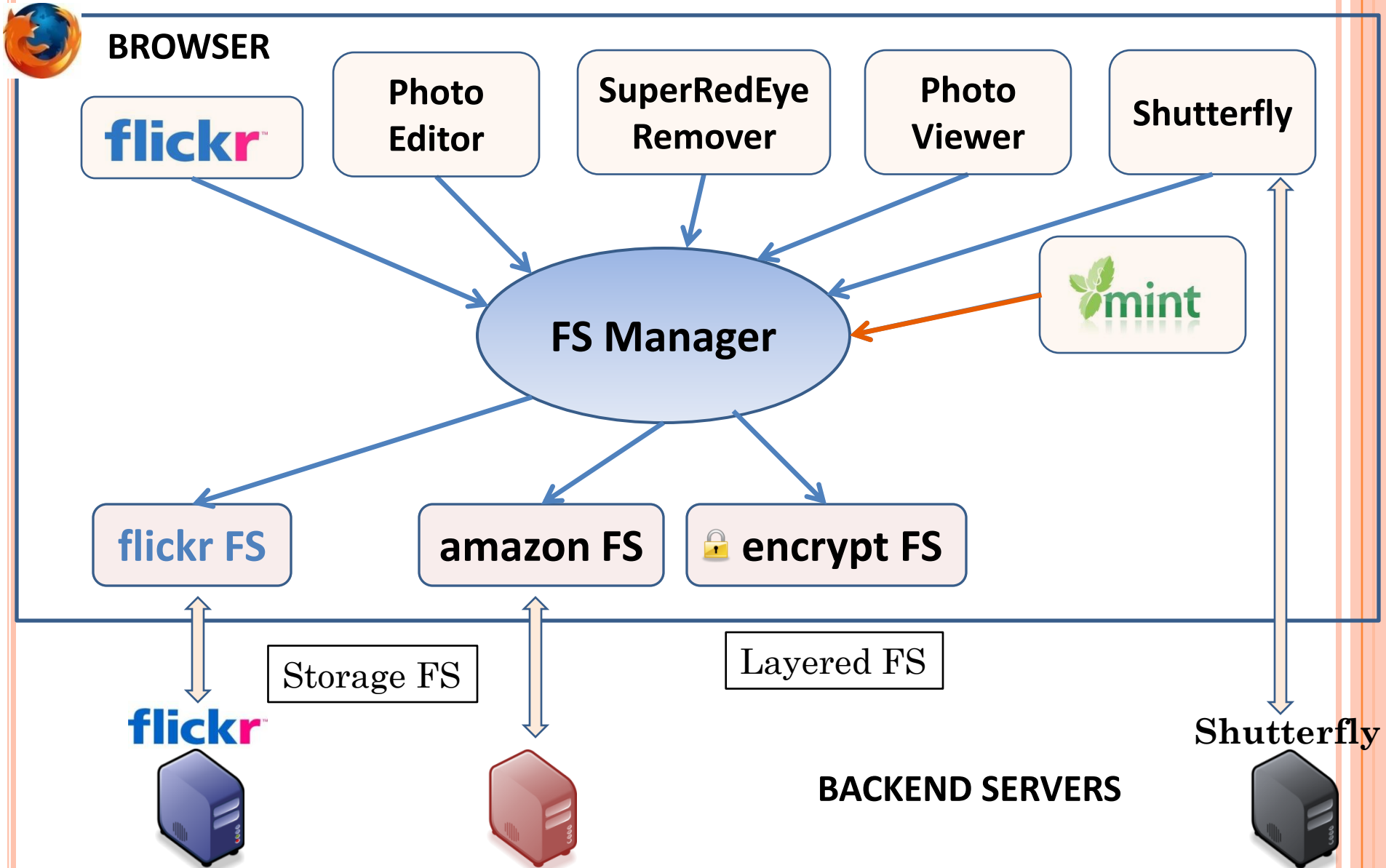
FILESYSTEM EXAMPLES



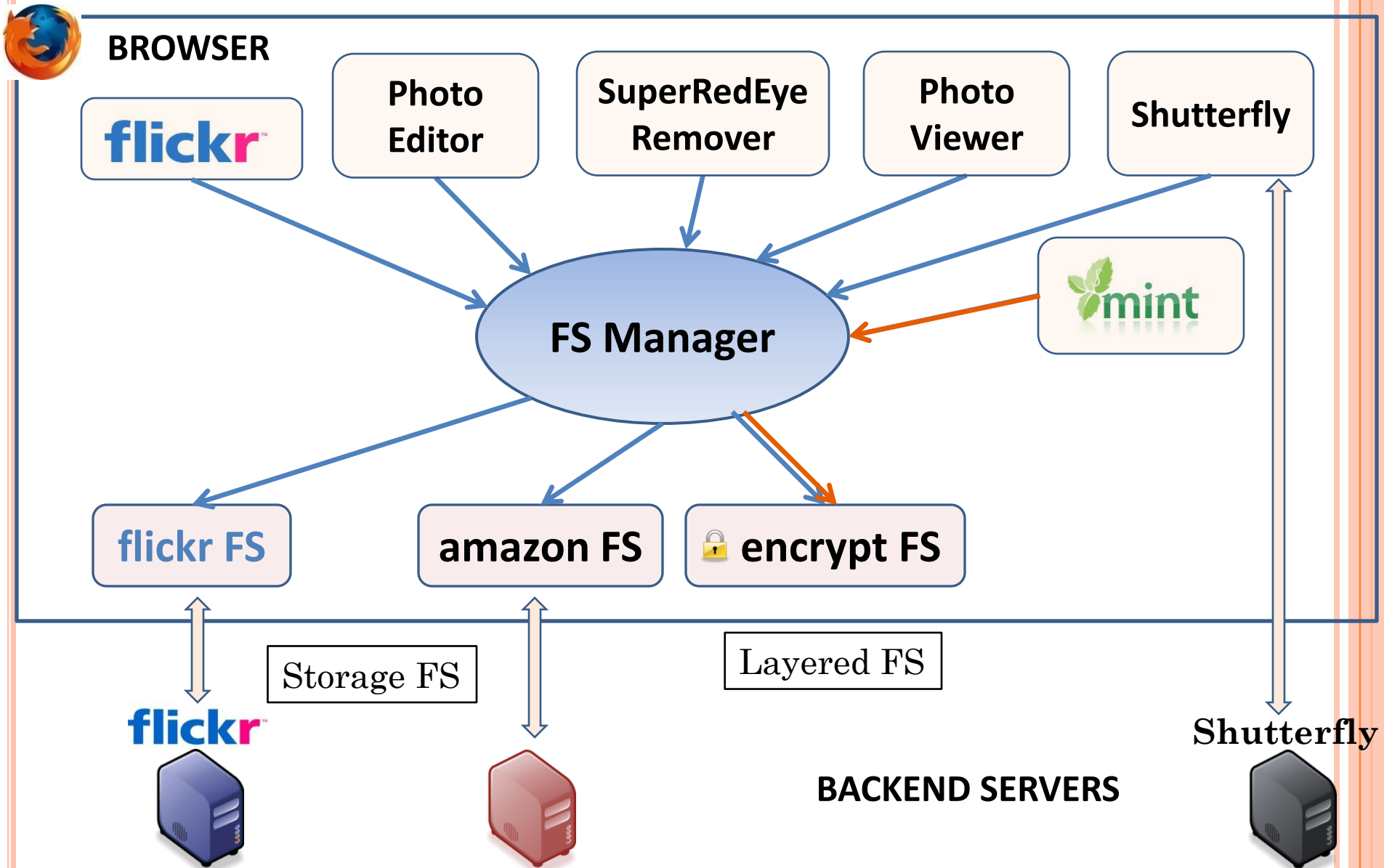
FILESYSTEM EXAMPLES



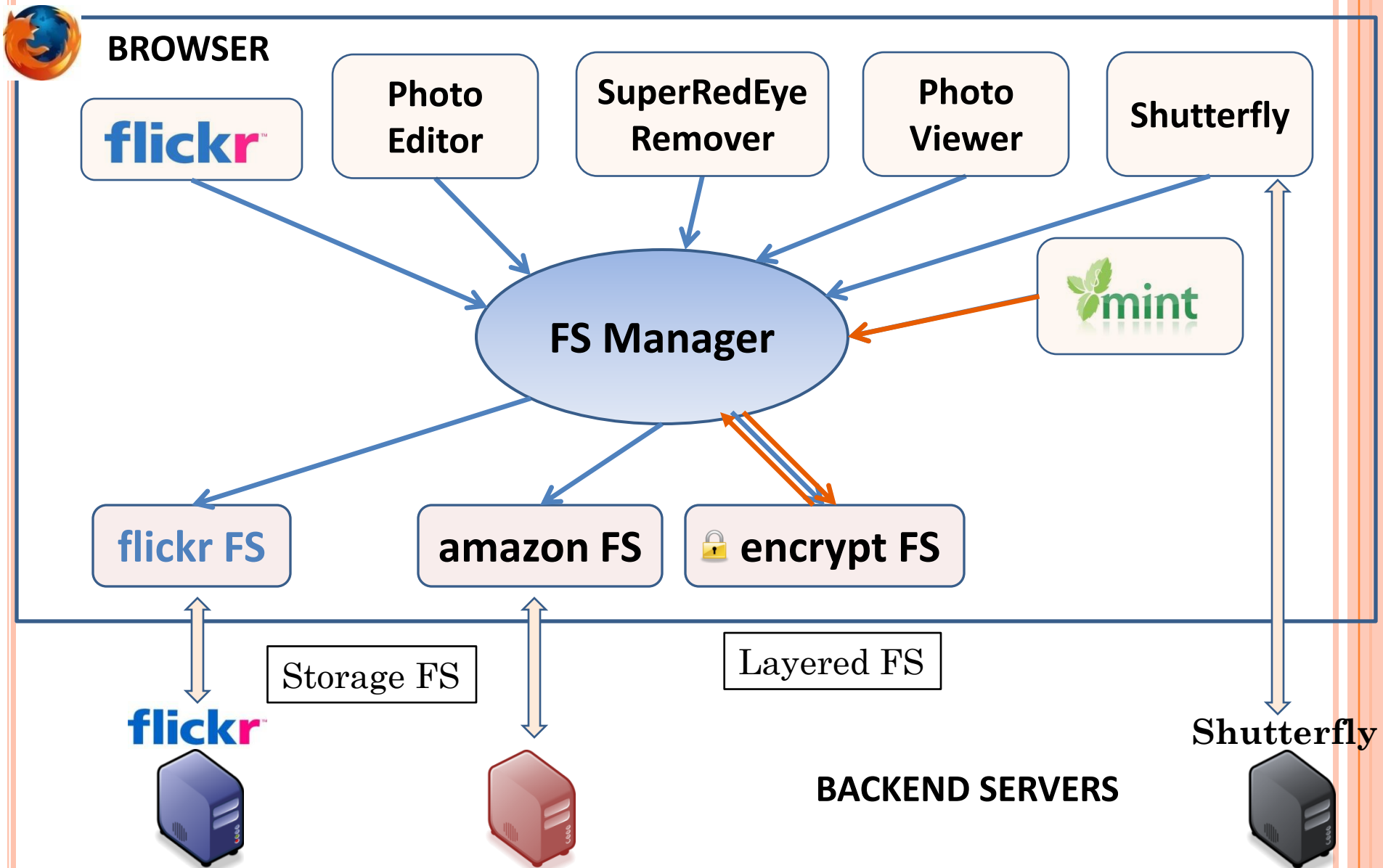
FILESYSTEM EXAMPLES



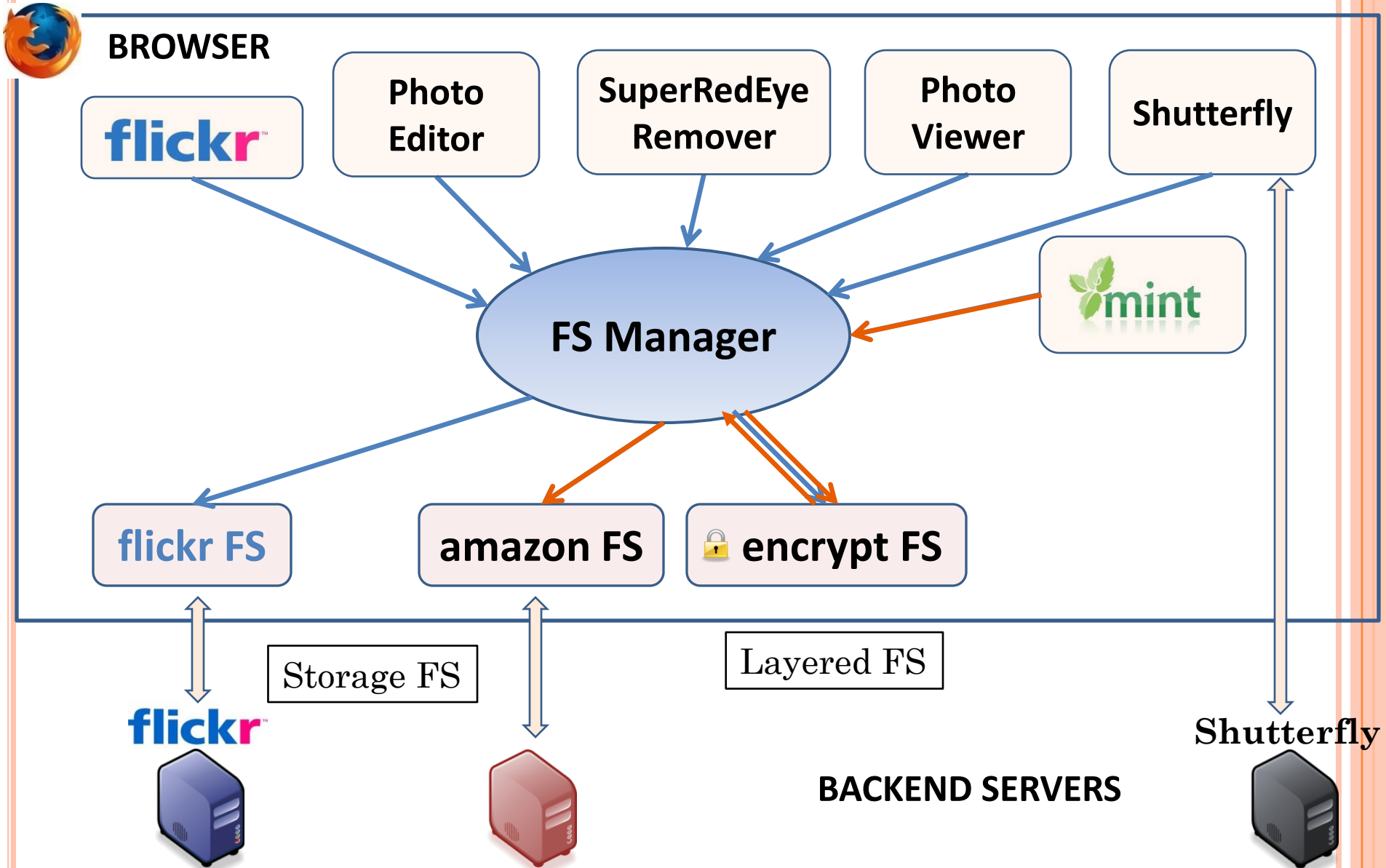
FILESYSTEM EXAMPLES



