*35th NMRG meeting @ Windsor Atlântica*
*Rio de Janeiro, Brazil, 17th November 2014*

# PiCsMu: A Cloud Overlay to Store and Manage Data

**Guilherme Sperb Machado, Burkhard Stiller**
*Department of Informatics IFI, Communication Systems Group CSG, UZH*
*[machado|stiller]@ifi.uzh.ch*


Universität Zürich[UZH]

CSG
Communication Systems Group

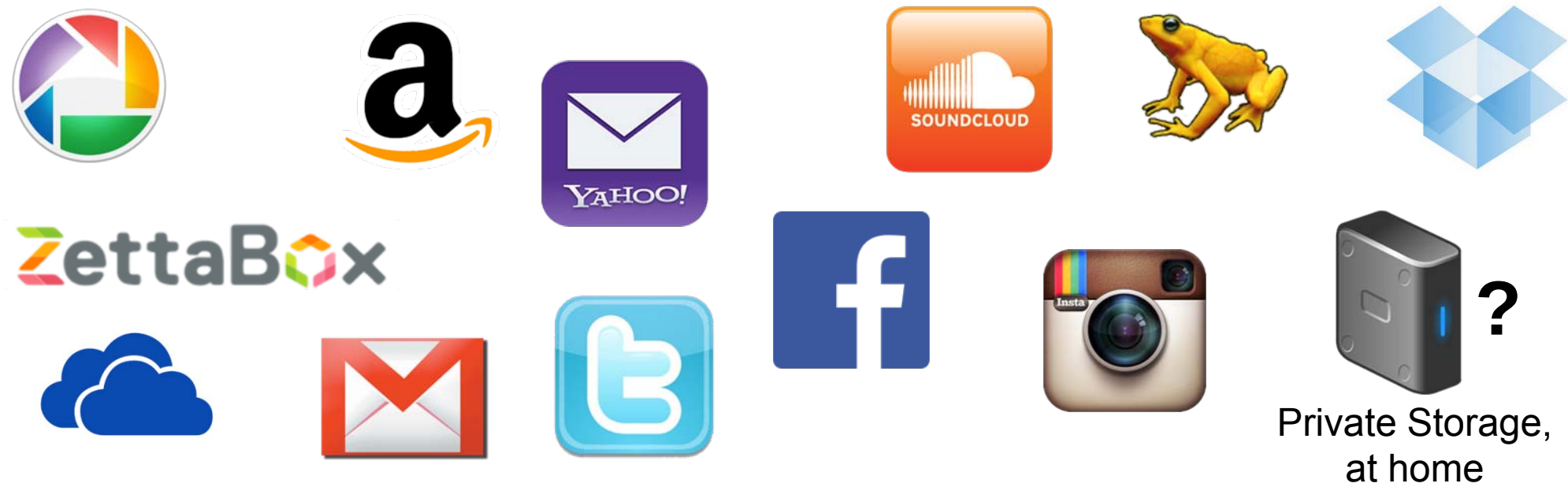


ifi

# Agenda

- Introduction and Motivation
- **PiCsMu** System
- Prototype Implementation
- Evaluation
- Related Work
- Summary and Conclusion

# Background




- Accessed via well-defined Application Programming Interface
- Services providing **2 types** of storage
  1. Generic storage services
  2. Data-specific storage services

# Introduction and Motivation

- **Generic** and **data-specific** Cloud Services
  - What do they have in common?
    - Data is stored on Cloud service's servers
  - What do they differ in?
    - APIs
    - Accounting and charging schemes
    - Privacy and security levels
    - Functionality and data type restrictions
- Idea and research questions:
  - Build an overlay to aggregate heterogeneous Cloud services' storage into a virtual, single, user-perceived storage
    - Is that possible? Is it scalable? What is the overhead?

# Overview

- **P**latform-**i**ndependent **C**loud **S**torage System for **M**ultiple **U**sage

