Facilitating Communal Data Sharing in Public Clouds

Roxana Geambasu
Steve Gribble
Hank Levy

University of Washington

Outline

- Vision: cloud as a platform for sharing code and data
- Why now: favorable cloud technology trends
- CloudViews: convenient, scalable, and efficient data sharing in public clouds

Outline

- Vision: cloud as a platform for sharing code and data
- Why now: favorable cloud technology trends
- CloudViews: convenient, scalable, and efficient data sharing in public clouds

The Web's Move to Public Clouds

Private datacenters

Web service

Web service

Web service

Web service

Public clouds

(AWS, AppEngine, Azure)

Web service

Web service

Web service

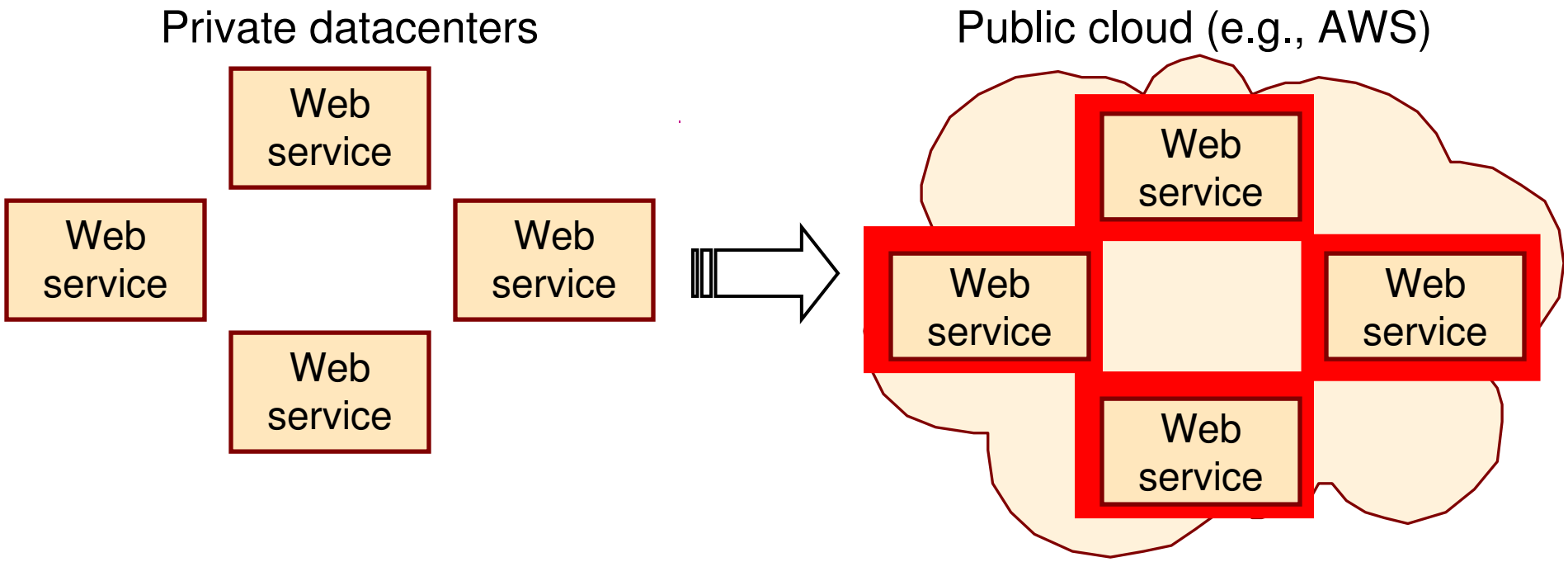
Web service

E.g.: SmugMug, Xignite, Techout, JungleDisk

The Current Perspective

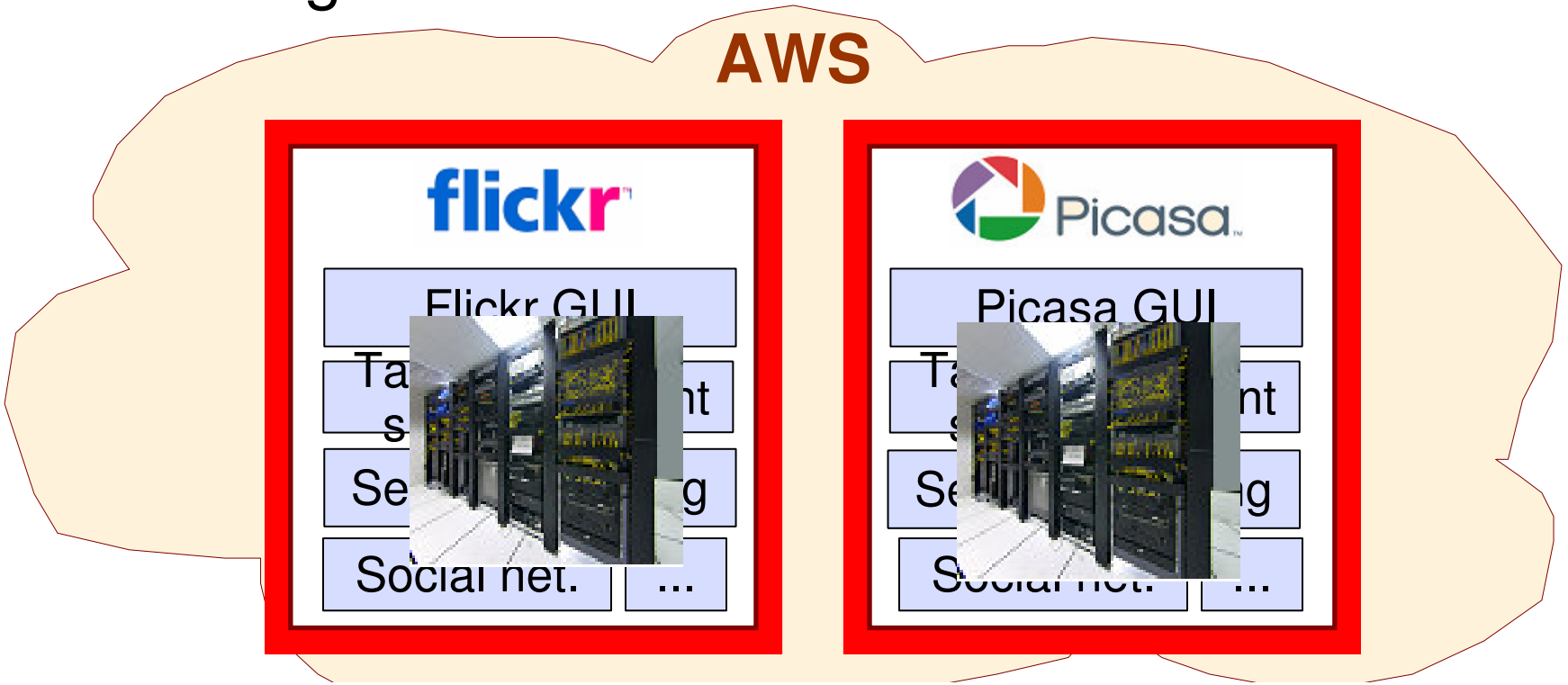
Top concerns have been to:

- Facilitate transition of individual Web services
- Isolate the Web services?



