Cloud Computing

Ennan Zhai

Computer Science at Yale University
ennan.zhai@yale.edu
Lecture Outline

• Cloud Computing
• Challenges in the Clouds
• A Concrete Cloud Reliability Case
Lecture Outline

- Cloud Computing
- Challenges in the Clouds
- A Concrete Cloud Reliability Case
What’s the Cloud Computing?
Cloud computing is a business model for enabling convenient network access to a shared pool of configurable resources which can be rapidly provisioned and released with minimal management effort or service provider interaction.

--- according to NIST (National Institute of Standards and Technology)
Have You Used the Cloud?
Have You Used the Cloud?
Have You Used the Cloud?
Have You Used the Cloud?
Have You Used the Cloud?
Have You Used the Cloud?
Why We Like It?
Why We Like It?

• Why users like it?

• Why CS researchers like it?
Why We Like It?

• Why users like it?
  - Do not care where it is, it is “just there”
  - Access from “any” platform

• Why CS researchers like it?
Why We Like It?

- Why users like it?
  - Do not care where it is, it is “just there”
  - Access from “any” platform

- Why CS researchers like it?
  - High-performance computation with less money
  - Lots of hard and interesting challenges
What Kinds of Clouds Exist Now?
What Kinds of Clouds Exist Now?

- Three types of services:
What Kinds of Clouds Exist Now?

- Three types of services:
  - Infrastructure as a Service (IaaS)
    - Analogy: Grocery store. Provides raw ingredients.
What Kinds of Clouds Exist Now?

• Three types of services:

- **Platform as a Service (PaaS)**
  - Analogy: Take-out food. Prepares meal but does not serve it.

- **Infrastructure as a Service (IaaS)**
  - Analogy: Grocery store. Provides raw ingredients.
What Kinds of Clouds Exist Now?

- Three types of services:
  - Software as a Service (SaaS)
    - Analogy: Restaurant. Prepares & serves entire meal, does the dishes, etc.
  - Platform as a Service (PaaS)
    - Analogy: Take-out food. Prepares meal but does not serve it.
  - Infrastructure as a Service (IaaS)
    - Analogy: Grocery store. Provides raw ingredients.
Software as a Service (SaaS)
Software as a Service (SaaS)

Cloud Provider (i.e., SaaS Provider)

- Application
- Middleware
- Hardware
Software as a Service (SaaS)

• SaaS provider offers an entire application

Cloud Provider (i.e., SaaS Provider)
Software as a Service (SaaS)

- SaaS provider offers an entire application
  - Word processor, spreadsheet, CRM software, etc.
Software as a Service (SaaS)

- **SaaS provider offers an entire application**
  - Word processor, spreadsheet, CRM software, etc.
  - Customer pays cloud provider and uses the service
Software as a Service (SaaS)

- SaaS provider offers an entire application
  - Word processor, spreadsheet, CRM software, etc.
  - Customer pays cloud provider and uses the service
Software as a Service (SaaS)

- **SaaS provider** offers an entire application
  - Word processor, spreadsheet, CRM software, etc.
  - Customer pays cloud provider and uses the service
  - Example: Google Apps, Salesforce.com, etc.
A Typical SaaS: Gmail
A Typical SaaS: Gmail

Gmail Provider

- Application
- Middleware
- Hardware
A Typical SaaS: Gmail

• Outsourcing your e-mail software:
  - Distributed, replicated message store in BigTable
A Typical SaaS: Gmail

- Outsourcing your e-mail software:
  - Distributed, replicated message store in BigTable
A Typical SaaS: Gmail

• Outsourcing your e-mail software:
  - Distributed, replicated message store in BigTable
A Typical SaaS: Gmail

• Outsourcing your e-mail software:
  - Distributed, replicated message store in BigTable
A Typical SaaS: Gmail

- Outsourcing your e-mail software:
  - Distributed, replicated message store in BigTable
A Typical SaaS: Gmail

- Outsourcing your e-mail software:
  - Distributed, replicated message store in BigTable
A Typical SaaS: Gmail

- Outsourcing your e-mail software:
  - Distributed, replicated message store in BigTable
  - Weak consistency model for some operations (e.g., msg read)
A Typical SaaS: Gmail

Outsourcing your e-mail software:

- Distributed, replicated message store in BigTable
- Weak consistency model for some operations (e.g., msg read)
- Stronger consistency for others (e.g., send msg)
Platform as a Service (PaaS)
Platform as a Service (PaaS)

- Cloud provides middleware/infrastructure

Cloud Provider (i.e., PaaS Provider)

- Middleware
- Hardware
Platform as a Service (PaaS)

Cloud Provider (i.e., PaaS Provider)

- Cloud provides middleware/infrastructure
  - For example, Microsoft Common Language Runtime (CLR)
Platform as a Service (PaaS)

- Cloud provides middleware/infrastructure
  - For example, Microsoft Common Language Runtime (CLR)
  - App provider pays the cloud for the platform
Platform as a Service (PaaS)

- Cloud provides middleware/infrastructure
  - For example, Microsoft Common Language Runtime (CLR)
  - App provider pays the cloud for the platform
Platform as a Service (PaaS)

- Cloud provides middleware/infrastructure
  - For example, Microsoft Common Language Runtime (CLR)
  - App provider pays the cloud for the platform
  - Customer pays app provider for the service
Platform as a Service (PaaS)

- Cloud provides middleware/infrastructure
  - For example, Microsoft Common Language Runtime (CLR)
  - App provider pays the cloud for the platform
  - Customer pays app provider for the service
Platform as a Service (PaaS)

- Cloud provides middleware/infrastructure
  - For example, Microsoft Common Language Runtime (CLR)
  - App provider pays the cloud for the platform
  - Customer pays app provider for the service
  - Example: Windows Azure, Google App Engine, etc.
A Typical PaaS: Facebook
A Typical PaaS: Facebook

Facebook Provider

Middleware

Hardware
A Typical PaaS: Facebook

• Facebook offers PaaS capabilities to App provider
A Typical PaaS: Facebook

• Facebook offers PaaS capabilities to App provider
A Typical PaaS: Facebook

- Facebook offers PaaS capabilities to App provider
  - Facebook APIs allow access to social network properties
A Typical PaaS: Facebook