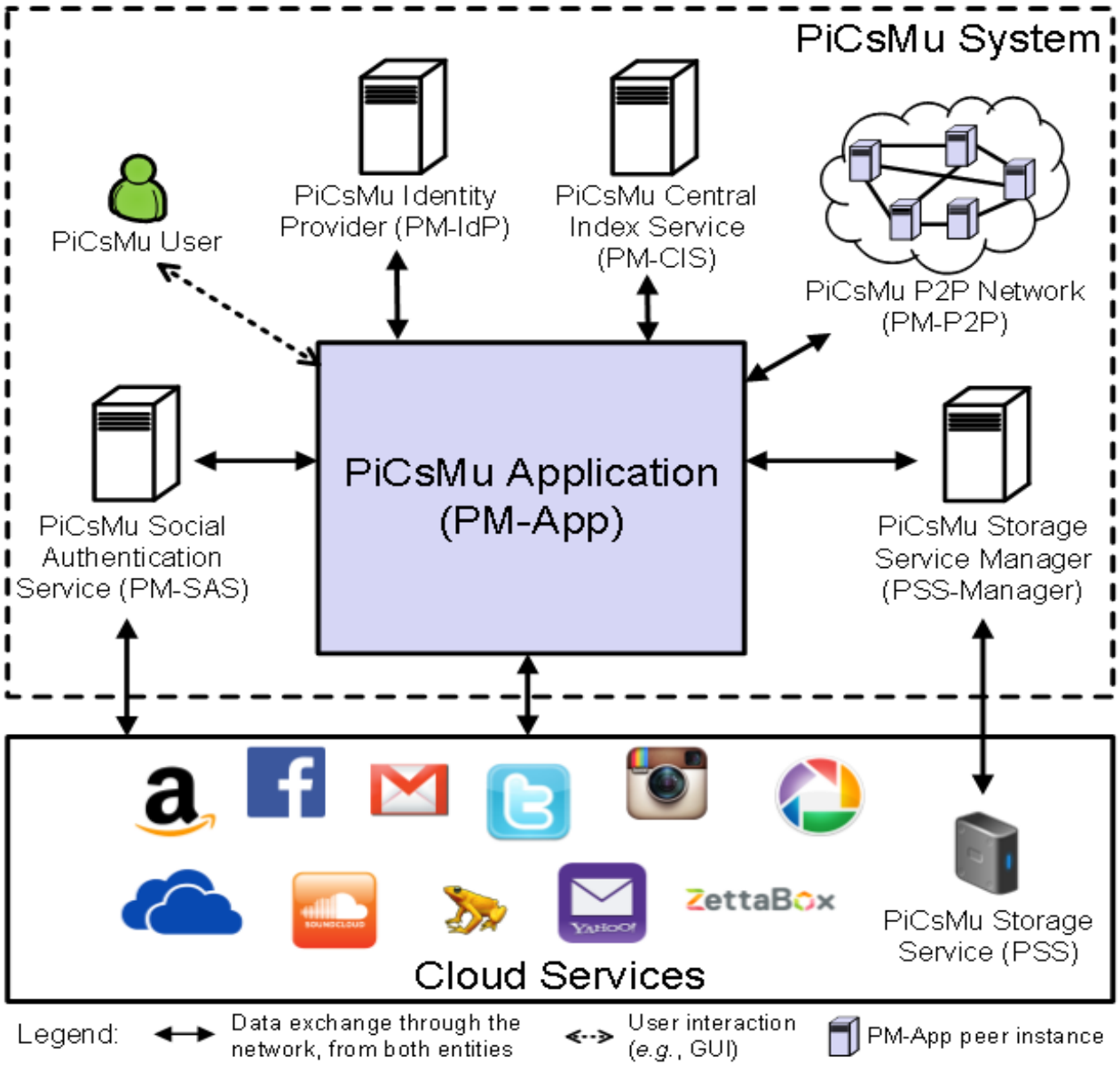
**Store private and share files**, by aggregating different Cloud services, with a certain level of confidentiality (privacy)

- PiCsMu **is**:
  - Overlay
  - Cloud Service's Aggregator
  - Storage Management System
  - Independent of allowed Cloud Service's data types: "store any file, in any Cloud service"
  - Hybrid approach
    - Centralized and Decentralized entities

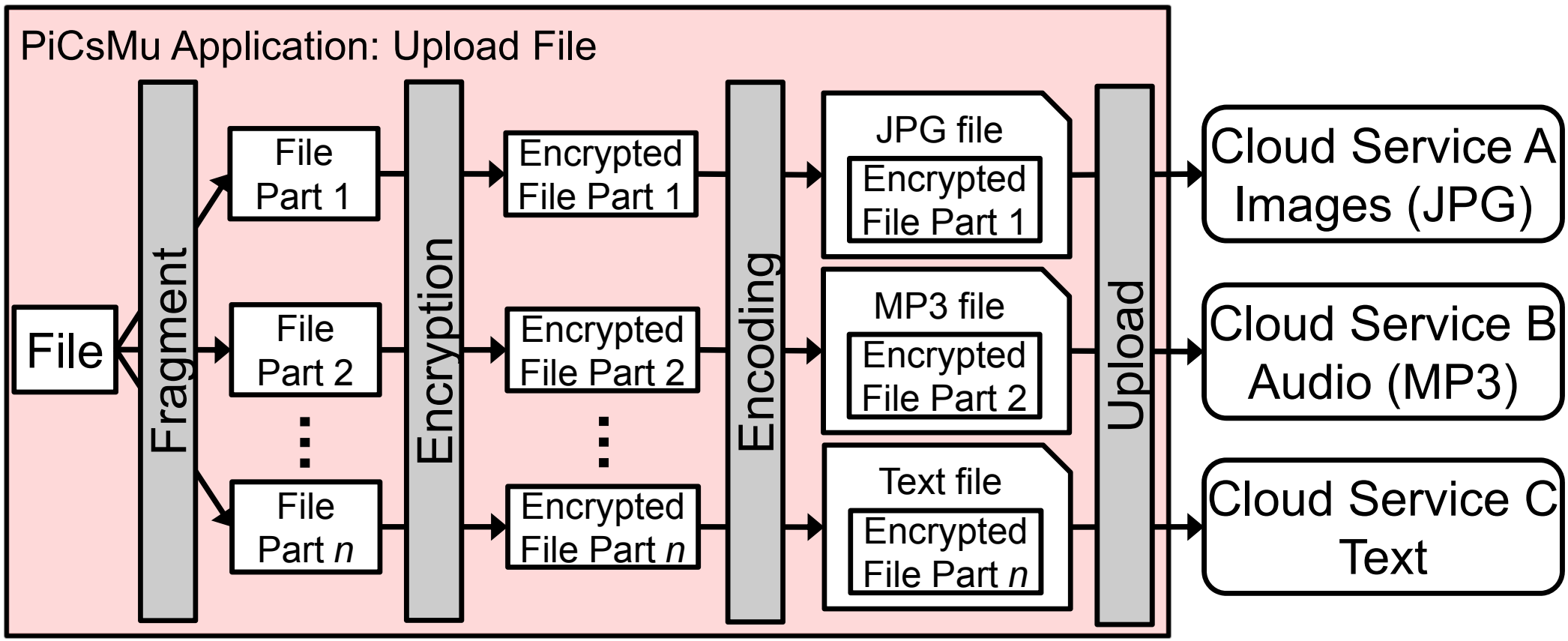
# Design Goals

- PiCsMu defines how to
  - Share files to everyone in the network, and to specific users
  - Upload personal files
  - Download personal and shared files
- Resulting in
  - **File upload and download processes definitions**
  - PiCsMu architecture and operation modes
- 3 modes of operation
  - Private storage
  - Private sharing
  - Public sharing