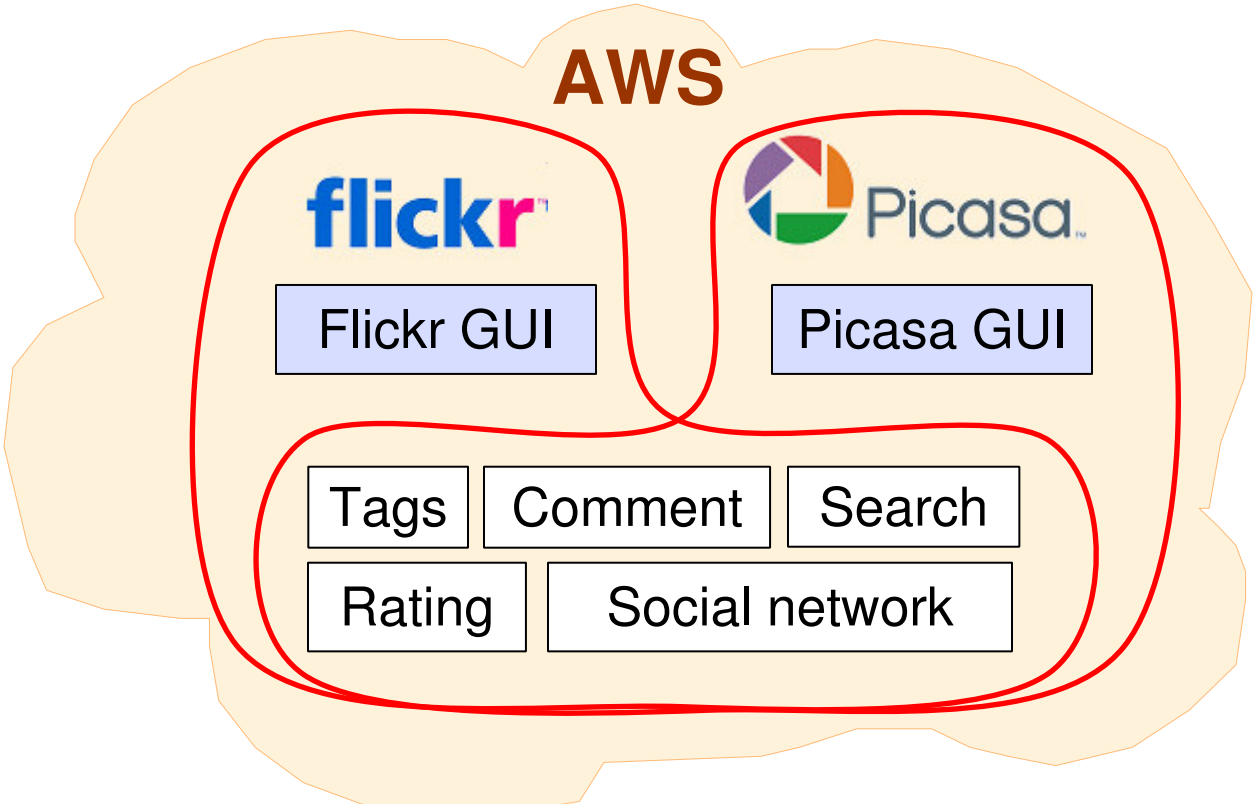
Isolation Leads To Stovepiping

- Web services are siloed
  - Each service implements the entire software stack
  - Many functions are common
- Building scalable services is hard even in the cloud



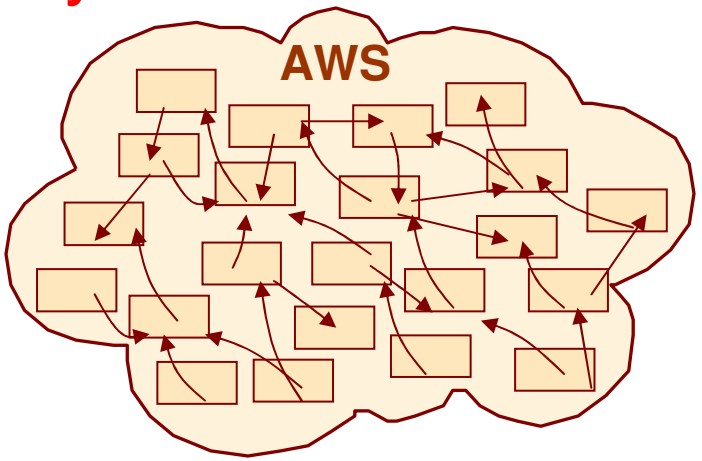
Our Perspective: Cloud as Sharing Platform

- Tens of thousands of co-located Web services
  - Most of the Web might be served from a few clouds
- What if some services rented themselves to others?



Our Vision

- Efficient, scalable service composition should be a primary function in public clouds
- Foresee a rich ecosystem of "utility services"
  - Examples from today: S3, SQS, Map/Reduce; RightScale



- Creating a large-scale service will be as easy as:
  - pick utility services;
  - write scripts to combine them; and
  - add service-specific logic (e.g., GUI).