- Facebook offers PaaS capabilities to App provider
  - Facebook APIs allow access to social network properties
  - App providers adopt their services (e.g., game) onto Facebook
Facebook offers PaaS capabilities to App provider
- Facebook APIs allow access to social network properties
- App providers adopt their services (e.g., game) onto Facebook
A Typical PaaS: Facebook

- Facebook offers PaaS capabilities to App provider
  - Facebook APIs allow access to social network properties
  - App providers adopt their services (e.g., game) onto Facebook
A Typical PaaS: Facebook

- Facebook offers PaaS capabilities to App provider
  - Facebook APIs allow access to social network properties
  - App providers adopt their services (e.g., game) onto Facebook
A Typical PaaS: Facebook

- Facebook offers PaaS capabilities to App provider
  - Facebook APIs allow access to social network properties
  - App providers adopt their services (e.g., game) onto Facebook
  - Facebook itself also uses PaaS provided by its company, e.g., log analysis for recommendations
Infrastructure as a Service (IaaS)
Infrastructure as a Service (IaaS)

Cloud Provider (i.e., IaaS Provider)

- Cloud provides raw computing resources
Infrastructure as a Service (IaaS)

- Cloud provides raw computing resources
  - Virtual machines, blade servers, hard disk, etc.
Infrastructure as a Service (IaaS)

- Cloud provides raw computing resources
  - Virtual machines, blade servers, hard disk, etc.
  - App provider pays the cloud for the resources
Infrastructure as a Service (IaaS)

- Cloud provides raw computing resources
  - Virtual machines, blade servers, hard disk, etc.
  - App provider pays the cloud for the resources
Infrastructure as a Service (IaaS)

- Cloud provides raw computing resources
  - Virtual machines, blade servers, hard disk, etc.
  - App provider pays the cloud for the resources
  - Customer pays App provider for the service
Infrastructure as a Service (IaaS)

- Cloud provides raw computing resources
  - Virtual machines, blade servers, hard disk, etc.
  - App provider pays the cloud for the resources
  - Customer pays App provider for the service
Infrastructure as a Service (IaaS)

- Cloud provides raw computing resources
  - Virtual machines, blade servers, hard disk, etc.
  - App provider pays the cloud for the resources
  - Customer pays App provider for the service
  - Example: Amazon Web Services, Rackspace Cloud, etc.
Typical IaaS: EC2 and S3

Amazon

Hardware
Typical IaaS: EC2 and S3
• Netflix (app) heavily depends on Amazon AWS:
Typical IaaS: EC2 and S3

- Netflix (app) heavily depends on Amazon AWS:
  - Media files are stored in S3
  - Transcoding to target devices (e.g., iPad)
  - Analysis of streaming sessions based on Elastic MapReduce
Typical IaaS: EC2 and S3

- Netflix (app) heavily depends on Amazon AWS:
• Netflix (app) heavily depends on Amazon AWS:
  - Media files are stored in S3
  - Transcoding to target devices (e.g., iPad) using EC2
Typical IaaS: EC2 and S3

- Netflix (app) heavily depends on Amazon AWS:
  - Media files are stored in S3
  - Transcoding to target devices (e.g., iPad) using EC2
• Netflix (app) heavily depends on Amazon AWS:
  - Media files are stored in S3
  - Transcoding to target devices (e.g., iPad) using EC2
Three types of services:

- **Software as a Service (SaaS)**
  - Analogy: Restaurant. Prepares & serves entire meal, does the dishes, etc.

- **Platform as a Service (PaaS)**
  - Analogy: Take-out food. Prepares meal but does not serve it.

- **Infrastructure as a Service (IaaS)**
  - Analogy: Grocery store. Provides raw ingredients.
The Major Cloud Providers
The Major Cloud Providers

- **Amazon** is the big player:
  - Infrastructure as a service (e.g., EC2)
  - Storage as a service (e.g., S3)

- Microsoft Azure: It has similar services to Amazon, with an emphasis on .Net programming model
- Google App Engine: It offers programming interface, Hadoop, also software as a service, e.g., Gmail and Google Docs
- IBM, HP, Yahoo!: They seem to focus on enterprise scale cloud apps
The Major Cloud Providers

- **Amazon** is the big player:
  - Infrastructure as a service (e.g., EC2)
  - Storage as a service (e.g., S3)

- **But there are many others:**
  - **Microsoft Azure**: It has similar services to Amazon, with an emphasis on .Net programming model
  - **Google App Engine**: It offers programming interface, Hadoop, also software as a service, e.g., Gmail and Google Docs
  - **IBM, HP, Yahoo!**: They seem to focus on enterprise scale cloud apps
Lecture Outline

- Cloud Computing
- Challenges in the Clouds
- A Concrete Cloud Reliability Case
Lecture Outline

• Cloud Computing

• Challenges in the Clouds

• A Concrete Cloud Reliability Case
What Kinds of Challenges?
What Kinds of Challenges?

• Scalability
• Availability and reliability
• Security and privacy
What Kinds of Challenges?

- Scalability
- Availability and reliability
- Security and privacy
Scalability
Scalability

- What if one computer is not enough?
Scalability

• What if one computer is not enough?
  - Buy a bigger (server-class) computer
Scalability

- What if one computer is not enough?
  - Buy a bigger (server-class) computer

- What if the biggest computer is not enough?
Scalability

- What if one computer is not enough?
  - Buy a bigger (server-class) computer

- What if the biggest computer is not enough?
  - Buy many computers
Scalability
Scalability
Scalability

Rack

Network switches (connects nodes with each other and with other racks)
Scalability

Network switches (connects nodes with each other and with other racks)

Many nodes/blades (often identical)
Scalability

- Rack
- Network switches (connects nodes with each other and with other racks)
- Many nodes/blades (often identical)
- Storage device(s)
Scalability

• What if cluster is too big to fit into machine room?
Scalability

- What if cluster is too big to fit into machine room?
  - Build a separate building for the cluster
  - Building can have lots of cooling and power
Scalability

• What if cluster is too big to fit into machine room?
  - Build a separate building for the cluster
  - Building can have lots of cooling and power
  - Result: Data center
Google Data Center in Oregon
Google Data Center in Oregon

Data centers (size of a football field)
Google Data Center in Oregon

A warehouse-sized computer
- A single data center can easily contain 10,000 racks with 100 cores in each rack (1,000,000 cores total)

Data centers (size of a football field)
Google Data Center Locations
Google Data Centers in the USA
Google Data Centers in Europe
Google Data Centers World Wide
Open Challenges
Open Challenges

- Can you manage thousands of racks effectively?