# System Architecture



# File Upload (1/2)

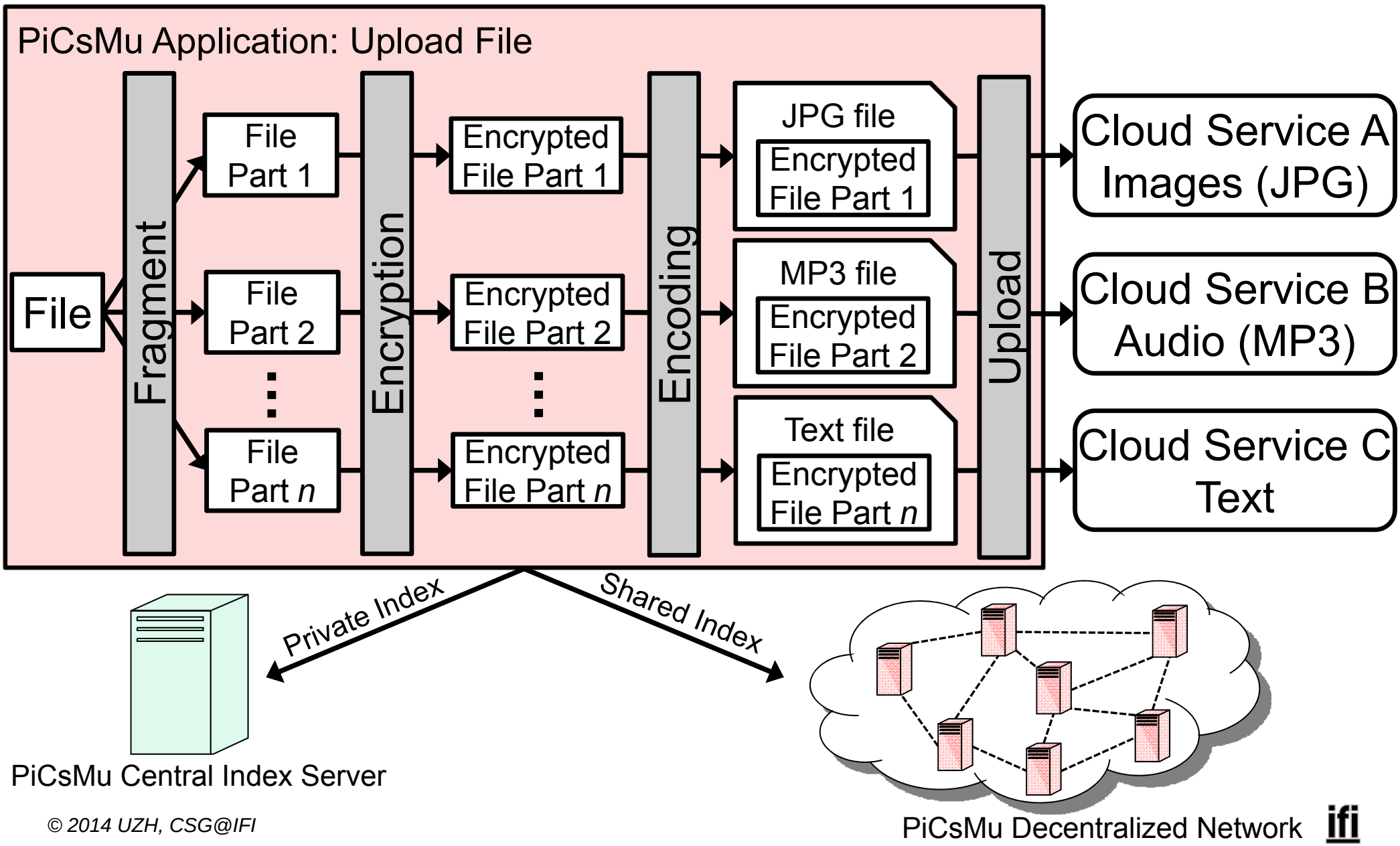



- Index:
  - Stores information about the PiCsMu upload process result

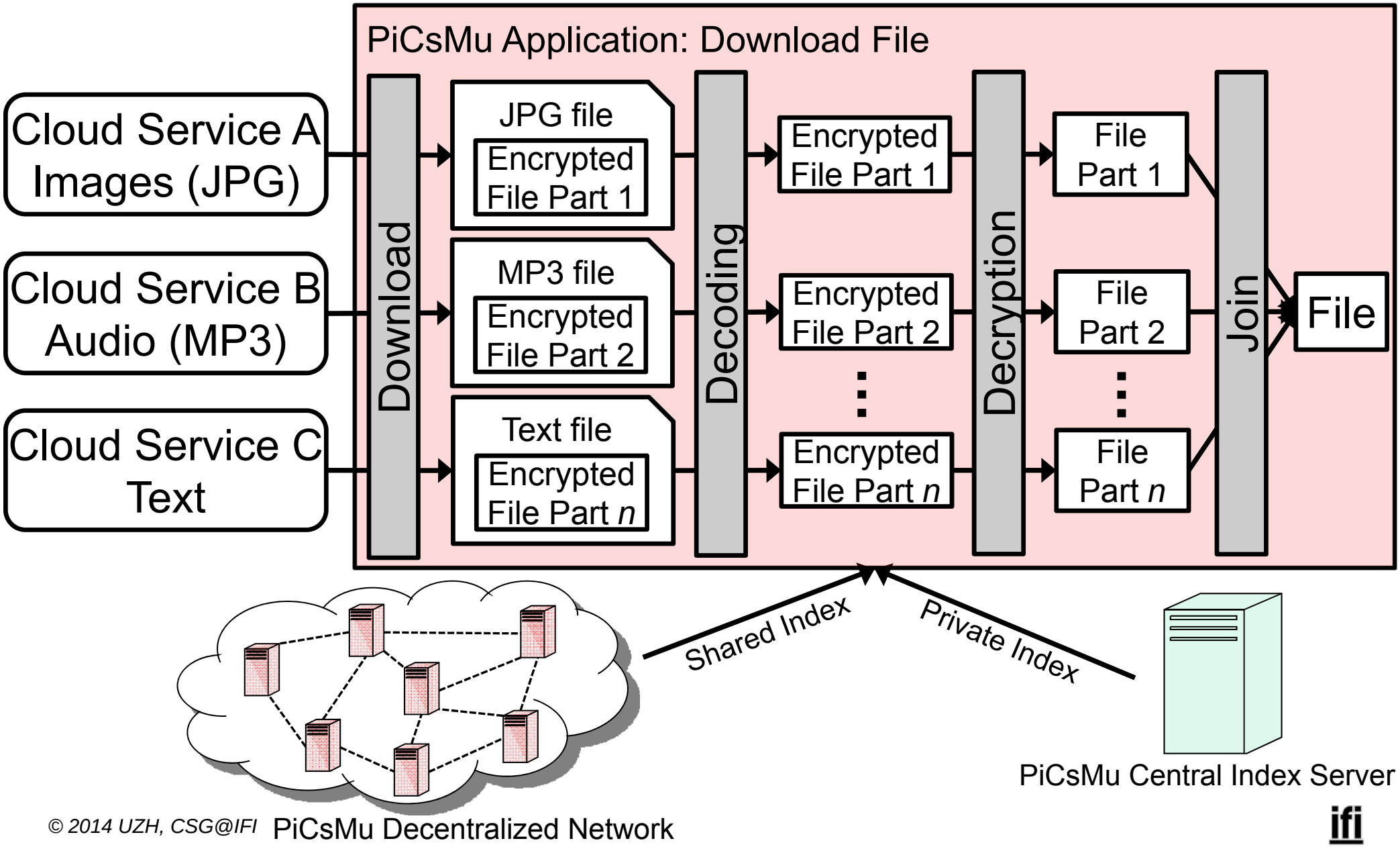
⇨

  - How was file fragmented?
    - Size, order
  - How were file parts encoded / encrypted?
    - Which encoder and parameters were used?
    - Embedded into file type JPG, MP3, or text
  - Where is a file part located?

# File Upload (2/2)

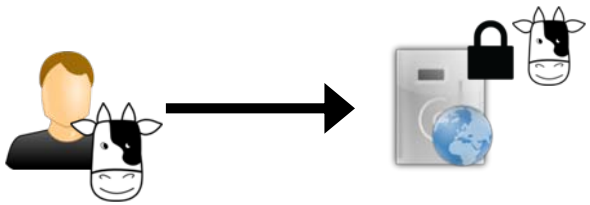


# File Download

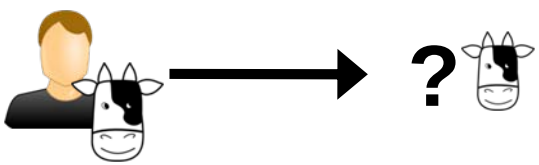