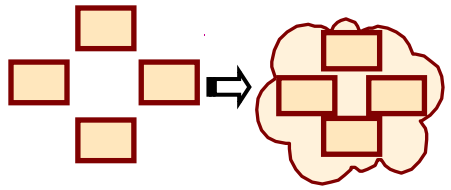
Supporting Composition in Public Clouds

- Lots of challenges:
  - Programming model
  - Efficient and scalable inter-service communication
  - Auditing computation (e.g., for billing)
  - Diagnosing problems in service chains
  - Service-level agreements
  - ...
- This talk addresses one vital type of composition: data-driven composition

Outline

- Vision: cloud as a platform for sharing code and data
- Why now: favorable cloud technology trends
- CloudViews: convenient, scalable, and efficient data sharing in public clouds

Favorable Cloud Tech. Trends

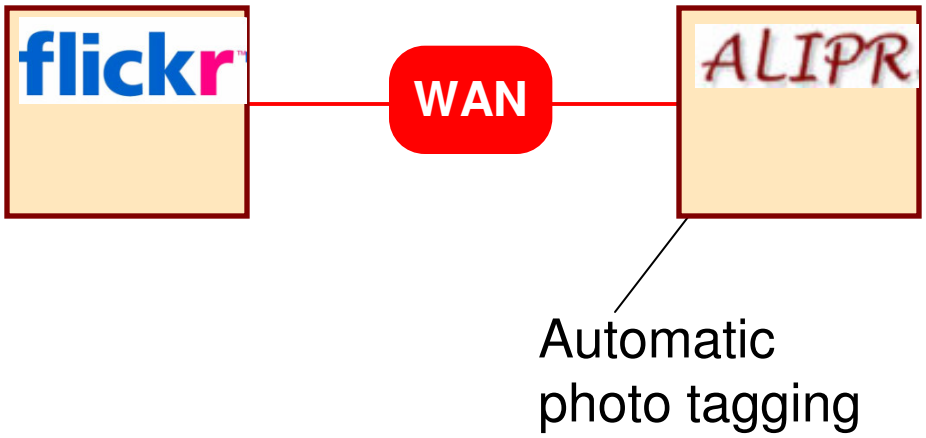


- Sharing was argued for in private-datacenter Web
  - E.g., Web 2.0 mashups, service-oriented architecture

- Two technology features make public clouds ideal for data sharing:
  1. A cheap, high-performance network
  2. A common database

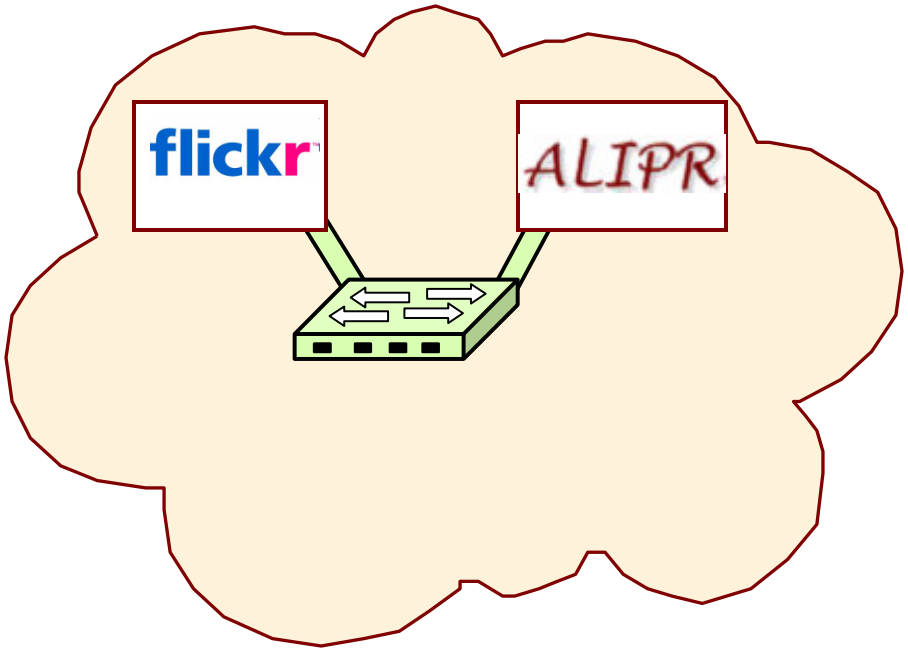
1. The Free and Fast Network

Private datacenters



Expensive, slow
inter-service network

Public cloud (e.g., AWS)

