An Overview on Client-Centric Cloud Computing

Károly Bósa, Roxana Chelemen, Harald Lampesberger, Mariam Rady, Boris Vleju

The Christian Doppler Laboratory for Client-Centric Cloud Computing

June 24th, 2014

Outline

Introduction

- Nowadays "cloud computing" is one of the most often used buzzword in computing
  - Many providers (Amazon, Google, Microsoft, IBM, etc.),
  - Many kind of cloud services(IaaS, SaaS, PaaS, DaaS, . . .) and
  - Many benefits of outsourcing application into a (private or public) cloud.
  - **Is it a mature technology? Is ready to be massively used?**

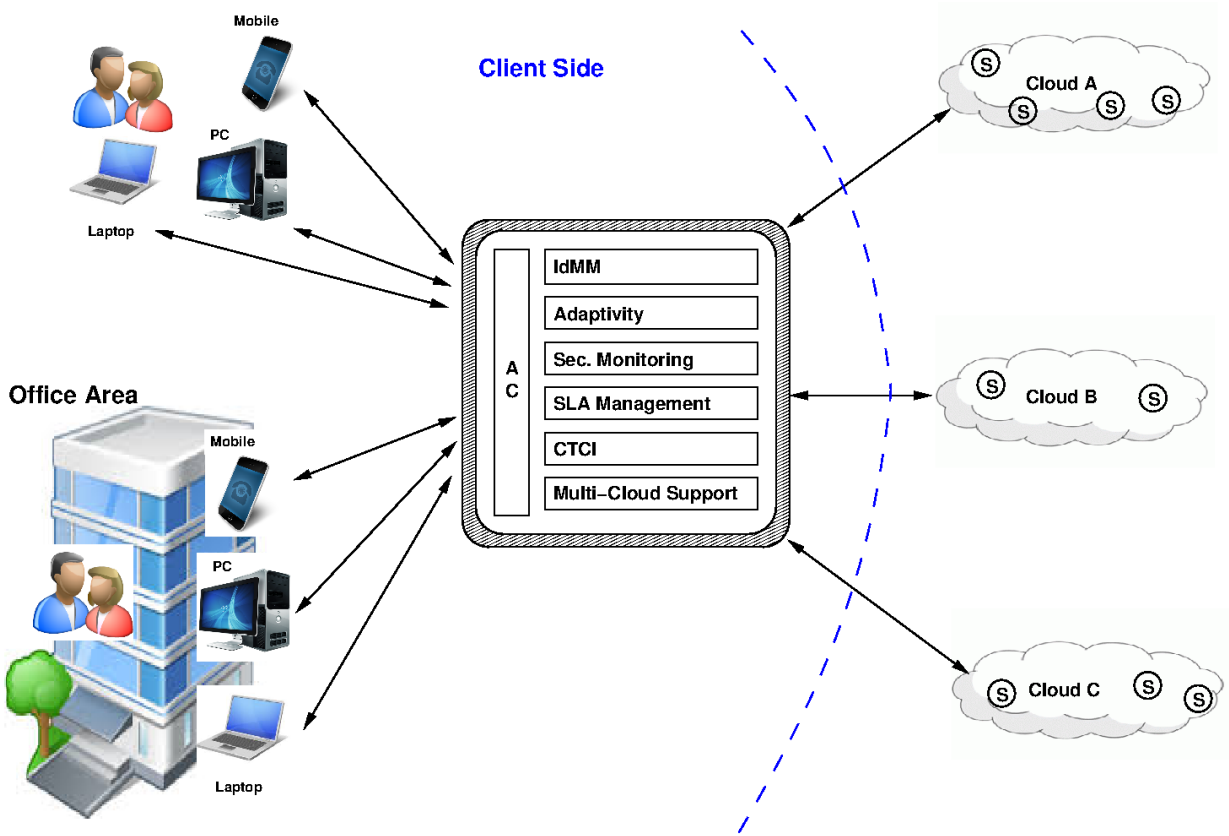
Introduction

- Our Conviction (at CDCC) cloud computing still requires lots of fundamental research.

  Among many other problems we identified e.g.:

