COMET: A Distributed Metadata Service for Federated Clouds

Cong Wang, *Komal Thareja, Michael Stealey,*
Paul Ruth, Ilya Baldin
RENCI, University of North Carolina at Chapel Hill

renci
RESEARCH \ ENGAGEMENT \ INNOVATION

Today's Research Clouds

- Globally distributed, multi provider, individually operated
- Example: Three NSF funded research clouds
- GENI/ExoGENI
  - Distributed/Networking focused
  - 18 US sites
- Chameleon Cloud
  - High capacity, reconfigurable cloud for repeatable science experiments
  - 2 sites: Illinois, Texas
- CloudLab
  - Flexible, distributed scientific cloud
  - 3 sites: Utah, Wisconsin, South Carolina
- Challenge: Difficult to provision resources across multiple clouds

renci

Toward Multi-Cloud Research Infrastructure

- Current trend: go multi-cloud
  - Enhanced compute and networking performance
  - Reduced cloud resource prices
  - Emergence of IoT and edge computing
  - Diverse compute and networking needs
- Federated clouds
  - User may integrate multiple cloud resources in single reservation
- Multi-cloud market place
  - User may choose resources from multiple clouds based on needs (e.g., geographical location)

renci

Metadata for Clouds

- Metadata services store basic information about resources
  - Instance ID, region (location), project ID
  - Network interfaces (Mac/IP address)
  - Instances' configurations (e.g., CPU, RAM)
  - Storage (e.g., NFS)
- Configuration data
  - SSH-keys
  - Host names
- Application data
  - Scripts to run when nodes are alive
  - Routes