# Storage Modes

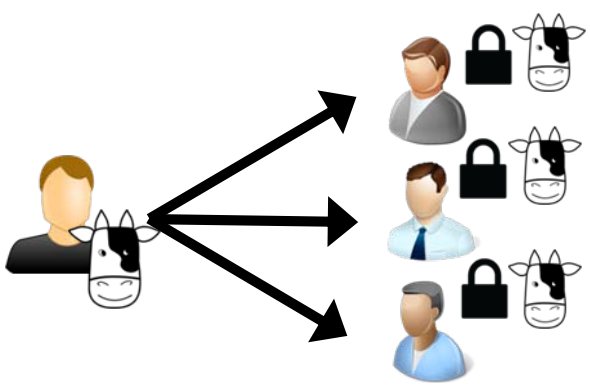
- Private Storage Mode (centralized)



- Public Sharing Mode (decentralized)




- Private Sharing Mode (decentralized)



# Prototype Implementation

- Java-based backend + GUI (Java Swing)
- PiCsMu Application implemented in *modules*

- Cloud Services (upload/download)


25 MByte emails


20 MByte images


63,206 text chars

60 min audio


5 MByte images

100 MByte files

20MByte files

- Data Encoders
- Encryption Strategies
- Scheduler
- Distributed Hash Table[1] and PiCsMu Central Index Server
  - Persist index information