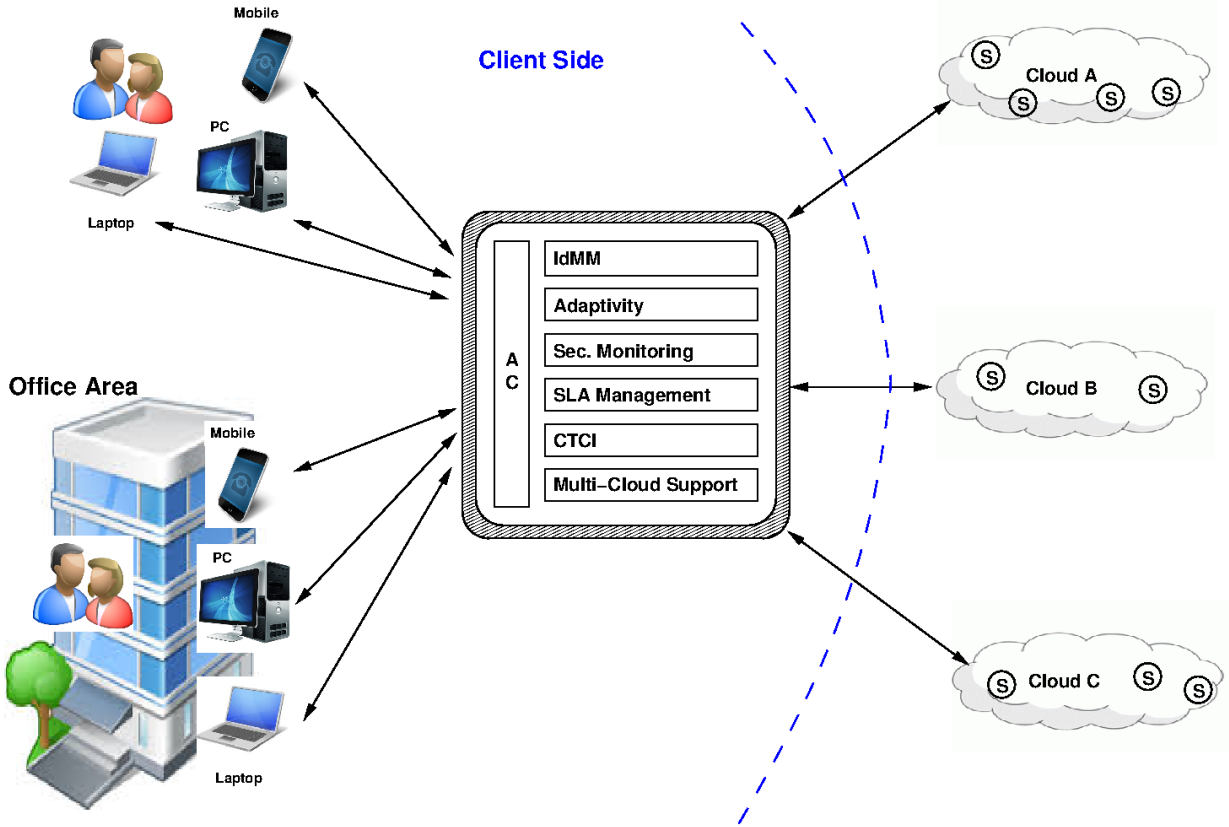
  - **lack of client orientation (tenants/customers loose the control over their intellectual properties)** and
  - lack of formal models and foundation (e.g.: notion of key components like *services* not defined).

Middleware



- The components of this *client-controlled interaction middleware* are only loosely coupled (they are independent from each other).

Middleware



- Since middleware is located on the client side, it can be easily adapted to nowadays leading cloud architectures (e.g.: Amazon S3, Microsoft Azure, IBM SmartCloud, etc.).
- The feasibility of the developed proof of concept solutions is tested on OpenStack cloud infrastructure software on an IBM CloudBurst machine.

Security&Privacy → Access Control (AC) based on Service Plots

- **Goals:**
  - Providing a transparent and uniform access control mechanism for cloud services without giving up the flexibility of heterogeneous cloud access to these services,
  - via which owners or clients of cloud services are able to fully control the usages of their cloud services for their end users.
- **Advantage of our Approach:**
  - Not only certain service usage can be allowed, but their **permitted combinations (plot)** of their functions can be composed individually for end-users by clients.

Use Case(s)

- We target a new kind of AC (cloud) service
  - whose customers are enterprizes
  - who own or pay for various cloud services provided by one or multiple clouds and
  - who want to have direct influence how these cloud services are accessed and used by their end-users (employees or partners).

Security&Privacy → Dynamic Context-Aware Access Control

- The Access Control component should also consider the context constraints in organizational settings:
  - temporal context of a particular task execution (which time a request was triggered),
  - spatial context of a particular task execution (from where and on which end-device a request was triggered),
  - user-specific attributes (e.g.: the task execution history of a user).
- Work related to this topic just has been started...

Use Case(s)

- Nowadays, due to modern mobile devices and cloud technology
  - difficult to manage devices and the cloud
  - difficult to protect sensitive data and
  - almost impossible to separate the treatment of corporate and private identities, etc.
- Because of this lots of people are simply no longer willing to deal with all these things and they use everything everywhere, e.g.:
  - they work from home or from public places (e.g.: cafes,airports, etc.),
  - they read the corporate e-mails on their (private) mobile phones and/or
  - they store business critical data in Dropbox, etc.
- A cloud-enabled context-aware RBAC solution could handle these issues properly.