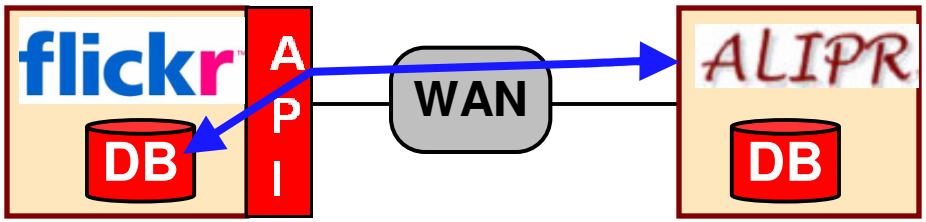


Free, high-speed
parallel network

Opportunity: large-scale, low-delay data sharing for free

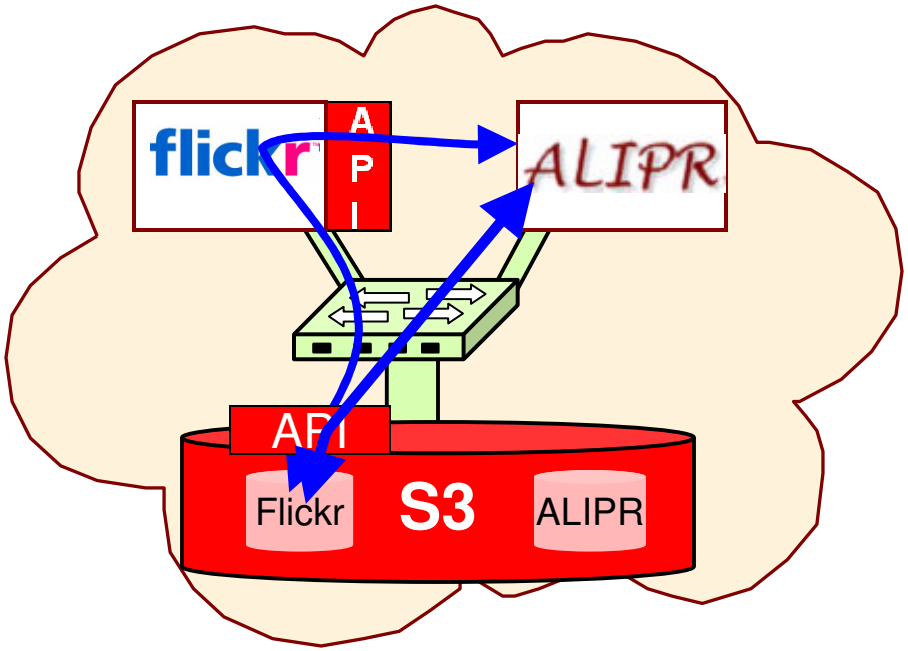
2. The Common Database

Private datacenters



Each service must provide & manage APIs

Public cloud (e.g., AWS)



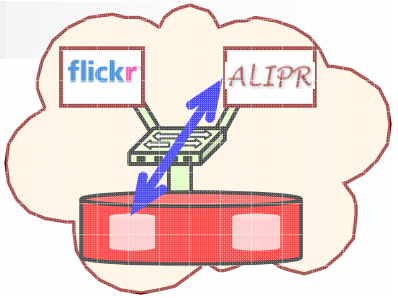
Common DB can handle data sharing

Opportunity: convenient, effortless data sharing

Outline

- Vision: cloud as a platform for sharing code and data
- Why now: favorable cloud technology trends
- CloudViews: convenient, scalable, and efficient data sharing in public clouds

Motivation

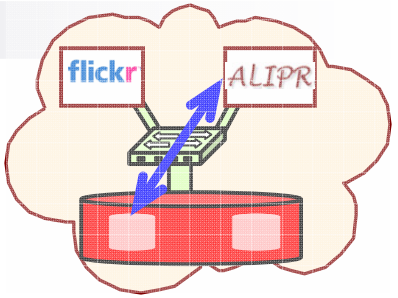


Today's clouds not designed for this type of sharing

- Inappropriate data sharing abstractions
  - E.g., buckets in S3, column families in Bigtable
- Limiting protection mechanisms
  - E.g., ACL sizes in S3 are limited to 100
- Resource allocation when sharing is involved
  - Rely on data partitioning for performance isolation

- What would the DB look like if designed for sharing?