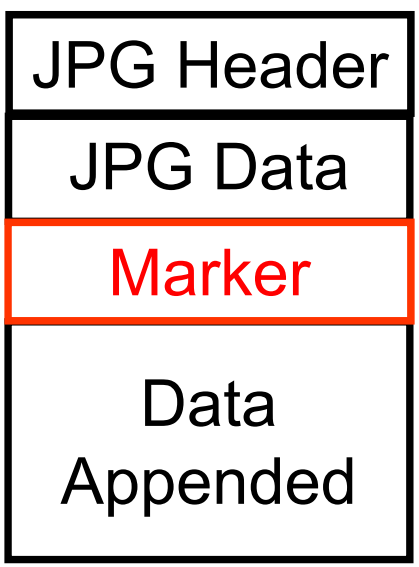


[1] TomP2P DHT Library: http://tomp2p.net

# Encoders (1)

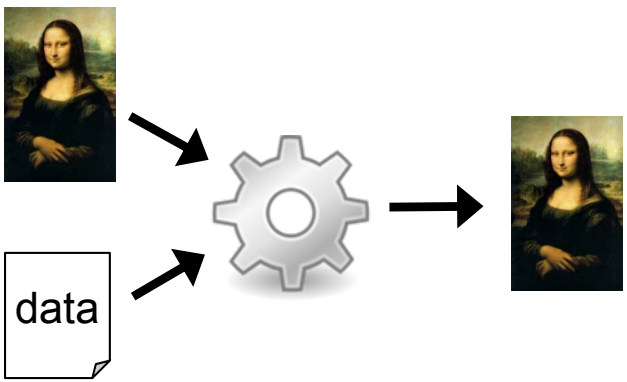
- Approach: Appender
  - Appends data to the end of a file
  - Uses a “marker” in order to separate original file format from additional data

- Example: JPG-represented Picture



# Encoders (2)

- Approach: Steganography




  - 3 different file types
    - JPG File Format
      - LSB (Least Significant Bit)
      - 3 bits of each image pixel contain data
    - Text
      - English dictionary to convert from binary data to “English words”
      - No semantic (yet)
    - MP3 and WAV
      - LSB (Least Significant Bit)
      - 3 bits (inaudible in WAV)

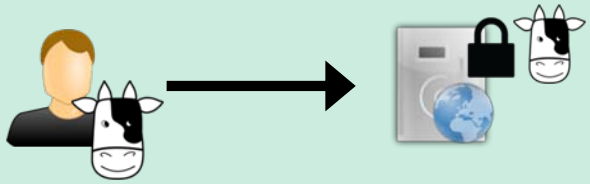
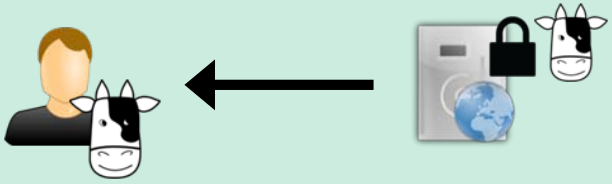
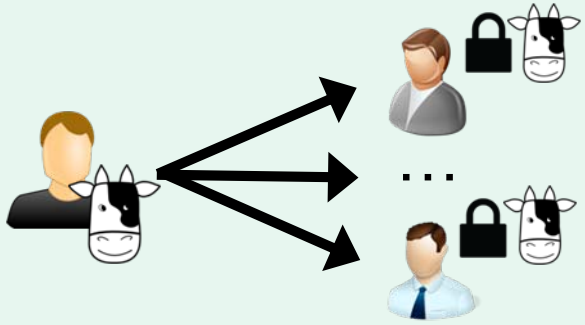
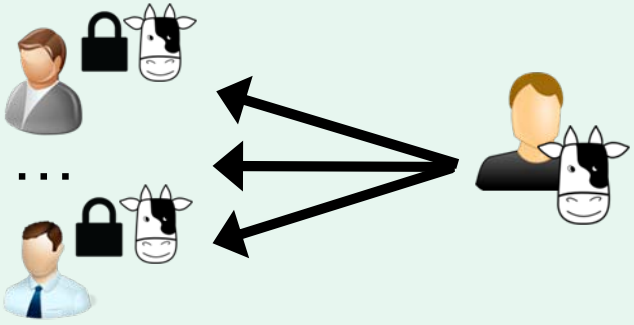
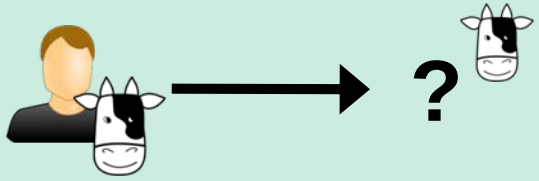
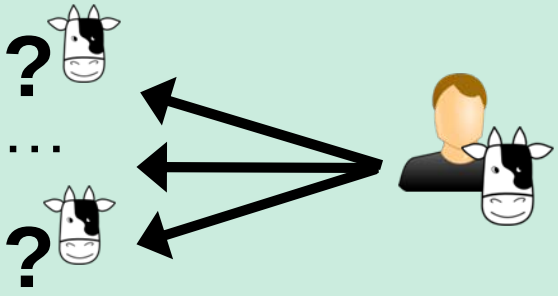
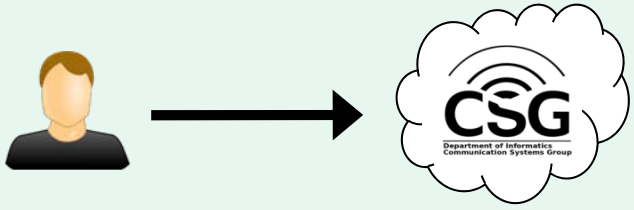
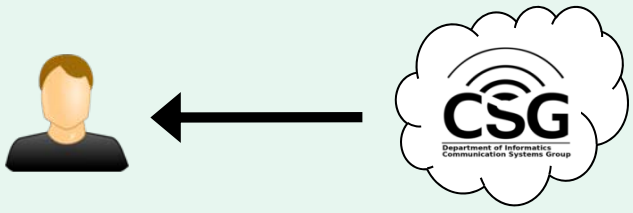
# Encoders (3)

- Approach: IDv3 Tag Encoder
  - Injects data into IDv3 Tags
  - 256 MByte of data, according to IDv3 Tag specification
  - Distributed in several attributes (*e.g.*, album, song title)
  - Injects as much data as possible following ID3v2 standard