- Cloud monitor systems (e.g., PlanetSeer [OSDI'04])

- Can you design more scalable data center network?
Open Challenges

• Can you manage thousands of racks effectively?
  - Cloud monitor systems (e.g., PlanetSeer [OSDI’04])
Open Challenges

- Can you make data center more scalable?
  - Scalable data center architecture (e.g., VL2 [SIGCOMM'09])
- Can you design more scalable data center network?
- Can you manage thousands of racks effectively?
  - Cloud monitor systems (e.g., PlanetSeer [OSDI’04])
- Can you make data center more scalable?
Open Challenges

• Can you make data center more scalable?
  - Scalable data center architecture (e.g., VL2 [SIGCOMM’09])

• Can you manage thousands of racks effectively?
  - Cloud monitor systems (e.g., PlanetSeer [OSDI’04])
What Kinds of Challenges?

- Scalability
- Availability and reliability
- Security and privacy
What Kinds of Challenges?

- Scalability
- Availability and reliability
- Security and privacy
Availability & Reliability

• Is the cloud always there when you need it?
  - Service outages
  - Connectivity outages
Recent Cloud Disasters

Data Center Outages Generate Big Losses

Downtime in a data center can cost an average of $505,500 per incident, according to a Ponemon Institute study.

By Chandler Harris InformationWeek
May 12, 2011 01:22 PM

Sure data center failures are costly, but how costly? Try an average of $5,600 per minute, according to a study of outages at U.S.-based data centers by the Ponemon Institute.

"Calculating the Cost of Data Center Outages," by the Ponemon Institute, analyzed costs associated with downtime at 41 data centers across varying industry segments with a minimum size of 2,500 square feet. The study was sponsored by Emerson Network Power, a provider of storage and energy products and services, among other things.

Analytics Slideshow: 2010 Data Center
Recent Cloud Disasters

Amazon: Networking Error Caused Cloud Outage

By: Rich Miller
April 29th, 2011

Last week’s lengthy outage for the Amazon Web Services cloud computing platform was caused by a network configuration error as Amazon was attempting to upgrade capacity on its network. That error triggered a sequence of events that culminated in a “re-mirroring storm” in which automated replication of storage volumes maxed out the capacity of Amazon’s servers in a portion of their platform.
Recent Cloud Disasters

The same lightning storms that knocked out power to residents of Northern Virginia on June 30 also took down some of the world's best-known websites.

Photo by Win McNamee/Getty Images

Thanks to the cloud, websites and apps around the world can tap into vast, remote stores of data and computing power.

And thanks to the cloud, one good blow to one of those vast, remote storage centers can take down websites and apps around the world.

That's what happened this past weekend. A ferocious lightning storm in Northern Virginia took down Netflix, Instagram, Pinterest, Heroku, and more—not because any of those companies are based in Northern Virginia, but because they all apparently rely heavily on Amazon’s Elastic Compute Cloud facility there. Amazon said the storm, for reasons not immediately explained, took out both its main power supply and its backup generator.

The outage brought to mind a similar incident a year ago, in which an outage at the same Amazon facility felled Reddit, Quora, and several other sites.
Top10 Cloud Service Outages

WORST CLOUD OUTAGES OF 2013 (SO FAR)
Open Challenges

• Can you build a system to find out the root-cause when a service becomes unavailable?
Open Challenges

• Can you build a system to find out the root-cause when a service becomes unavailable?
  - Diagnosis systems (e.g., Sherlock [SIGCOMM’07])
Open Challenges

• Can you build a system to find out the root-cause when a service becomes unavailable?
  - Diagnosis systems (e.g., Sherlock [SIGCOMM’07])
  - Accountable cloud (e.g., AVM [OSDI’10])
Open Challenges

• Can you build a system to find out the root-cause when a service becomes unavailable?
  - Diagnosis systems (e.g., Sherlock [SIGCOMM’07])
  - Accountable cloud (e.g., AVM [OSDI’10])

• Can you propose an approach to make the clouds more robust?
Open Challenges

• Can you build a system to find out the root-cause when a service becomes unavailable?
  - Diagnosis systems (e.g., Sherlock [SIGCOMM’07])
  - Accountable cloud (e.g., AVM [OSDI’10])

• Can you propose an approach to make the clouds more robust?
  - Fault tolerate systems (e.g., F10 [NSDI’13])
What Kinds of Challenges?

- Scalability
- Availability and reliability
- Security and privacy
What Kinds of Challenges?

- Scalability
- Availability and reliability
- Security and privacy
Security & Privacy

- Compromised your cloud accounts
  - Hacker does not need to break into your home to steal all your private data, if he can break or guess your cloud password
  - Even worse, if hack who cracks your Facebook account get into your accounts everywhere online
Security & Privacy

• Compromised your cloud accounts
  - Hacker does not need to break into your home to steal all your private data, if he can break or guess your cloud password
  - Even worse, if hack who cracks your Facebook account get into your accounts everywhere online

• You do not know if the cloud providers read your private data
Open Challenges

• Can you build a system to preserve the privacy of your data on the clouds?

- MAC for MapReduce (e.g., Airavat [NSDI'10])
- Trusted storage (e.g., Depot [OSDI'10])
Open Challenges

• Can you build a system to preserve the privacy of your data on the clouds?
  - MAC for MapReduce (e.g., Airavat [NSDI’10])
Open Challenges

• Can you build a system to preserve the privacy of your data on the clouds?
  - MAC for MapReduce (e.g., Airavat [NSDI’10])
  - Trusted storage (e.g., Depot [OSDI’10])
Open Challenges

• Can you build a system to preserve the privacy of your data on the clouds?
  - MAC for MapReduce (e.g., Airavat [NSDI’10])
  - Trusted storage (e.g., Depot [OSDI’10])

• Can you propose an approach to verify if the cloud provider modifies your data?
Open Challenges

• Can you build a system to preserve the privacy of your data on the clouds?
  - MAC for MapReduce (e.g., Airavat [NSDI’10])
  - Trusted storage (e.g., Depot [OSDI’10])

• Can you propose an approach to verify if the cloud provider modifies your data?
  - Trusted cloud computing (e.g., Excalibur [USENIX Sec’12])
More Risks?
More Risks?