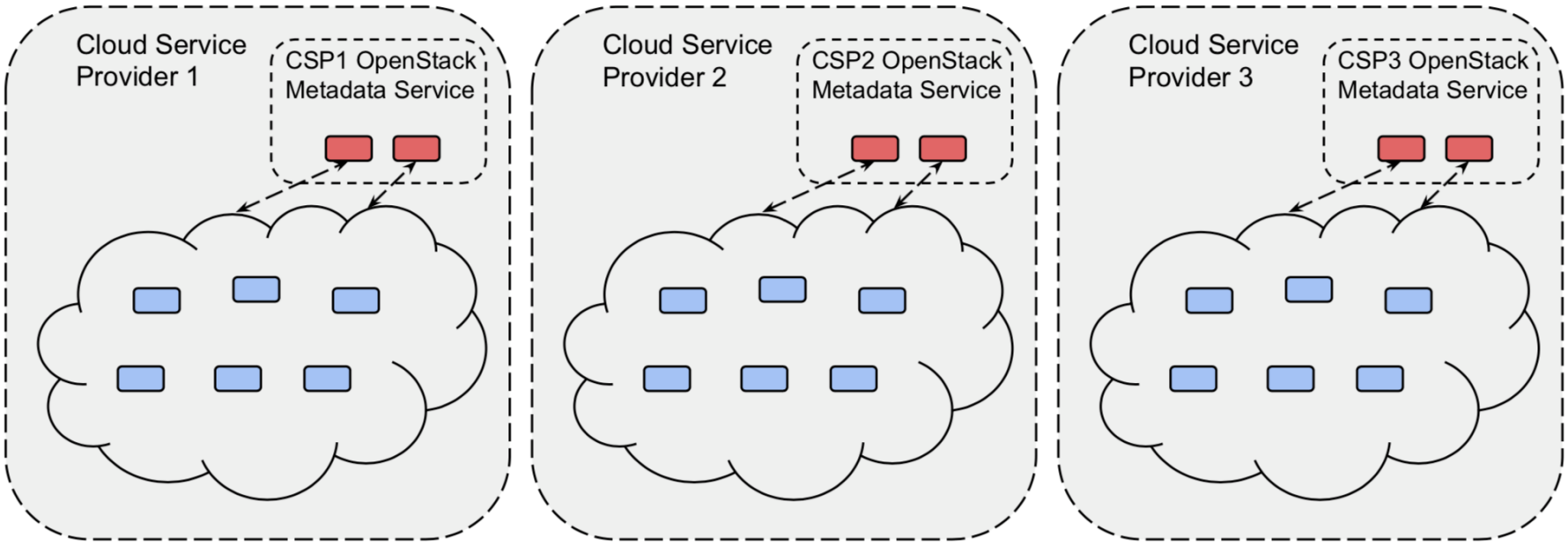
renci

COMET: A Distributed Metadata Service for Multi-Cloud Infrastructures

- Designed to store metadata for applications running on multiple clouds
- Who needs to access metadata?
- Users
  - Tenants responsible for creating VMs or slices
  - Other users with shared access
- Cloud provider agents
  - Such as Controllers or Aggregate managers
- Tenant infrastructure controllers
  - Such as SDN controllers
- Applications running inside compute nodes
  - Such as Hadoop and HTCondor