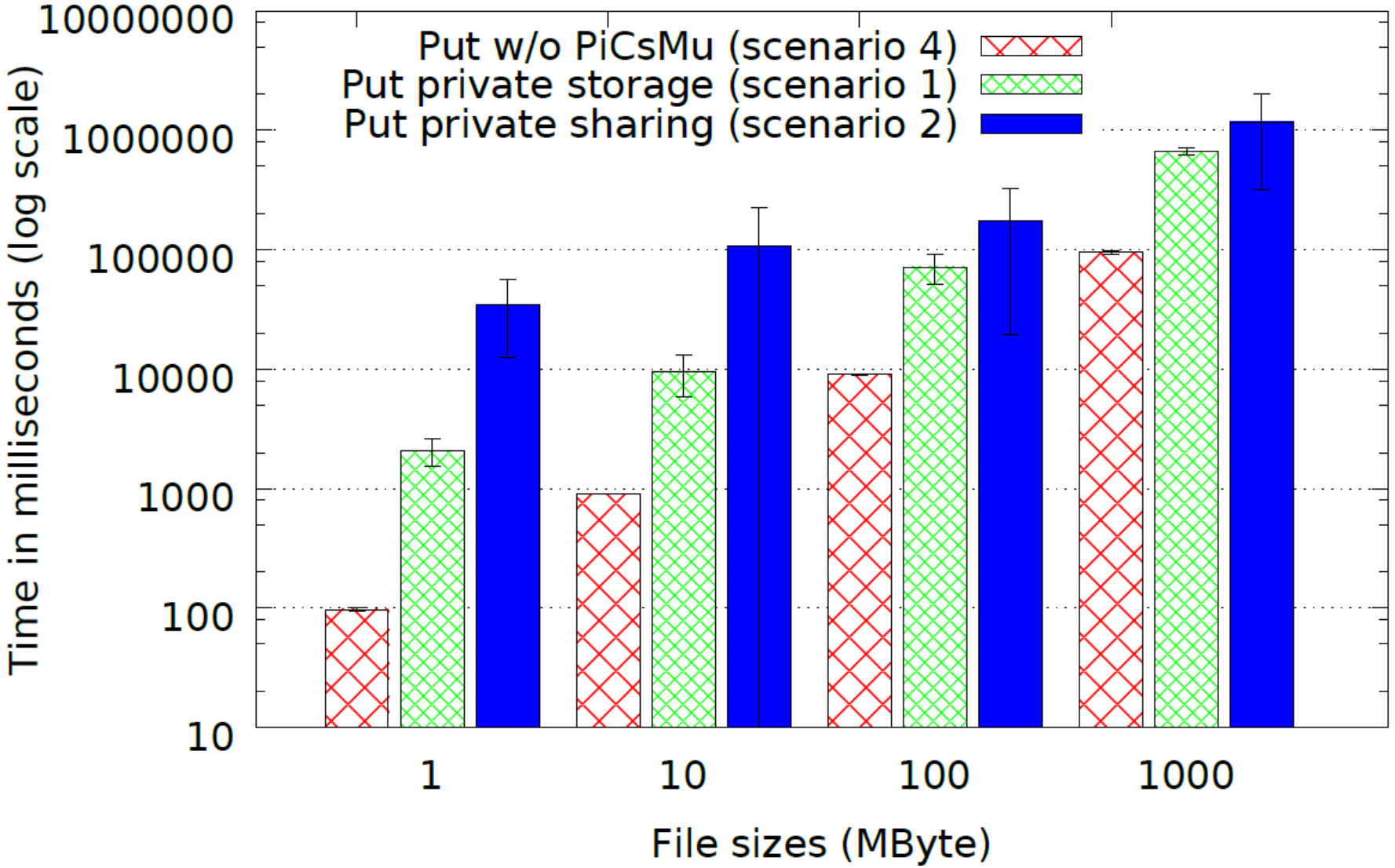
- Approach: Gmail Encoder
  - Data is represented in
    - Email body…
    - …and as an attachment