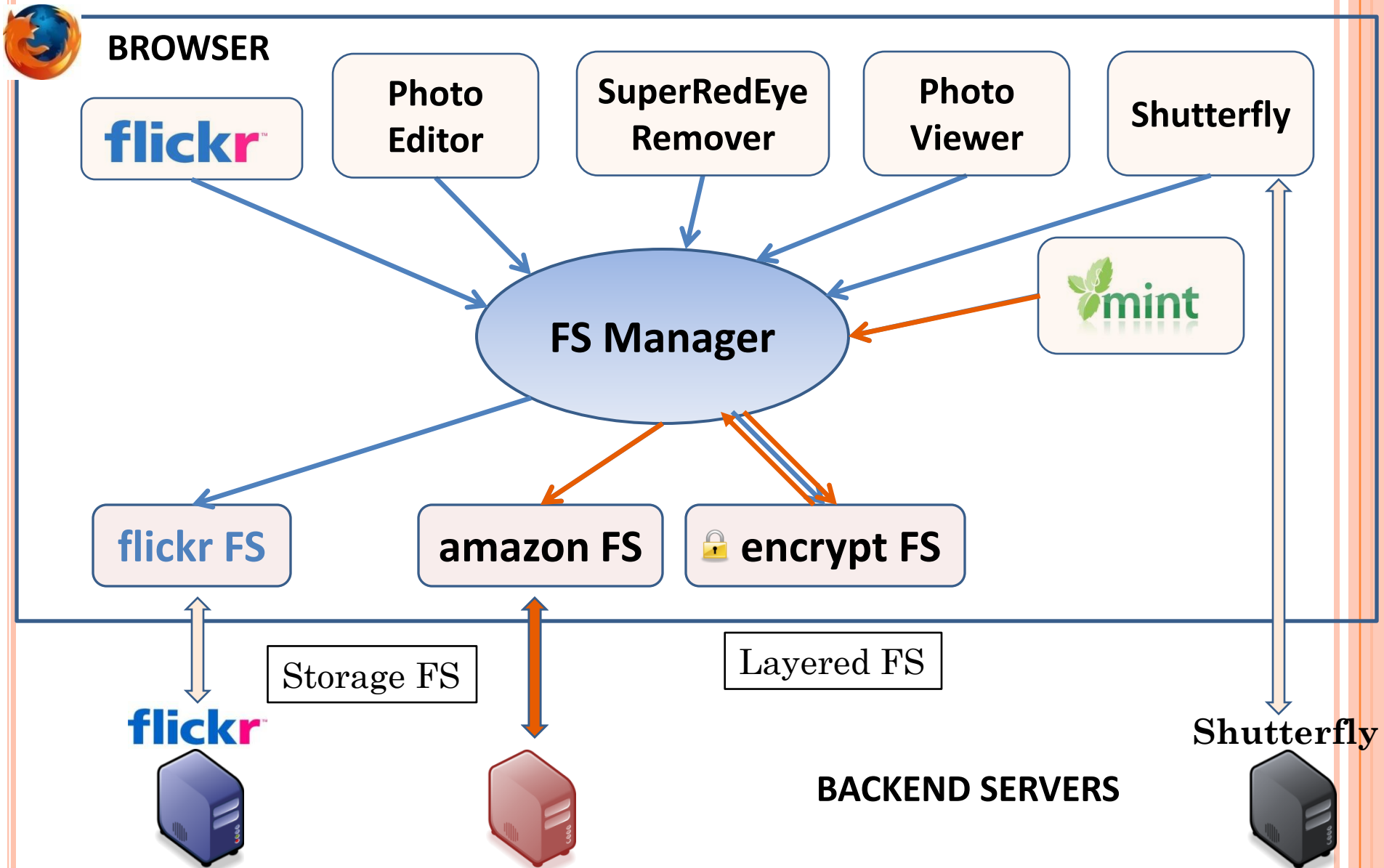
FILESYSTEM EXAMPLES



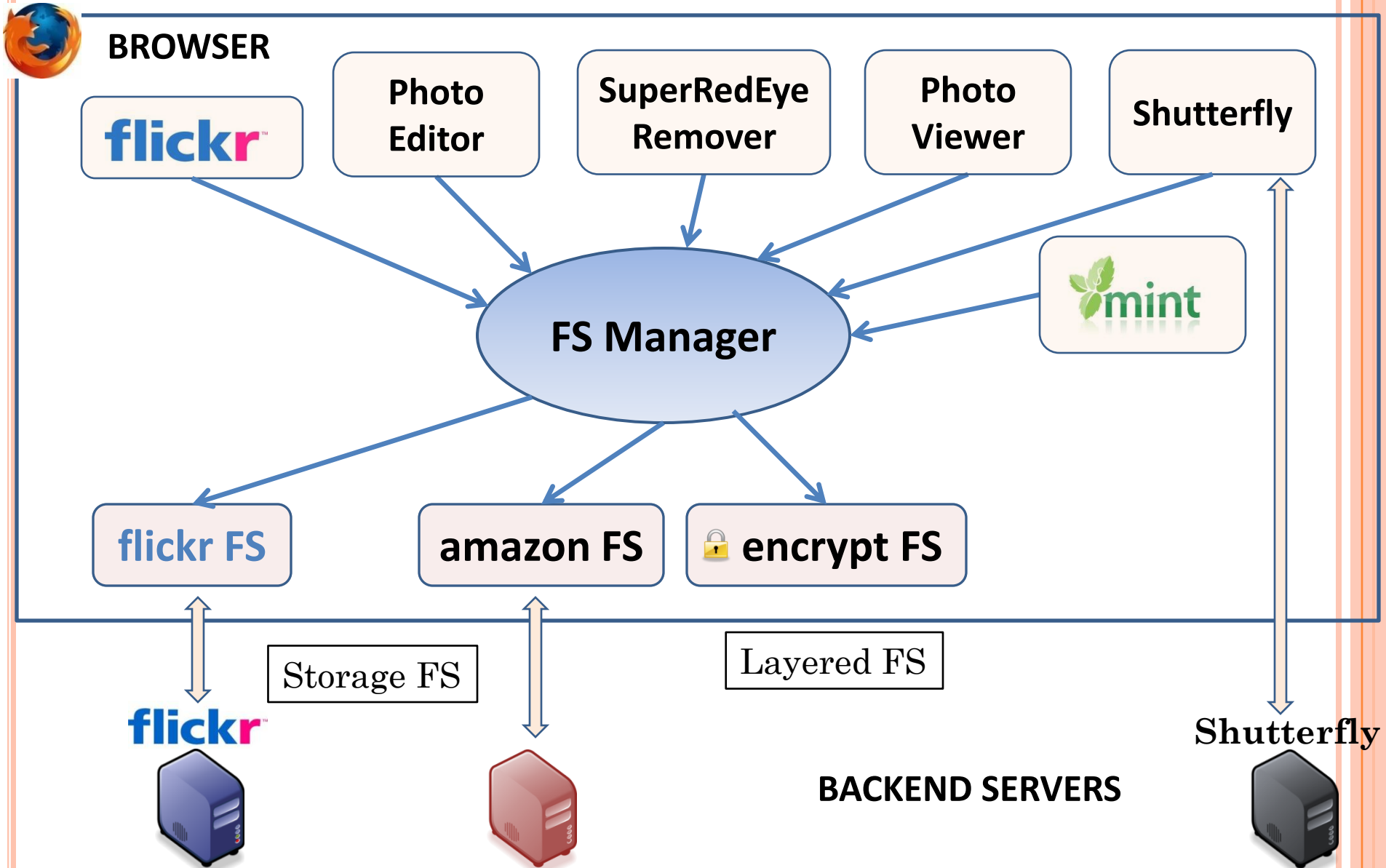
FILESYSTEM EXAMPLES



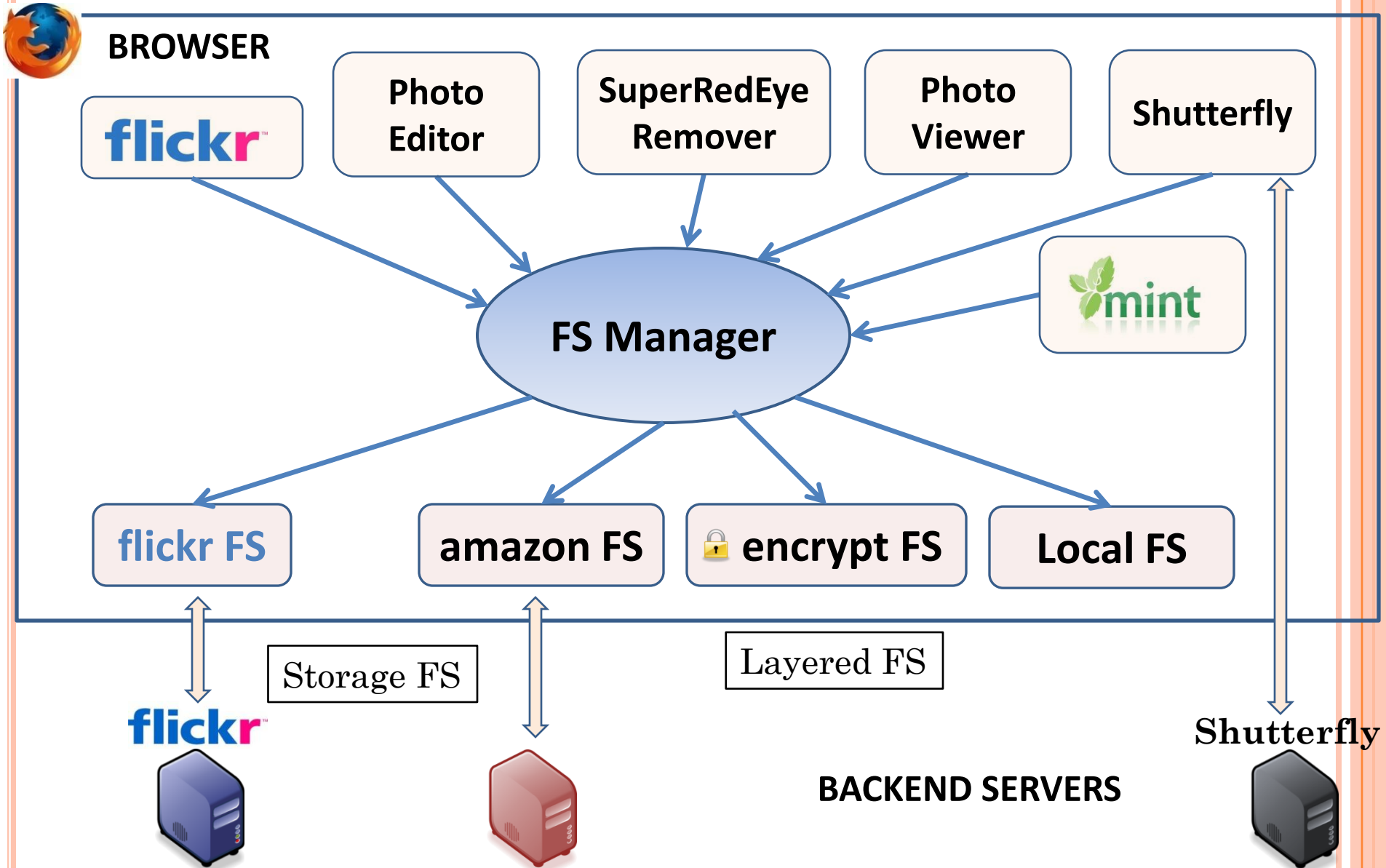
FILESYSTEM EXAMPLES



FILESYSTEM EXAMPLES



FILESYSTEM EXAMPLES



TALK OUTLINE

- Motivation
- Current solutions
- BSTORE overview
- Design details:
 - Filesystem API
 - FS organization with tags
 - Tag-based access control
 - Filesystem setup
- Implementation
- Evaluation
- Related work

IMPLEMENTATION

Component		LOC	
		JavaScript	PHP
Core Components	Util Library	1348	-
	FS Manager	617	-
	File Manager	337	-
Filesystems	StorageFS	58	712
	EncryptFS	239	-
	CheckPointFS	595	-

IMPLEMENTATION

Component		LOC	
		JavaScript	PHP
Core Components	Util Library	1348	-
	FS Manager	617	-
	File Manager	337	-
Filesystems	StorageFS	58	712
	EncryptFS	239	-
	CheckPointFS	595	-

- Browser-side Javascript components:
 - Communication between browser windows: `postMessage`
 - Supports Firefox and Google Chrome browsers
 - No browser modifications
- Storage FS:
 - PHP backend on Linux / Apache
 - Communication between JS and PHP: `XMLHttpRequest POST`

EVALUATION

- What is the effort to port apps to BSTORE?
- What is BSTORE's performance overhead?