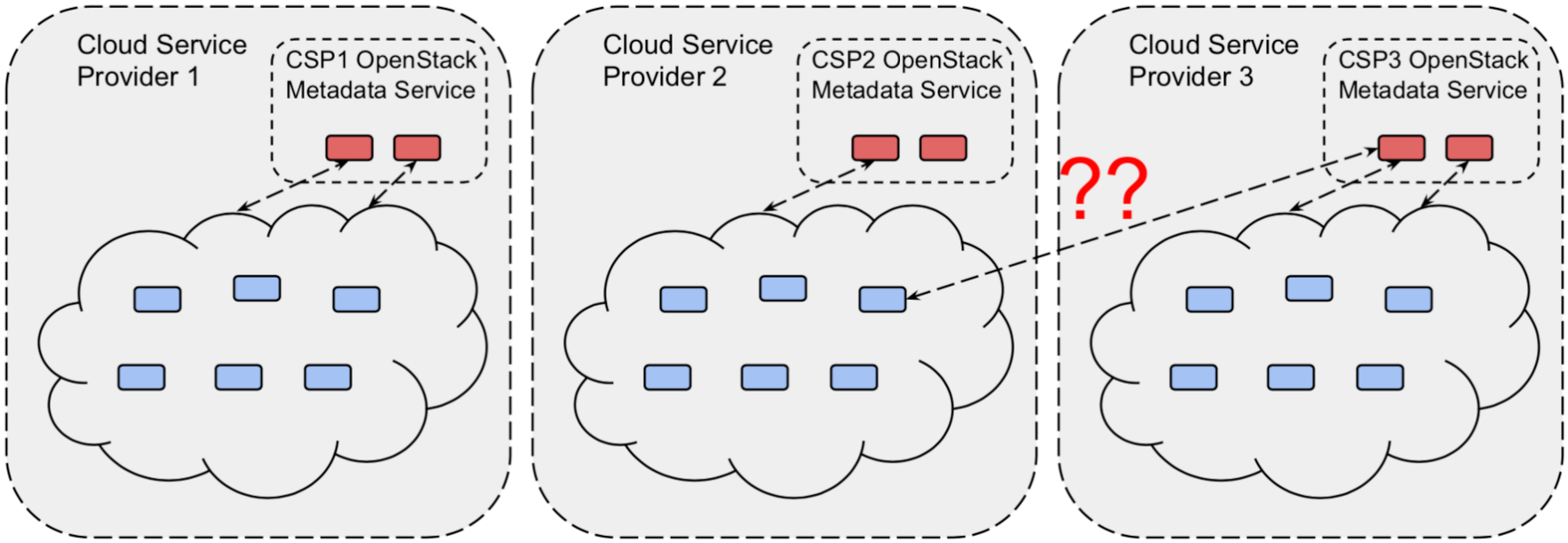
renci

Existing Metadata Service: OpenStack



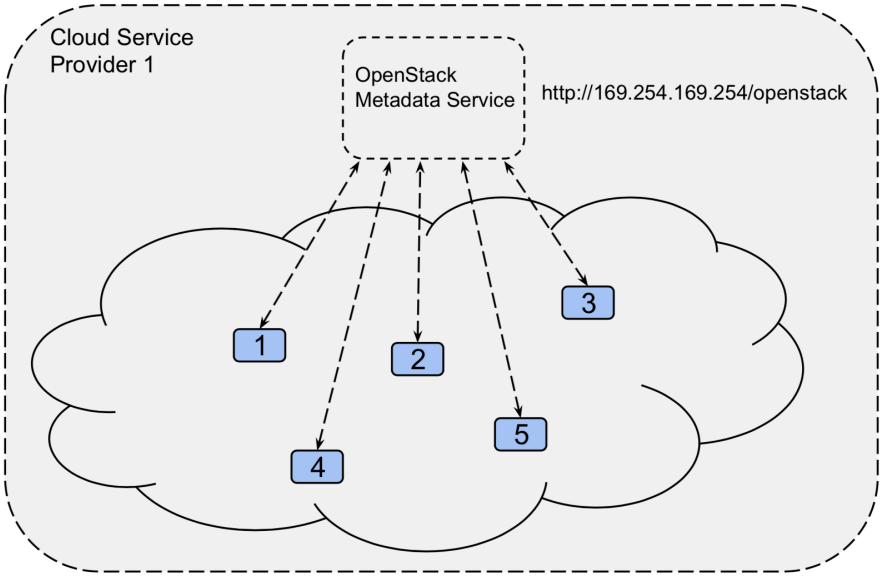
renci

What if querying metadata between clouds?



renci

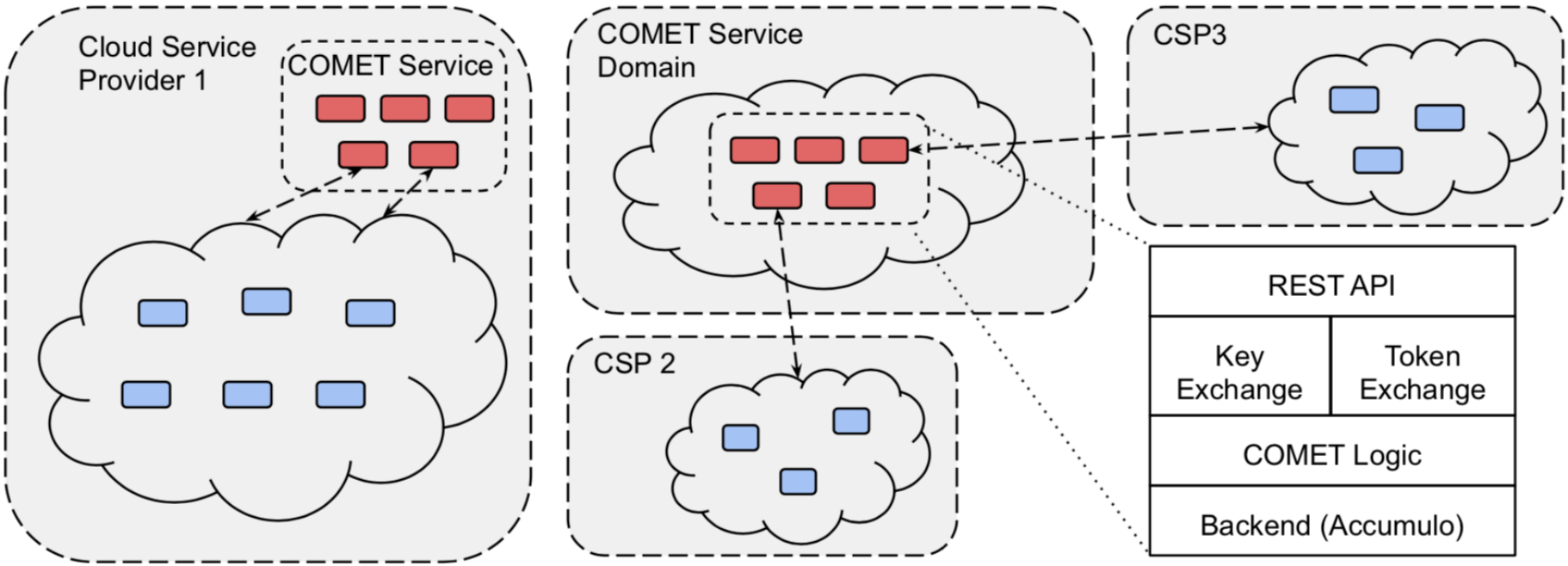
OpenStack Metadata Service



- No password-based access options
- A node can only retrieve its own metadata
- Difficult to sync metadata among cluster of nodes
  - Key, IP, hostname exchange is usually needed by distributed applications
  - Hadoop
  - Condor
  - MPI