- Others Encoders, *e.g.*, JPG Header, PNG Header

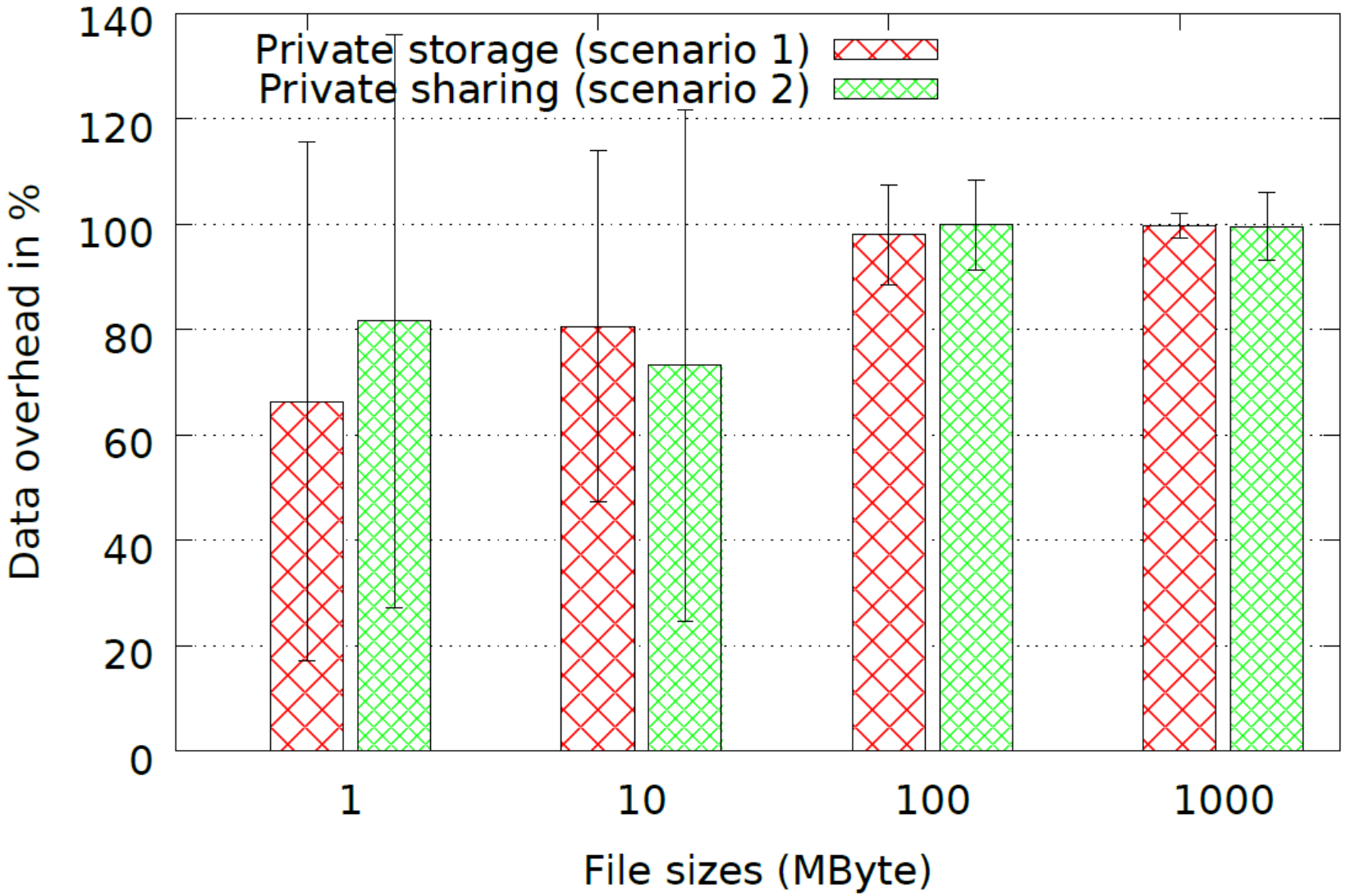
# Evaluation: Scenarios and Test Cases

| Scenarios | Upload (put) | Download (get) |
|---|---|---|
| (1) Private Storage Mode |  |  |
| (2) Private Sharing Mode |  |  |
| (3) Public Sharing Mode |  |  |
| (4) CSG Service ***without*** PiCsMu |  |  |