CloudViews



Goal:

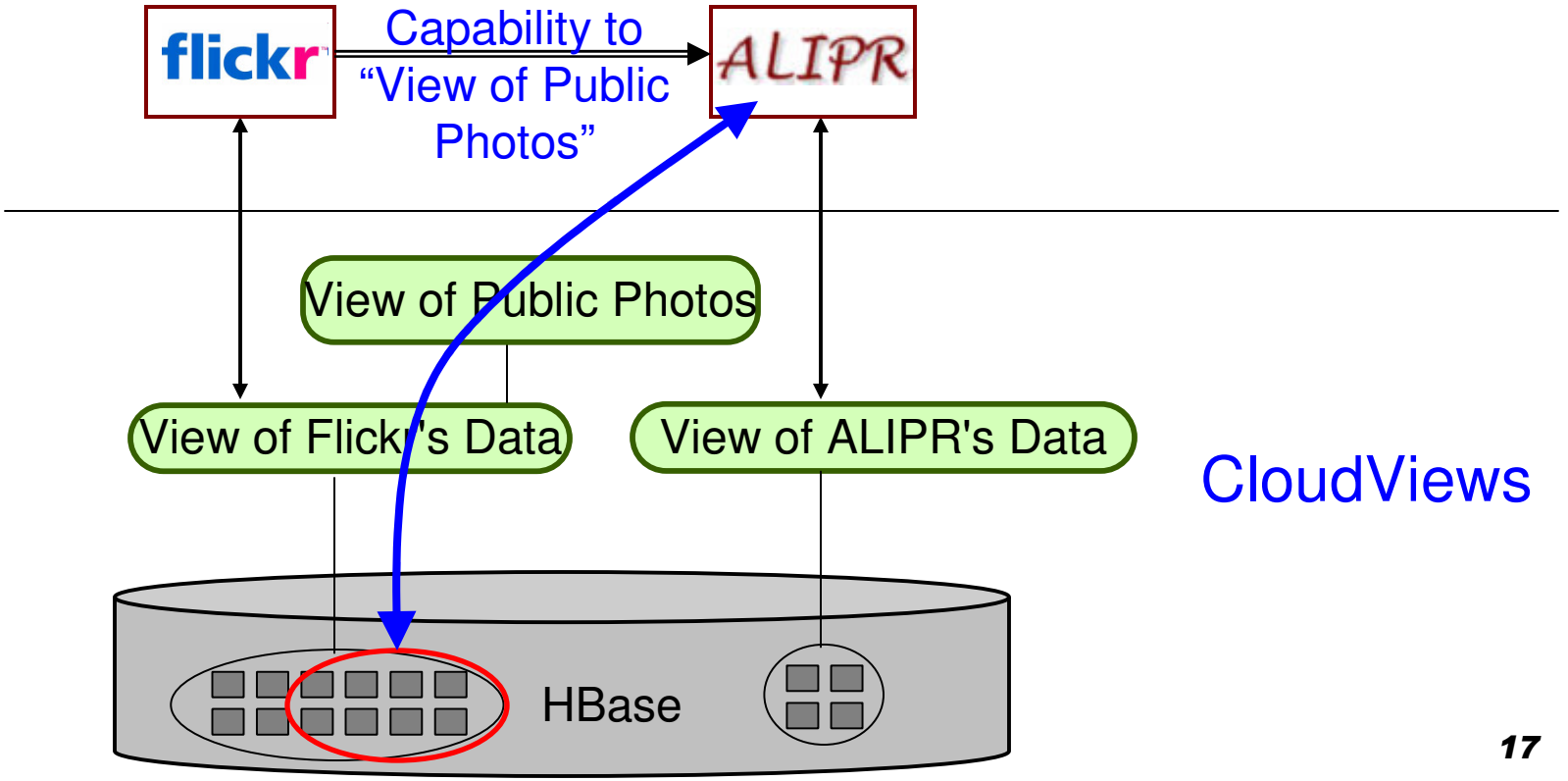
- Leverage cloud trends to facilitate scalable, efficient, protected data sharing

Requirements:

- Flexible and scalable sharing abstraction
  - Must allow expressing of service APIs
- Scalable protection mechanism
  - 10,000s services sharing data with each other
- Fair resource allocation for queries on shared data

CloudViews Overview

- Enhanced DB-style views for sharing
- Capabilities for protection
- Query admission control and QoS for resource allocation



Conclusions

- Today's clouds focus on single services and isolation
- Clouds should nurture large-scale data and code sharing
  - Opens great opportunities for simplifying service creation
  - Enables a rich ecosystem of "utility services" of the future
  - Supported by technology trends
- CloudViews: design cloud DB to take advantage of cloud technologies to support sharing
  - Supports convenient, large-scale, efficient data sharing