renci

COMET Architecture



renci

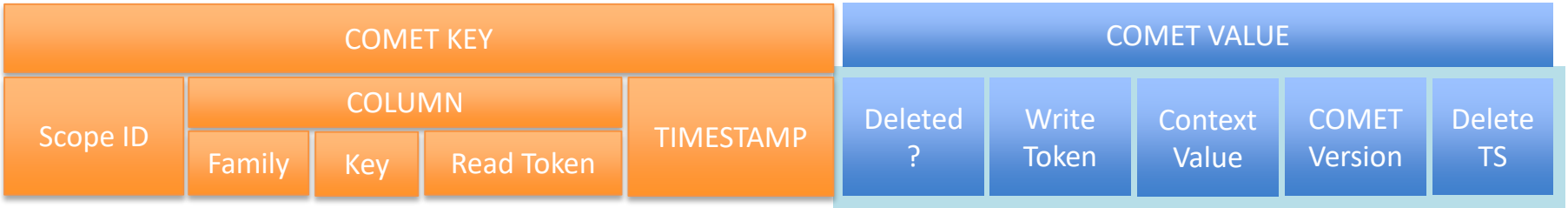
Example Naming Hierarchy for COMET API

- ScopeID
  - Unique ID of the scope of resources (slice or sliver)
- Family
  - User-defined string
- Key
  - User-defined sub-level string
- Read Token/Write Token
  - Tokens needed for read and write access
- Value
  - Single value or a serialized byte array

renci

COMET Data Model

- ScopeID – unique ID of a slice or sliver
- Family – user-defined string with user-imposed semantics
- Key – user-defined string with user-imposed semantics
- Value – single value or a map
- Read token – client defined ‘visibility’ tag
- If Deleted, Write Token, Context Value, Comet Version, Deletion Timestamp



renci

COMET Operations

- WriteScope
  - Create or modify a named scope within a context
- DeleteScope
  - Delete scope within a context
- ReadScope
  - Retrieve a value from a named scope within a context
- EnumerateScopes
  - Return a list of existing scopes within a context