- **EverClouds** is a project of DeDiS group (Bryan is PI):
  - aims to solve tricky cloud security problems (e.g., timing channels)
  - tries to make the clouds more reliable (e.g., failure detection)
More Risks?

• **EverClouds** is a project of DeDiS group (Bryan is PI):
  - aims to solve tricky cloud security problems (e.g., timing channels)
  - tries to make the clouds more reliable (e.g., failure detection)

• We already have some efforts:
More Risks?

• **EverClouds** is a project of DeDiS group (Bryan is PI):
  - aims to solve tricky cloud security problems (e.g., timing channels)
  - tries to make the clouds more reliable (e.g., failure detection)

• We already have some efforts:
  - **SRA**: A cloud structural reliability auditing system (submitted)
More Risks?

- **EverClouds** is a project of DeDiS group (Bryan is PI):
  - aims to solve tricky cloud security problems (e.g., timing channels)
  - tries to make the clouds more reliable (e.g., failure detection)

- We already have some efforts:
  - **SRA**: A cloud structural reliability auditing system (submitted)
  - **iRec**: A cloud independence recommender system (HotDep’13)
More Risks?

• **EverClouds** is a project of DeDiS group (Bryan is PI):
  - aims to solve tricky cloud security problems (e.g., timing channels)
  - tries to make the clouds more reliable (e.g., failure detection)

• We already have some efforts:
  - **SRA**: A cloud structural reliability auditing system (submitted)
  - **iRec**: A cloud independence recommender system (HotDep’13)
  - **P-SRA**: A privacy-preserving structural-reliability auditor (CCSW’13)
  - Timing channel control with provider-enforced deterministic execution (CCSW’10)
More Risks?

- EverClouds is a project of DeDiS group (Bryan is PI):
  - aims to solve tricky cloud security problems (e.g., timing channels)
  - tries to make the clouds more reliable (e.g., failure detection)

- We already have some efforts:
  - SRA: A cloud structural reliability auditing system (submitted)
  - iRec: A cloud independence recommender system (HotDep’13)
  - P-SRA: A privacy-preserving structural-reliability auditor (CCSW’13)
  - Timing channel control with provider-enforced deterministic execution (CCSW’10)
Lecture Outline

• Cloud Computing

• Challenges in the Clouds

• A Concrete Cloud Reliability Case
Lecture Outline

• Cloud Computing

• Challenges in the Clouds

• A Concrete Cloud Reliability Case
An Untold Story of Redundant Clouds: Making Your Service Deployment Truly Reliable

Ennan Zhai\textsuperscript{1}, Ruichuan Chen\textsuperscript{2}, David Isaac Wolinsky\textsuperscript{1}, Bryan Ford\textsuperscript{1}

\textsuperscript{1}Yale University & \textsuperscript{2}Bell Labs
Road-Map

- Motivations
- Goal & Insight
- iRec System
- Next Steps
Background

- Application providers:
  - enjoy the simplicity of using the clouds
Background

• Application providers:
  - enjoy the simplicity of using the clouds
  - have no idea about what happen in the clouds
Background

- Application providers:
  - enjoy the simplicity of using the clouds
  - have no idea about what happen in the clouds
  - rent multiple clouds for redundancy
Example 1: Netflix Application Service

Netflix Application Service

EC2 availability zone

EC2 availability zone

EC2 availability zone

App

IaaS
Example 2: iCloud

iCloud Application Service

Amazon EC2 Service

Microsoft Azure Service

IaaS

App
Problem

Email App

Cloud Provider A

Cloud Provider B
Problem

Cloud Provider A  Email App  Cloud Provider B

Third-party infrastructure components
Problem

Cloud Provider A

Cloud Provider B

Third-party infrastructure components

ISP A

ISP B

ISP C
Problem

Cloud Provider A  Email App  Cloud Provider B

Third-party infrastructure components

ISP A  ISP B  ISP C

Power Source
Problem

Cloud Provider A

Cloud Provider B

Email App

ISP A

ISP B

ISP C

Third-party infrastructure components

Power Source
Problem

Become unavailable!

Cloud Provider A

Cloud Provider B

Third-party infrastructure components

ISP A

ISP B

ISP C

Power Source
Problem

Lightning strikes Amazon's European cloud

Summary: The lightning strike damaged a power company's transformer, causing disruption to Amazon Web Services's European cloud, and may have affected Microsoft's BPOS as well.

The outage, which Amazon Web Services (AWS) acknowledged on Sunday evening, affected its Dublin-based Elastic Compute Cloud (EC2) and Relational Database Service (RDS) cloud services, among others. The damage to the electricity infrastructure may have affected Microsoft's Business Productivity Online Services (BPOS) cloud as well, Microsoft said in a separate statement.
Existing Efforts

• Cloud providers allocate or tolerate failures via:
  - diagnosis systems, e.g., Sherlock.
  - fault-tolerant systems, e.g., F10, Skute.
Existing Efforts

- Cloud providers allocate or tolerate failures via:
  - diagnosis systems, e.g., Sherlock.
  - fault-tolerant systems, e.g., F10, Skute.

- Solving the problem after the outage occurs
Existing Efforts

- Cloud providers allocate or tolerate failures via:
  - diagnosis systems, e.g., Sherlock.
  - fault-tolerant systems, e.g., F10, Skute.

- Solving the problem after the outage occurs

- We want to prevent the problem before the outage occurs
Existing Efforts

- Cloud providers allocate or tolerate failures via:
  - diagnosis systems, e.g., Sherlock.
  - fault-tolerant systems, e.g., F10, Skute.
- Solving the problem after the outage occurs
- We want to prevent the problem before the outage occurs
- Recommending truly independent redundancy services when deploying applications
Road-Map

- Motivations
- Goal & Insight
- iRec System
- Next Steps
Road-Map

• Motivations

• Goal & Insight

• iRec System

• Next Steps
Goal & Insight

App Provider

Cloud A

Cloud B

Cloud C
Goal & Insight

App Provider

Select two clouds for redundancy

Cloud A

Cloud B

Cloud C
Goal & Insight

App Provider

A and B?