EFFORT TO PORT APPS

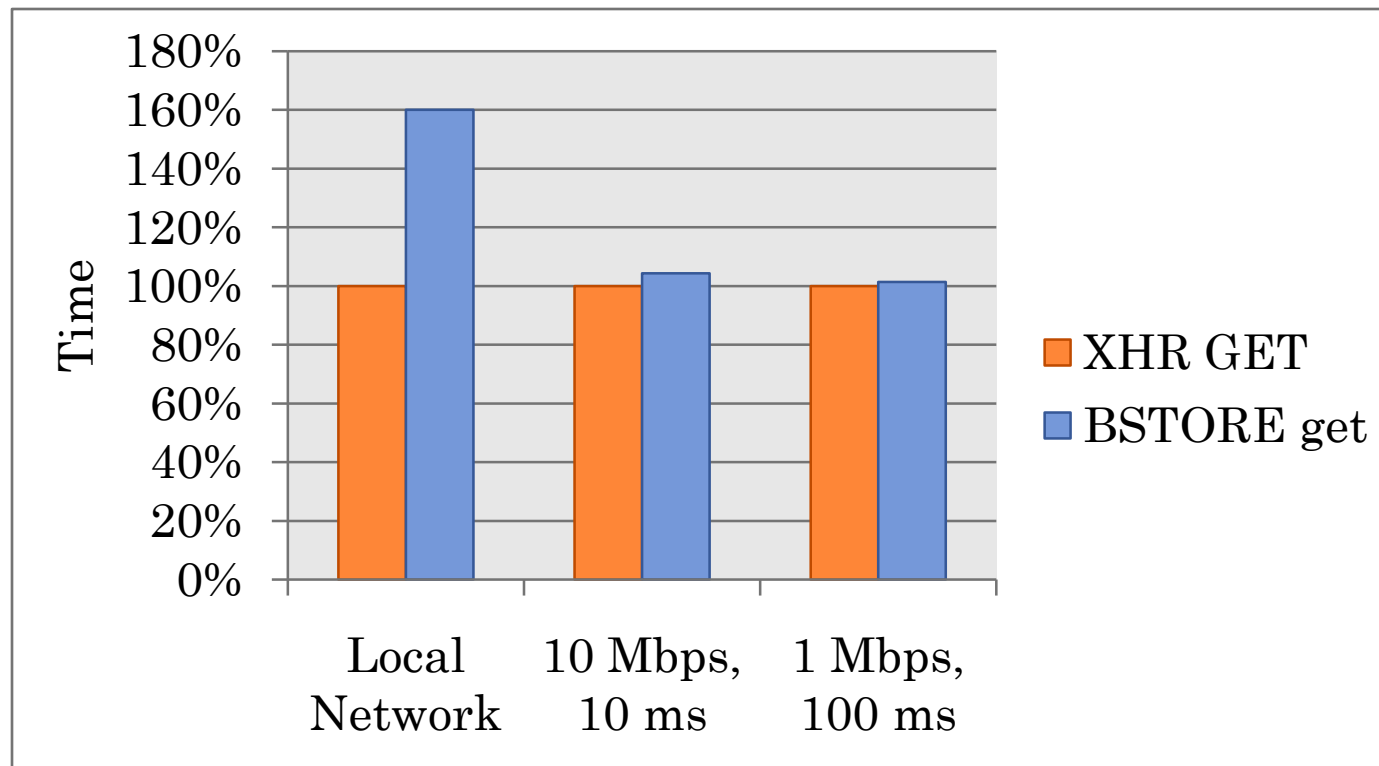
Application	Original LOC	Modified LOC	% Modified
Pixastic	4245	81	1.9%
vi	3471	74	2.1%
TrimSpreadsheet	1293	66	5.1%