renci

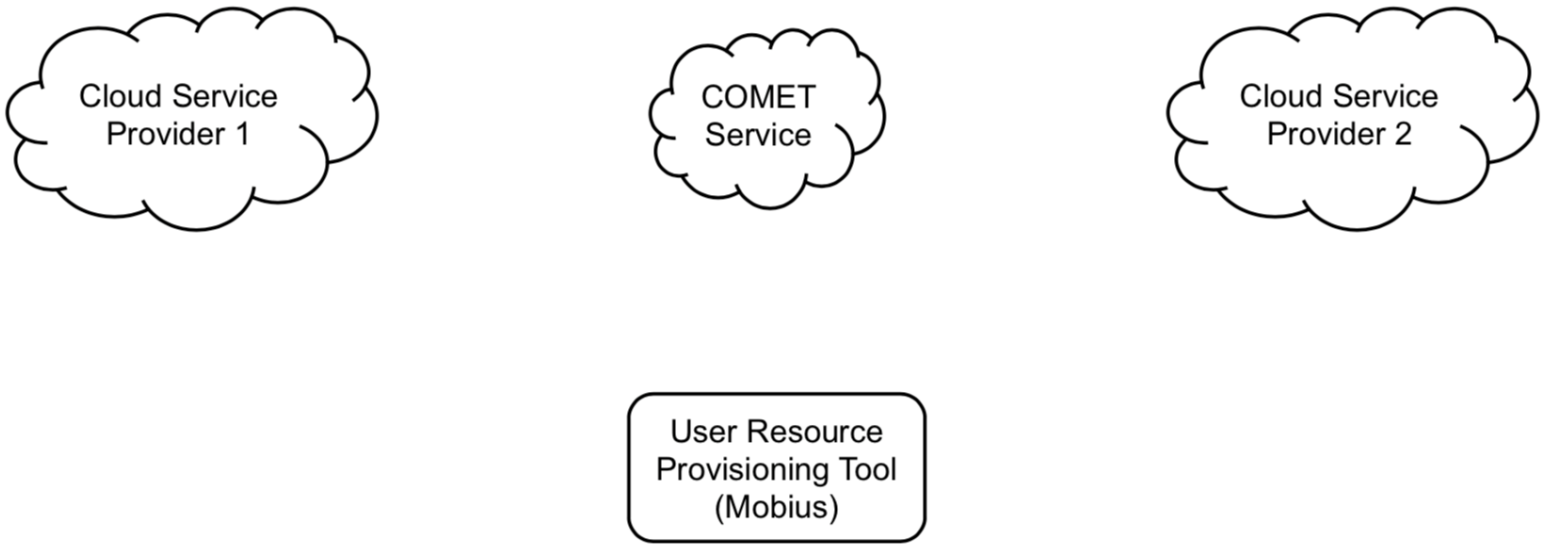
COMET APIs and Cert Requirements

API Call	Semantics	Valid Cert	Trusted Cert	Read Token	Write Token
`WriteScope`	create new entry	-	V	N	N
`WriteScope`	modify existing entry	O	-	V	V
`ReadScope`	read existing entry	O	-	V	-
`EnumerateScope`	enumerate entries in a scope	O	-	V	-
`DeleteScope`	delete an existing entry	-	V	V	V

COMET API calls and certificate requirements. V = validate if a client cert/or token is trusted, N = specify a new token, O = a valid client cert is optional.