Cloud A  Cloud B  Cloud C
Goal & Insight

App Provider: B and C?
App Provider

A and C?

Cloud A

Cloud B

Cloud C
Goal & Insight

Select two clouds for redundancy: A&B? B&C? or A&C?
Goal & Insight

App Provider → Recommender

Cloud A → Cloud B → Cloud C
Assessing independence by the # of overlapping components between clouds
Goal & Insight

App Provider

ISP A  Power A

Cloud A

Power B

Cloud B

Recommender

Cloud C
Goal & Insight

App Provider

ISP A
Power A
Power B

Cloud A

Cloud B

Cloud C

Recommender
Goal & Insight

App Provider

ISP A
Power A
Power B

Cloud A

Cloud B

Cloud C

Recommender

ISP A
Power A
Power B
Goal & Insight

App Provider

ISP A
Power A
Power B

Cloud A

ISP B
Power A
Power B

Cloud B

ISP B

Power A
Power B

Cloud C

ISP A
Power A

Power B

ISP B
App Provider

ISP A
Power A
Power B

Cloud A

ISP B
Power A
Power B

Cloud B

ISP B
Power A
Power B

Cloud C

Recommender
Goal & Insight

App Provider

Cloud A
ISP A
Power A
Power B

Cloud B
ISP B
Power A
Power B

Cloud C
ISP B
Power C

Recommender
Goal & Insight

App Provider

ISP A
Power A
Power B

Cloud A

\[ \cap = 2 \]

ISP B
Power A
Power B

ISP B
Power C

Cloud B

Cloud C

ISP A
Power A
Power B

ISP B
Power B

ISP B

Recommended

Deployment

<table>
<thead>
<tr>
<th>n</th>
</tr>
</thead>
</table>

ISP A
Power A

ISP B
Power B

ISP B

Power C
Goal & Insight

<table>
<thead>
<tr>
<th>Deployment</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud A, B</td>
<td>2</td>
</tr>
</tbody>
</table>

App Provider

ISP A
Power A
Power B

Cloud A

ISP B
Power A
Power B

Cloud B

ISP B
Power C

Cloud C

ISP A
Power A

ISP B
Power B

ISP B
Power C
Goal & Insight

App Provider

ISP A
Power A
Power B

Cloud A

ISP B
Power A
Power B

Cloud B

ISP B
Power C

Cloud C

ISP A
Power A

ISP B
Power B

ISP B
Power C

ISP A
Power A

ISP B
Power B

ISP B
Power C

= |n|
Goal & Insight

| Deployment          | |n| |
|---------------------|--|--|
| Cloud B, C          | 1 |
| Cloud A, B          | 2 |

ISP A
Power A
Power B

Cloud A

ISP B
Power A
Power B

Cloud B

ISP B
Power C

Cloud C
Goal & Insight

| Deployment       | $|n|$ |
|------------------|-----|
| Cloud B, C       | 1   |
| Cloud A, B       | 2   |

Cloud A, C
Cloud B, C
Cloud A, B

ISP A
Power A
ISP B
Power C
ISP A
Power A
ISP B
Power C
ISP B
Power C
Goal & Insight

| Deployment | |n|
|------------|--|
| Cloud B, C | 1 |
| Cloud A, B | 2 |

Cloud A, C
Cloud B, C
Cloud A, B

ISP A
Power A
ISP B
Power B
ISP C
Power C

ISP A
Power A
ISP B
Power B
ISP C
Power C
Goal & Insight

<table>
<thead>
<tr>
<th>Deployment</th>
<th></th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud A, C</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Cloud B, C</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Cloud A, B</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

App Provider

ISP A
Power A

Cloud A

ISP B
Power C

Cloud B

ISP B
Power C

Cloud C

ISP A
Power A

ISP B
Power B

ISP A
Power A

ISP B
Power B

ISP B
Power C

ISP A
Power A

ISP B
Power B

ISP B
Power C

ISP A
Power A

ISP B
Power B

ISP B
Power C
Goal & Insight

| Deployment  | |n| |
|-------------|-----|
| Cloud A, C  | 0   |
| Cloud B, C  | 1   |
| Cloud A, B  | 2   |

App Provider  \rightarrow Recommender

Cloud A

Cloud B

Cloud C
## Goal & Insight

| Deployment     | \(|n|\) |
|----------------|--------|
| Cloud A, C     | 0      |
| Cloud B, C     | 1      |
| Cloud A, B     | 2      |

### Diagram: Ranking List

- **App Provider**
- **Recommendation**

---

**Cloud A**

**Cloud B**

**Cloud C**
Road-Map

- Motivations
- Goal & Insight
- iRec System
- Next Steps
Road-Map

- Motivations
- Goal & Insight
- iRec System
- Next Steps
Strawman Solution 1

App Provider

Recommend

Cloud Provider1

Cloud Provider2

Cloud Provider3
Strawman Solution 1

Privacy Concern!

Cloud Provider1

Cloud Provider2

Cloud Provider3
Strawman Solution 2

App Provider

Cloud Provider1

Trusted Third Party

Cloud Provider2

Cloud Provider3
Strawman Solution 2

It is hard to find!