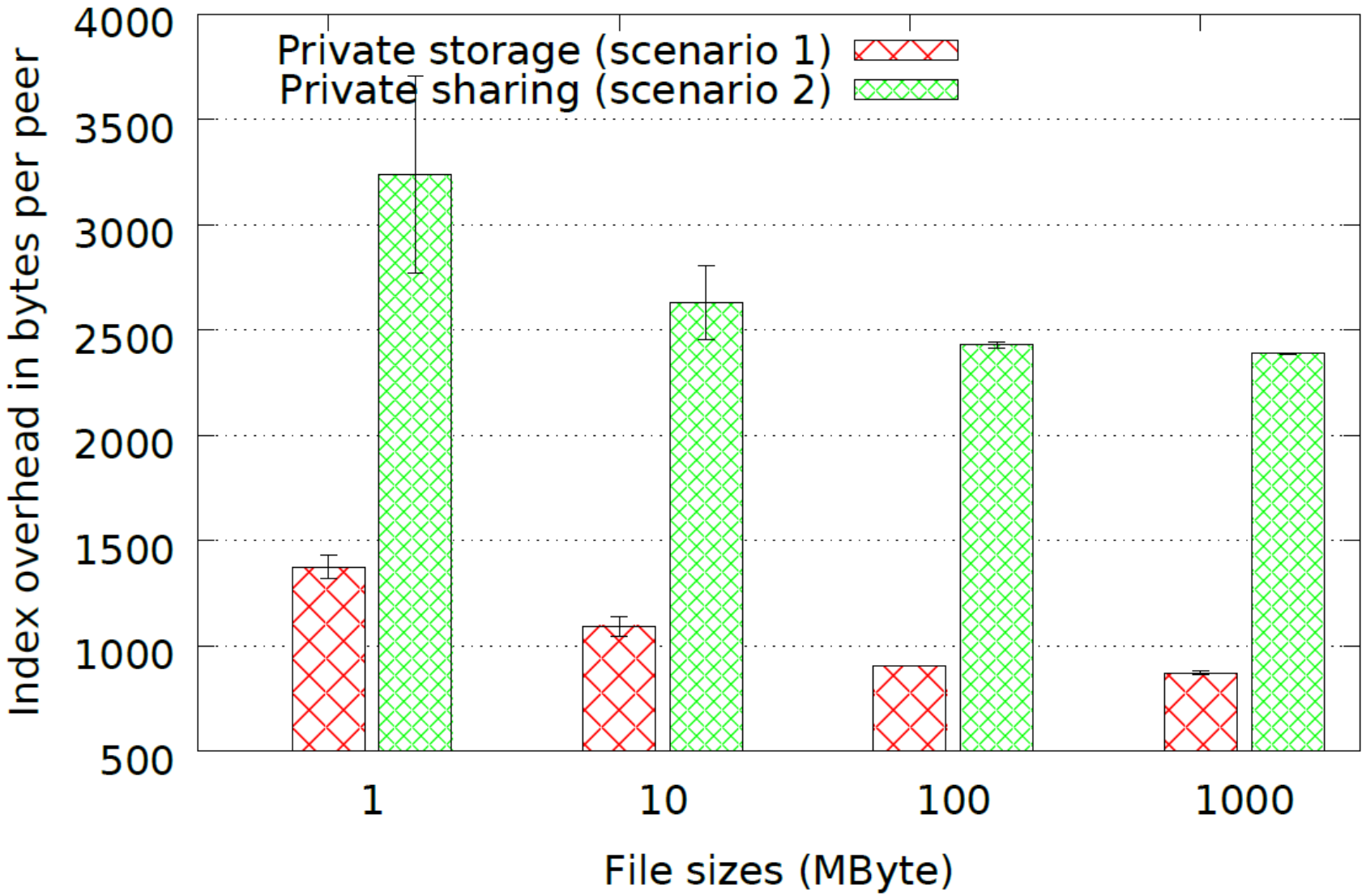
# Total Times for Upload



# File Data Overhead



# PiCsMu Index Overhead



# Storage Services

| | Overlay | Support Additional Services | Fragment to Multiple Clouds | Encrypt | Encode (Hide/Inject) | Share | |
|---|---|---|---|---|---|---|---|
| | | | | | | Centralized | Decentralized |
| Dropbox | no | no | no | no | no | yes | no |
| Google Drive | no | no | no | no | no | yes | no |
| Microsoft Skydrive | no | no | no | no | no | yes | no |
| SpiderOak | no | no | no | yes | no | yes | no |
| Wuala | no | no | no | yes | no | yes | no |
| Otixo | yes | yes | no | no (depends on the service) | no | yes | no |
| **PiCsMu** | **yes** | **yes** | **yes** | **yes** | **yes** | **yes** | **yes** |

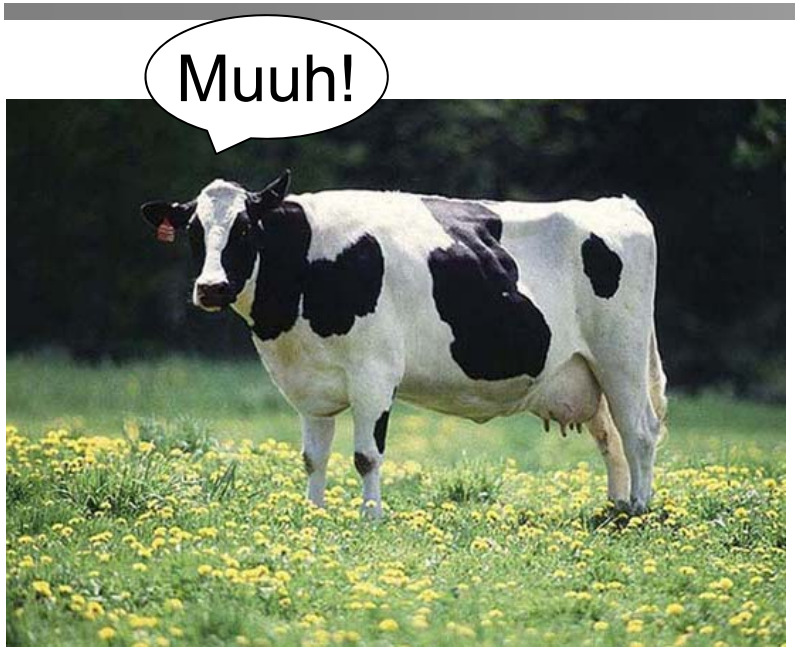
# Summary and Conclusion

- Novel storage overlay aggregating heterogeneous Cloud storage services
  - Supports generic and data-specific storage service types
  - Provides a single interface to end-users enabling storage and data sharing
  - Increased security and privacy

    → Turning it harder to gain access to content of original files

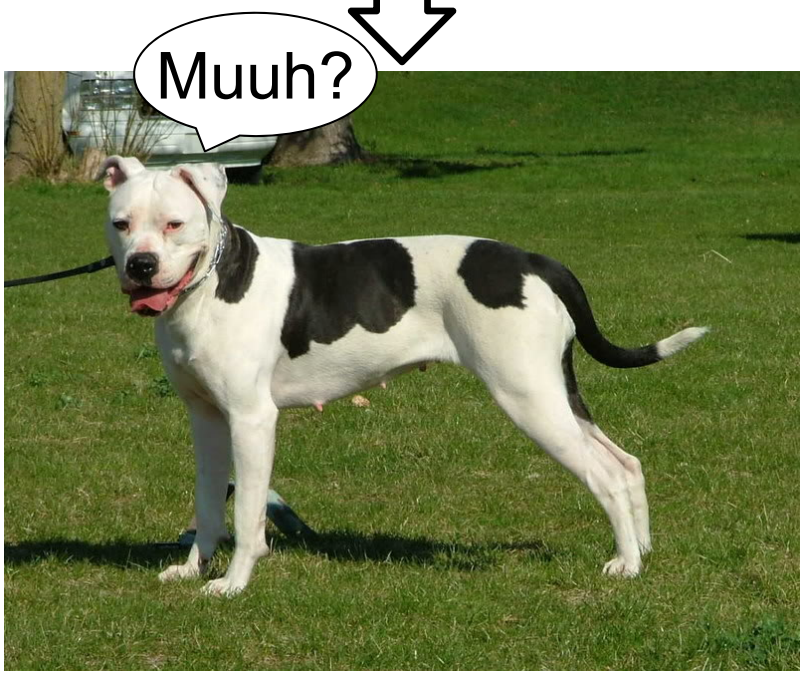
- Questions asked:
  - **It is possible**
  - **It is scalable**
  - **Moderate Overhead**

# PiCsMu in Brief ☺




Plain data stored in the Cloud.

Fragmentation + Encryption +
Encoding + Storage in multiple Clouds




Data stored in the Cloud using PiCsMu.

**Thank you!**