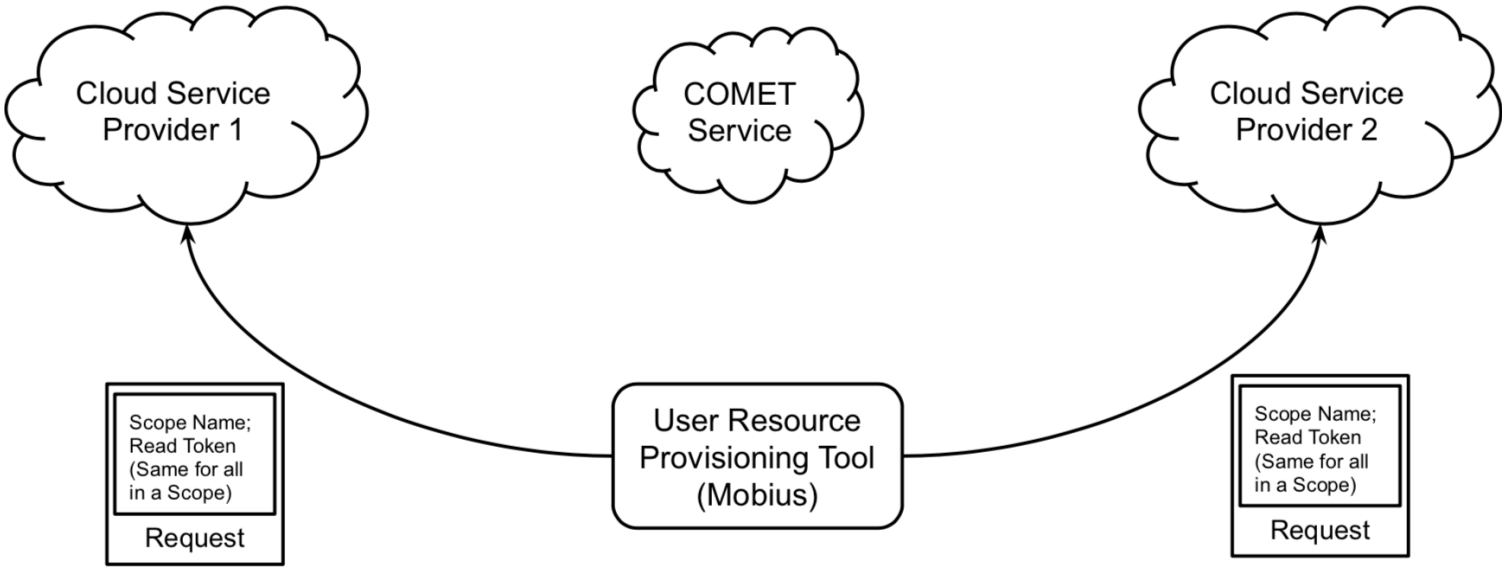
renci

User Metadata Exchange



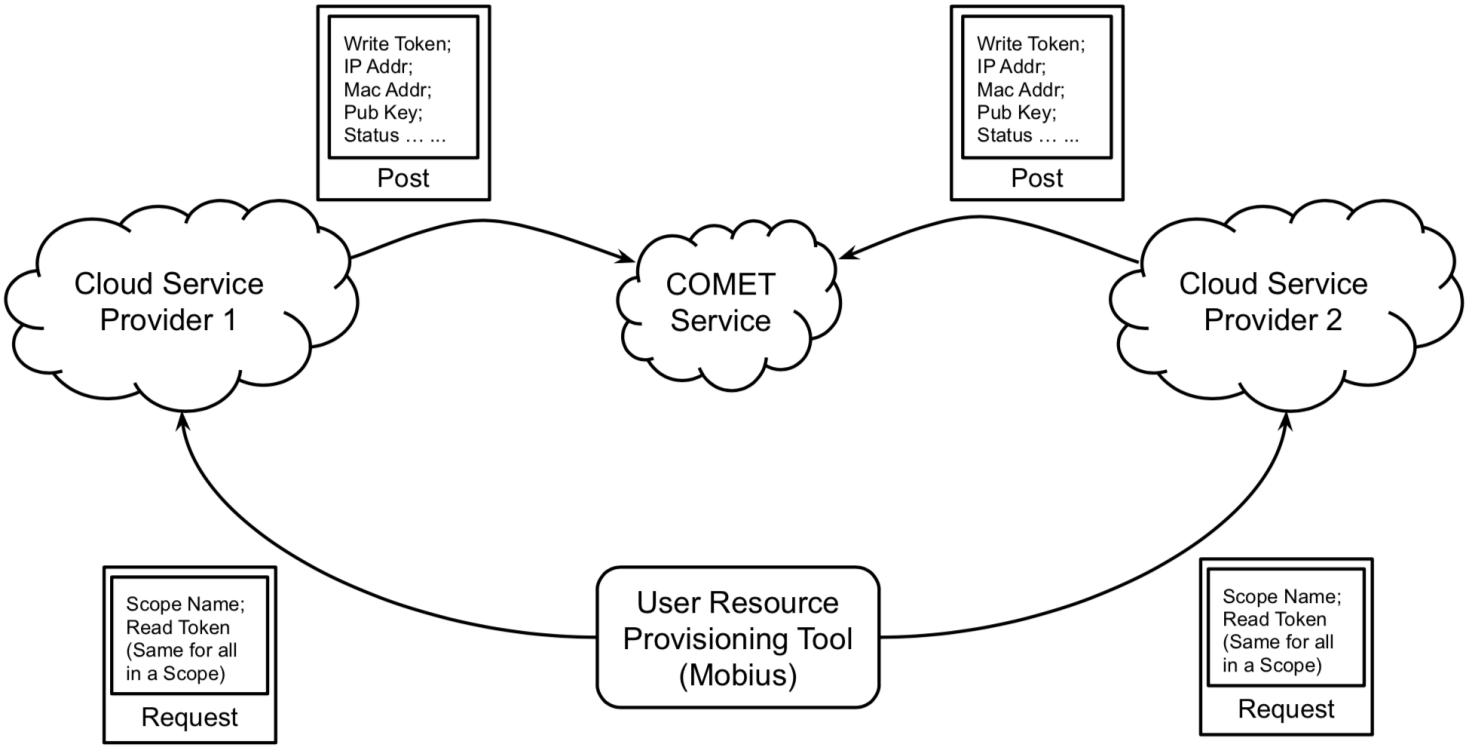
renci

User Metadata Exchange



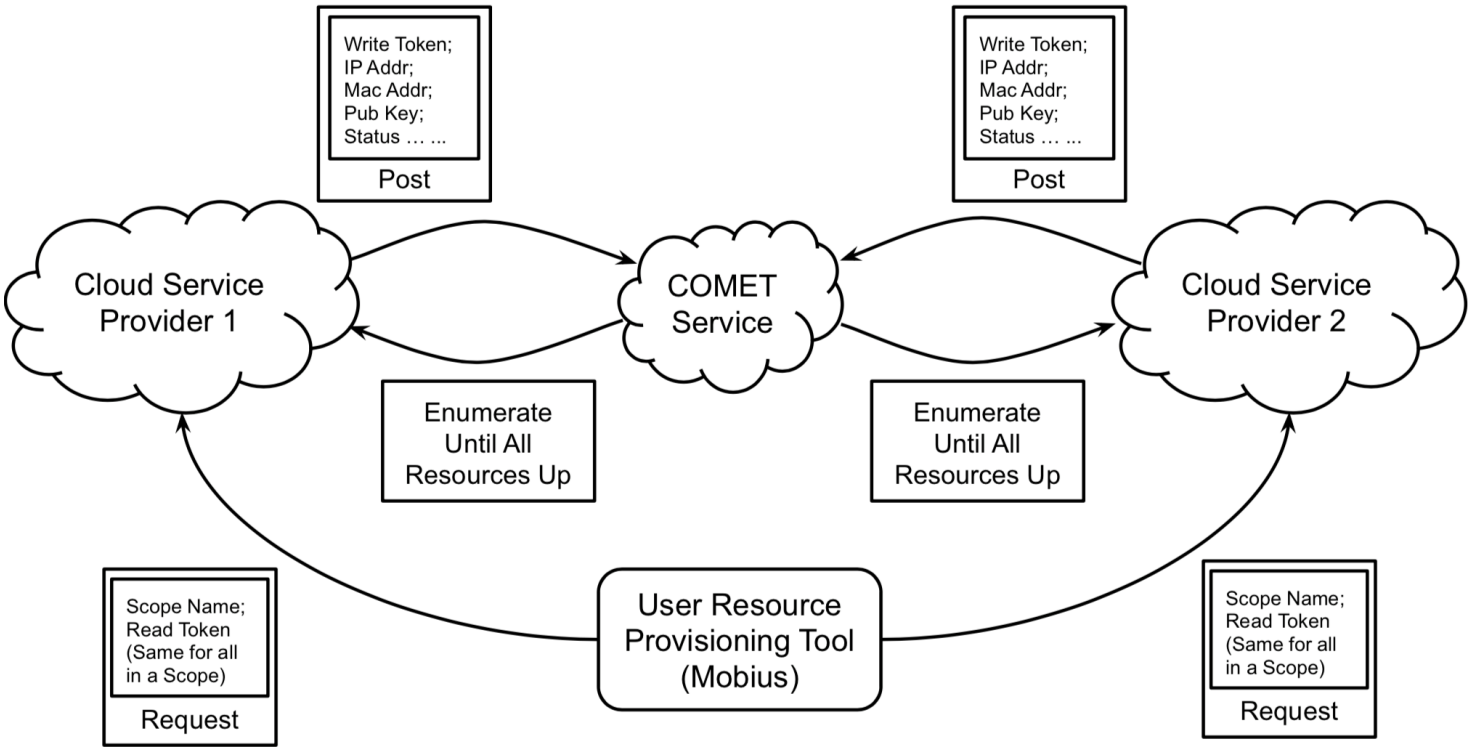
renci

User Metadata Exchange



renci

User Metadata Exchange



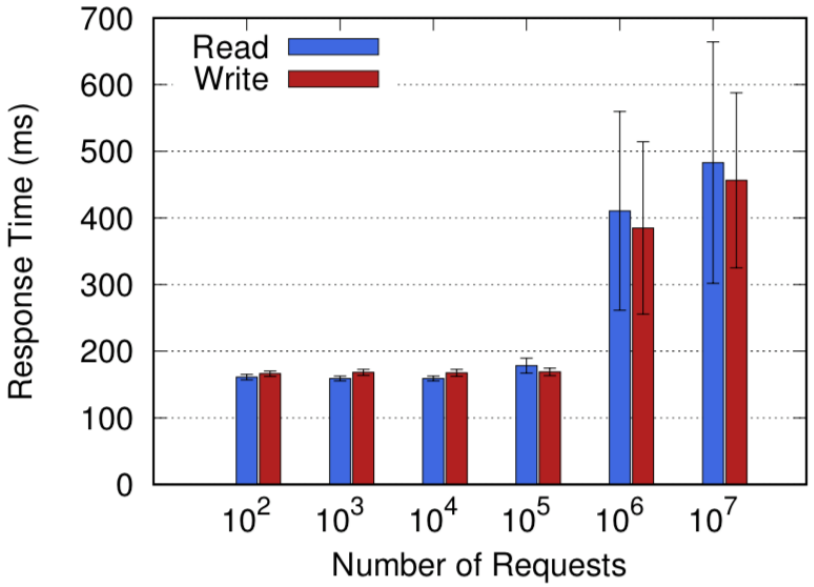
renci

Evaluations

- COMET hosted on AWS
  - US east Ohio region
  - EC2 t2.large compute nodes
  - 2 vCPUs, 8 GB RAM
  - Three COMET head nodes (Round Robin)