Cloud Provider1

Cloud Provider2

Cloud Provider3
Strawman Solution 3

[Xiao et al, CCSW’13]
Strawman Solution 3

[S Xiao et al, CCSW’13]

SMPC is difficult to scale!
Our Approach - iRec
Our Approach - iRec

- The first cloud independence recommender sys:
  - achieving our goal
  - preserving privacy of each cloud provider
  - practical
Our Approach - iRec

• The first cloud independence recommender sys:
  - achieving our goal
  - preserving privacy of each cloud provider
  - practical

Preliminary background: PSI-CA
Preliminary: PSI-CA

- Private set-intersection cardinality proposed by [Freedman et al, EuroCrypt’04].
- Allows $k$ parties to compute the # of overlapping elements without learning other information.
Preliminary: PSI-CA

- Private set-intersection cardinality proposed by [Freedman et al, EuroCrypt’04].
- Allows $k$ parties to compute the # of overlapping elements without learning other information.
Preliminary: PSI-CA

- Private set-intersection cardinality proposed by [Freedman et al, EuroCrypt’04].
- Allows $k$ parties to compute the # of overlapping elements without learning other information.
Preliminary: PSI-CA

- Private set-intersection cardinality proposed by [Freedman et al, EuroCrypt’04].
- Allows $k$ parties to compute the # of overlapping elements without learning other information.
Preliminary: PSI-CA

- Private set-intersection cardinality proposed by [Freedman et al, EuroCrypt’04].
- Allows $k$ parties to compute the # of overlapping elements without learning other information.
Alice and Bob have set A and B respectively, and Alice wants to jointly compute $|A \cap B|$.

- Alice makes a polynomial $P$ whose roots are the elements of data set A.
- Alice encrypts the coefficients of $P$ and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.
- Bob evaluates $P(B_i)$ for each element in data set B.
- Bob returns the encrypted evaluations to Alice.
- Alice decrypts it and counts the number of zeroes.

Data Set A

- 12
- 5
- 4

Data Set B

- 1
- 4
- 6
- 2

Preliminary: PSI-CA
Alice and Bob has set A and B respectively and Alice wants to jointly compute $|A \cap B|$.  

Alice makes a polynomial $P$ whose roots are the elements of data set A. 

Bob evaluates $P(B_i)$ for each element in data set B. 

Bob returns the encrypted evaluations to Alice. 

Alice decrypts it and counts the number of zeroes.

Preliminary: PSI-CA

$P = (X-12)(X-5)(X-4)$  
$= x^3 - 21x^2 + 128x - 240$  

- Alice and Bob has set A and B respectively and Alice wants to jointly compute $|A \cap B|$.  
- Alice makes a polynomial $P$ whose roots are the elements of data set A.
• Alice and Bob has set A and B respectively and Alice wants to jointly compute $|A \cap B|$.
• Alice makes a polynomial $P$ whose roots are the elements of data set A.
• Alice encrypts the coefficients of $P$ and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.

Preliminary: PSI-CA

\[ P = (X-12)(X-5)(X-4) \]
\[ = x^3 - 21x^2 + 128x - 240 \]
\[ \{ E(1), E(-21), E(128), E(-240) \} \]
Alice and Bob has set A and B respectively and Alice wants to jointly compute \(|A \cap B|\).

Alice makes a polynomial \(P\) whose roots are the elements of data set A.

Alice encrypts the coefficients of \(P\) and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.

Bob evaluates \(P(B_i)\) for each element in data set B.
Alice and Bob have set A and B respectively and Alice wants to jointly compute $|A \cap B|$. 

- Alice makes a polynomial $P$ whose roots are the elements of data set A.
- Alice encrypts the coefficients of $P$ and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.
- Bob evaluates $P(B_i)$ for each element in data set B.
Alice and Bob have set A and B respectively and Alice wants to jointly compute |A ∩ B|.

- Alice makes a polynomial $P$ whose roots are the elements of data set A.
- Alice encrypts the coefficients of $P$ and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.
- Bob evaluates $P(B_i)$ for each element in data set B.

\[
P = (X-12)(X-5)(X-4) = x^3-21x^2+128x-240
\]
• Alice and Bob have set A and B respectively, and Alice wants to jointly compute |A ∩ B|.
• Alice makes a polynomial P whose roots are the elements of data set A.
• Alice encrypts the coefficients of P and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.
• Bob evaluates P(B_i) for each element in data set B.
• Alice and Bob has set A and B respectively and Alice wants to jointly compute |A ∩ B|.
• Alice makes a polynomial P whose roots are the elements of data set A.
• Alice encrypts the coefficients of P and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.
• Bob evaluates P(B_i) for each element in data set B.
• Bob returns the encrypted evaluations to Alice.
Alice and Bob has set A and B respectively and Alice wants to jointly compute |A ∩ B|.
Alice makes a polynomial P whose roots are the elements of data set A.
Alice encrypts the coefficients of P and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.
Bob evaluates P(B_i) for each element in data set B.
Bob returns the encrypted evaluations to Alice.
Alice decrypts it and counts the number of zeroes.
Alice and Bob has set A and B respectively and Alice wants to jointly compute $|A \cap B|$.

- Alice makes a polynomial $P$ whose roots are the elements of data set A.
- Alice encrypts the coefficients of $P$ and sends them to Bob. Note that Alice sends homomorphic encryptions of the coefficients to Bob.
- Bob evaluates $P(B_i)$ for each element in data set B.
- Bob returns the encrypted evaluations to Alice.
- Alice decrypts it and counts the number of zeroes.

\[
P = (X-12)(X-5)(X-4) = x^3-21x^2+128x-240
\]

\[
\{-132, 0, -12, -60\}
\]
Preliminary: PSI-CA
Our Approach - iRec
Our Approach - iRec

Select two clouds for redundancy: A&B? B&C? or A&C?
Step 1

App Provider -> iRec

Cloud A
ISP A  Power A

Cloud B  Power B

Cloud C
ISP B  Power C
Step 2

App Provider

Cloud A

iRec

Cloud B

Cloud C

ISP A

Power A

Power B

ISP B

Power C
Step 3

App Provider

Cloud A

ISP A
- Power A
- Power B

Power A

Cloud B

ISP B
- Power A
- Power B

Power B

Cloud C

ISP B
- Power C

Power C

iRec
Step 3

App Provider

Cloud A

PSI-CA

ISP A
Power A
Power B

Cloud B

PSI-CA

ISP B
Power A
Power B

Cloud C

ISP B
Power C

iRec

ISP A
Power A

Power B

ISP B

Power C
Step 5

| Deployment          | \(|n|\) |
|---------------------|-------|
| Cloud A, C          | 0     |
| Cloud B, C          | 1     |
| Cloud A, B          | 2     |