- Less than a day per app
- Modifications are localized

PERFORMANCE EVALUATION

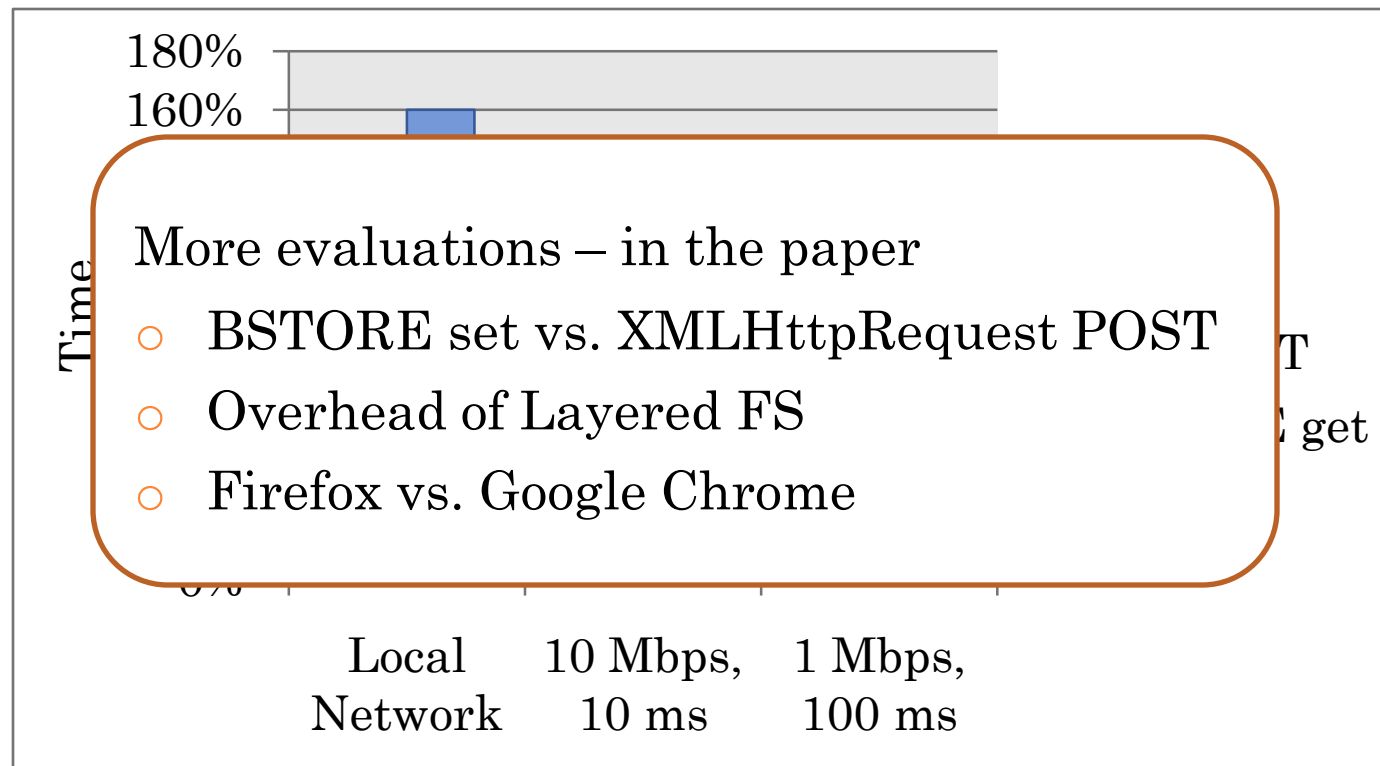
- Compare BSTORE get with XMLHttpRequest(XHR) GET
- Experimental setup:
 - Firefox browser on Intel Core i7 950 3.07GHz
 - Storage FS on Intel Xeon 3.06GHz
 - Network configurations:
 - Local network: 100Mbps Ethernet
 - Simulated wide area networks: 10Mbps/10ms, 1Mbps/100ms
 - File size: 1MB

LOW OVERHEAD FOR REMOTE SERVERS



- BSTORE overhead primarily in browser:
 - Encoding and decoding data
 - postMessage

LOW OVERHEAD FOR REMOTE SERVERS



- BSTORE overhead primarily in browser:
 - Encoding and decoding data
 - postMessage

LIMITATIONS

- No support for cross-user sharing
- Principals are limited by SOP to URL origin
- Don't support apps that:
 - Need a DB interface
 - Need server-side computation

RELATED WORK

- Server-side sharing:
 - Cannot support browser-only apps or offline usage
 - Pair-wise sharing
 - E.g: OAuth, REST APIs, Menagerie
- Client-side sharing:
 - Position paper by Hsu and Chen, with vision similar to BSTORE
- Tag-based file organization in other domains:
 - E.g: Semantic file systems, Presto

CONCLUSIONS

- Current web app storage model constrains users and app developers
- BSTORE solves these issues with:
 - Browser-based sharing
 - Common API
 - Tag-based rights delegation
- BSTORE works on unmodified browsers and has low overhead

CONCLUSIONS

- Current web app storage model constrains users and app developers
- BSTORE solves these issues with:
 - Browser-based sharing
 - Common API
 - Tag-based rights delegation
- BSTORE works on unmodified browsers and has low overhead

Thank you!