- Read and Write tests
  - Sequential and random R/W
  - Number of requests 100 -- 10^7

renci

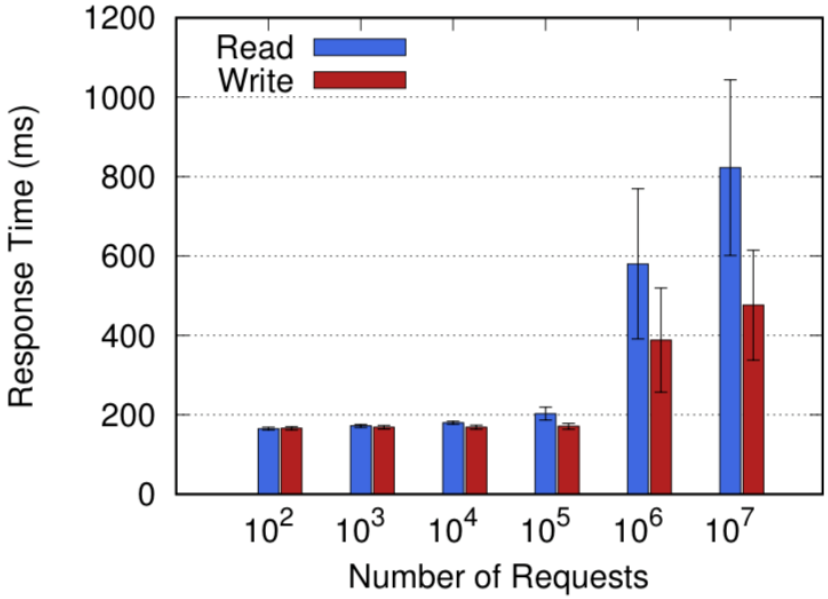
Evaluations – Sequential R/W



- Similar read and write speed
- Similar response time with less than 10^5 requests
- Significant longer response time with more than 10^5 requests

renci

Evaluations – Random R/W



- Similar read and write speed with less than 10^5 requests
- Similar response time with less than 10^5 requests
- Longer read time (1.5x) with more than 10^5 requests

renci

Conclusions

- COMET: metadata management service that focuses on security and flexibility for multi-cloud applications
- Discussion on design, implementation and evaluation of COMET services
- COMET open-source code base:
  - https://github.com/RENCI-NRIG/COMET-Accumulo/releases/tag/comet-spring-1.0.0

renci

Thank you!
Questions?

renci