App Provider

iRec

Cloud A

Cloud B

Cloud C

ISP A

Power A

Power B

ISP B

Power C
Step 5

Ranking List

| Deployment   | |n| |
|--------------|---|---|
| 1. Cloud A, C | 0 |
| 2. Cloud B, C | 1 |
| 3. Cloud A, B | 2 |

Deployment

- Cloud A, C: 0
- Cloud B, C: 1
- Cloud A, B: 2

Network Diagram:

- Cloud A
- Cloud B
- Cloud C
- ISP A
- Power A
- Power B
- ISP B
- Power C
An Improvement Version

- Different infrastructure components play different roles in the clouds
An Improvement Version

- Different infrastructure components play different roles in the clouds
- Power source might be much more likely to fail than ISPs
An Improvement Version

• Different infrastructure components play different roles in the clouds

• Power source might be much more likely to fail than ISPs

• We propose an improvement version
An Improvement Version

• Different infrastructure components play different roles in the clouds

• Power source might be much more likely to fail than ISPs

• We propose an improvement version
  - Using Weighted PSI-CA (W-PSI-CA) to instead of PSI-CA in Step3
Recall: Step 3

App Provider

Cloud A

ISP A
Power A
Power B

PSI-CA

Cloud B

ISP B
Power A
Power B

PSI-CA

Cloud C

ISP B
Power C

ISP A
Power A

Power B

ISP B

Power C

iRec
Recall: Step 3

Result is 2
Using W-PSI-CA
Using W-PSI-CA
Using W-PSI-CA

<table>
<thead>
<tr>
<th>ISP</th>
<th>Power A</th>
<th>Power B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

- Cloud A
- Cloud B
- ISP A
- ISP B
- Power A
- Power B
Using W-PSI-CA

Weights

<table>
<thead>
<tr>
<th>ISP A</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power A</td>
<td>2</td>
</tr>
<tr>
<td>Power B</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISP B</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power A</td>
<td>2</td>
</tr>
<tr>
<td>Power B</td>
<td>2</td>
</tr>
</tbody>
</table>
Using W-PSI-CA

Cloud A

ISP A
Power A
Power B

Power A
2
Power B
2

Cloud B

ISP B
Power A
Power B

Power A
2
Power B
2

DSI
ISP A
Power A
Power B
Power B

ISP B
Power A
Power B

Power A
2
Power B
2
Using W-PSI-CA

Cloud A

<table>
<thead>
<tr>
<th>ISP A</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power A</td>
<td>2</td>
</tr>
<tr>
<td>Power B</td>
<td>2</td>
</tr>
</tbody>
</table>

Power A

Power B

Cloud B

<table>
<thead>
<tr>
<th>ISP B</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power A</td>
<td>2</td>
</tr>
<tr>
<td>Power B</td>
<td>2</td>
</tr>
</tbody>
</table>

Power A

Power B

ISP A

ISP B
Using W-PSI-CA

Cloud A

<table>
<thead>
<tr>
<th>DSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISP A</td>
</tr>
<tr>
<td>Power A</td>
</tr>
<tr>
<td>Power A</td>
</tr>
<tr>
<td>Power B</td>
</tr>
<tr>
<td>Power B</td>
</tr>
</tbody>
</table>

Cloud B

<table>
<thead>
<tr>
<th>DSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISP B</td>
</tr>
<tr>
<td>Power A</td>
</tr>
<tr>
<td>Power A</td>
</tr>
<tr>
<td>Power B</td>
</tr>
<tr>
<td>Power B</td>
</tr>
</tbody>
</table>

ISP A

Power A

Power B

ISP B
Using W-PSI-CA

Result is 4

Cloud A

<table>
<thead>
<tr>
<th>DSI</th>
<th>ISP A</th>
<th>Power A</th>
<th>Power A</th>
<th>Power B</th>
<th>Power B</th>
</tr>
</thead>
</table>

Cloud B

| DSI | ISP B | Power A | Power A | Power B | Power B |

PSI-CA

ISP A

Power A

ISP B

Power B
Case Study
Step 1

Select two clouds for redundancy: A&B? B&C? or A&C?
Step 1
Step 2

App Provider

Cloud A

iRec

Cloud B

Cloud C

Weights

ISP A

Power A

Power B

ISP B

Power C
Step 3 & 4 with W-PSI-CA

| Deployment      | \(|n|\) |
|-----------------|-------|
| Cloud A, C      | 0     |
| Cloud B, C      | 1     |
| Cloud A, B      | 2     |

- App Provider
- iRec

- ISP A
  - Power A: 1
  - Power B: 1
- ISP B
  - Power A: 1
  - Power B: 1
- ISP B
  - Power C: 1

Cloud A, Cloud B, Cloud C
Step 3 & 4 with W-PSI-CA

| Deployment       | |n| |
|------------------|---|---|
| Cloud A, C       | 0 |
| Cloud B, C       | 1 |
| Cloud A, B       | 2 |

**App Provider**

<table>
<thead>
<tr>
<th>ISP A</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power A</td>
<td>1</td>
</tr>
<tr>
<td>Power B</td>
<td>1</td>
</tr>
</tbody>
</table>

**Cloud A**

- ISP A
- ISP A
- ISP A
- Power A
- Power A
- Power A

**Cloud B**

- ISP B
- ISP B
- ISP B
- Power A
- Power A
- Power A

**Cloud C**

- ISP B
- ISP B
- ISP B
- Power C
- Power C
- Power C

**iRec**

- ISP B
- ISP B
- ISP B
- Power C
- Power C
- Power C
Step 3 & 4 with W-PSI-CA
Step 3 & 4 with W-PSI-CA

| Deployment       | |n|  |
|------------------|---|---|
| Cloud A, C       | 0 |
| Cloud B, C       | 1 |
| Cloud A, B       | 2 |

App Provider

Cloud A

ISP A
ISP A
ISP A
Power A
Power B

Cloud B

Power B

Cloud C

ISP B
ISP B
ISP B
Power C

iRec

PSI-CA

ISP A
ISP A
ISP A
Power A
Power B

ISP B
ISP B
ISP B
Power C

Power A

Power B

Power C
Step 3 & 4 with W-PSI-CA

| Deployment   | $|n|$ |
|--------------|-----|
| Cloud A, C   | 0   |
| Cloud B, C   | 1   |
| Cloud A, B   | 2   |

App Provider

Cloud A
- ISP A
- ISP A
- ISP A
- Power A
- Power B

Cloud B

Cloud C
- ISP B
- ISP B
- ISP B
- Power C

iRec

ISP A
- Power A

Power B

ISP B

Power C

PSI-CA
Step 3 & 4 with W-PSI-CA

| Deployment   | |n|
|--------------|---|
| Cloud A, C   | 0 |
| Cloud B, C   | 1 |
| Cloud A, B   | 2 |

- **App Provider**
  - ISP A
  - ISP A
  - ISP A
  - Power A
  - Power B

- **Cloud A**

- **Cloud B**
  - ISP B
  - ISP B
  - ISP B
  - Power A
  - Power B

- **Cloud C**
  - ISP B
  - ISP B
  - ISP B
  - Power C

- **iRec**
  - ISP A
  - ISP A
  - ISP A
  - Power A
  - Power B

- **PSI-CA**

- **ISP A**
- **Power A**
- **Power B**
- **ISP B**
- **Power C**
Step 3 & 4 with W-PSI-CA

| Deployment | |n| |
|------------|---|---|
| Cloud A, C | 0 |
| Cloud B, C | 3 |
| Cloud A, B | 2 |

App Provider

Cloud A
- ISP A
- ISP A
- ISP A
- Power A
- Power B

Cloud B
- ISP B
- ISP B
- ISP B
- Power A
- Power B

Cloud C
- ISP B
- ISP B
- ISP B
- Power C

iRec

PSI-CA

ISP A
- Power A

Power B

ISP B
- Power A

ISP C
- Power C
Step 3 & 4 with W-PSI-CA

<table>
<thead>
<tr>
<th>Deployment</th>
<th></th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud A, C</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Cloud B, C</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Cloud A, B</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

App Provider

Cloud A
- ISP A
- ISP A
- ISP A
- Power A
- Power B

Cloud B
- ISP B
- ISP B
- ISP B
- Power A
- Power B

Cloud C
- ISP B
- ISP B
- ISP B
- Power C

iRec

PSI-CA

ISP A
- Power A

Power B

ISP B
- Power A

Power C
Step 3 & 4 with W-PSI-CA

Deployment

<table>
<thead>
<tr>
<th></th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud A, C</td>
<td>0</td>
</tr>
<tr>
<td>Cloud B, C</td>
<td>3</td>
</tr>
<tr>
<td>Cloud A, B</td>
<td>2</td>
</tr>
</tbody>
</table>

App Provider

Cloud A, C

Cloud B, C

Cloud A, B

ISP A
ISP A
ISP A
Power A
Power B

Computer

ISP A

Power A

Computer

ISP B
ISP B
ISP B
Power A
Power B

Computer

ISP B

ISP B

ISP B

Power C

Computer

ISP A

Power A

Computer

ISP B

Power B

Computer

ISP B

Power C

Computer
Step 5

| Deployment    | |n| |
|---------------|---|---|
| Cloud A, C    | 0 |
| Cloud B, C    | 3 |
| Cloud A, B    | 2 |

App Provider  

iRec  

Cloud A  

Cloud B  

Cloud C
App Provider

Cloud A

Cloud B

Cloud C

Deployment

<table>
<thead>
<tr>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud A, C</td>
</tr>
<tr>
<td>Cloud B, C</td>
</tr>
<tr>
<td>Cloud A, B</td>
</tr>
</tbody>
</table>

Step 5

iRec
Step 5

App Provider

iRec

<table>
<thead>
<tr>
<th>Deployment</th>
<th>|n|</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud A, C</td>
<td>0</td>
</tr>
<tr>
<td>Cloud A, B</td>
<td>2</td>
</tr>
<tr>
<td>Cloud B, C</td>
<td>3</td>
</tr>
</tbody>
</table>

Cloud A

Cloud B

Cloud C
Step 5

App Provider

Cloud A

Cloud B

Cloud C

iRec

### Ranking List

| Deployment     | |n| |
|----------------|---|---|
| 1. Cloud A, C  | 0 |
| 2. Cloud A, B  | 2 |
| 3. Cloud B, C  | 3 |

### Deployment

| Cloud          | |n| |
|----------------|---|---|
| Cloud A, C     | 0 |
| Cloud A, B     | 2 |
| Cloud B, C     | 3 |
Step 5

Rating List

<table>
<thead>
<tr>
<th>Deployment</th>
<th>(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Cloud A, C</td>
<td>0</td>
</tr>
<tr>
<td>2. Cloud A, B</td>
<td>2</td>
</tr>
<tr>
<td>3. Cloud B, C</td>
<td>3</td>
</tr>
</tbody>
</table>

V.S.

Rating list with W-PSI-CA

<table>
<thead>
<tr>
<th>Deployment</th>
<th>(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Cloud A, C</td>
<td>0</td>
</tr>
<tr>
<td>2. Cloud B, C</td>
<td>1</td>
</tr>
<tr>
<td>3. Cloud A, B</td>
<td>2</td>
</tr>
</tbody>
</table>

Rating list with PSI-CA
Road-Map

- Motivations
- Goal & Insight
- iRec System
- Next Steps
Road-Map

- Motivations
- Goal & Insight
- iRec System
- Next Steps
Next Steps

• Can we provide stronger privacy preservation?
Next Steps

• Can we provide stronger privacy preservation?
• Do cloud providers have incentives to join?
Next Steps

• Can we provide stronger privacy preservation?
• Do cloud providers have incentives to join?
• Will the clouds behave honestly?
Next Steps

- Can we provide stronger privacy preservation?
- Do cloud providers have incentives to join?
- Will the clouds behave honestly?
- Can we make iRec more scalable?
- How do we evaluate iRec with realistic cloud dependency datasets?
The End

• Do you want to:
  - do a cloud related final project?
  - do anything for EverClouds?
  - discuss your smart idea about the clouds?
The End

• Do you want to:
  - do a cloud related final project?
  - do anything for EverClouds?
  - discuss your smart idea about the clouds?

I have a lot of interesting (even crazy) ideas about the cloud security and reliability, but I do not have time to implement all of them ...
Thanks!

Questions?