Security&Privacy → Identity Management Machine (IdMM)

- **Goal:** It is an identity management facility whose purpose is
  - to make the adoption or migration of user accounts to cloud services faster by handling identities of end users and providing a privacy-enhanced solution.
- Can be used as an integrated service or as a stand-alone service
- All user identity data is stored in a directory which is located on-premise

Advantages of our Approach

- Provides a single sign on proxy for the user
- Automatically authorizes and authenticates the user to a given service
  - Uses real and obfuscated identities for authentication
  - A user may not be aware of his or her credentials to the cloud service.
- Manages user provisioning/de-provisioning
  - Creates, edits or removes cloud-based accounts (identities)
  - Periodic password resets of these accounts
  - Creates, edits and removes client side accounts
  - When a user leaves the company:
    - Only the client-side identity is removed
    - The cloud-based accounts may be re-used for obfuscated cases
    - Since the user does not know the credentials he or she cannot use the service
- Acts as an identity provider

Security&Privacy → Intrusion Detections

- **Goal:**
  - Our intrusion detection system is going to monitor the effectively communicated protocol language to detect anomalies in client-cloud interaction.
  - Anomalies can correlate to attacks that exploit software vulnerabilities on the client- or service-side.

- **Advantages of our Approach:**
  - State-of-the-art systems match for known patterns or symptoms. But modern attacks try to evade detection.
    - Nevertheless, an attack affects the protocol language, therefore constructing specific language models for accessed cloud services and **learning to recognize deviations from the modeled intended usage** is possible.
  - For now we focus on automaton-based language models for XML-based protocols.
    - The automaton captures the intended common structure of XML documents and content.

Use Case(s)

- An intrusion detection component in the middleware detects anomalies in interaction between clients and services.
  - Depending on the the severity of the detected event, access control rules are changed dynamically to minimize risk.

Client Needs → Content Adaptation

- **Goal:** Providing on-the-fly adaptivity of cloud applications to different services, devices (e.g.: smartphones, tablets, laptops), preferences and environment.

- **Features/Advantages:**
  - No specific application for each type of device (only one (Web) application for all devices).
  - The content of a web page is adapted on the middleware corresponding to the properties of the device detected on the client-side.
  - The application is based on an abstract model built from the requirements(using Abstract State Machines).

Use Case(s)

- Nowadays people focus a lot on flexibility at work:
  - Content Adaptation could be used by some companies to make the necessary applications available to their employees also outside the office.
  - The employees could continue their work from home or while traveling using the devices they have at hand (mobile, tablet or laptop) and this without installing specific applications.
- A working scenario:
  - A database manager application is deployed on the Cloud.
  - The users (e.g.: employees) access the corresponding services (e.g.: create, read, update, delete) using different devices through the web application.
  - The services are always adapted to the properties of the device.

Client Needs → Client-to-Client Interaction (CTCI) in Cloud Computing

- **Goal:** The realization of our Client-to-Client Interaction (CTCI) mechanism can be regarded as a special kind of services we call *channels*
  - via which registered cloud users can interact with each other in a direct way
  - And they are able to share available cloud resources.

- **Advantages of our Approach:**
  - Cloud Service Transparency

Use Case(s)

1. Connecting devices of the *same* user via the cloud employing transport local area protocols.
2. Switching services: the role the middleware plays in this case is to switch service functionalities and stored data from one user/device to another (e.g.: dissemination of multi-media data such as photos or videos).
3. Anonymous cloud service usage:
   - Since all the shared services are used on behalf of its initial distributor
   - no traces of the user activities belonging to the shared services will be left on the cloud,

Client Needs → Supporting Multi-Cloud Applications

- **Goal:** Providing functionalities
  - which enables many-to-many relationship between cloud service providers and clients

- **Advantages of our Approach:** development of Service-Centric Applications (SCA) is made possible:
  - where some clients/customers of the middleware may combine several services offered by different service providers located on various clouds in some intelligent way and
  - then offer them as *new services*.

SLA Management

- **Goal:** Developing approaches, tools and mechanisms used in different phases of the SLA lifecycle
  - Creation Phase: the formal definition of SLAs
  - Operation Phase: monitoring of service execution adherence to the agreed terms.
  - Termination Phase: ending the agreement

  A Service Level Agreement is a contract to agree on the provided service level between the provider and the consumer.
- **Advantages of our Approach:**
  - expressive, structured and machine readable language for SLA definition.
  - client-centric (user-oriented) SLA, making the contracting process more realistic.
  - model captures abstract client-cloud interaction, on which the monitoring tools can be based.
  - removing the burden from the user to recognize breaches of the SLA, notify the service provider and request credit.

Use Case(s)

- An SLA is comparable to a warranty. It guarantees the service as a guarantee promises your car won't break down.
- Nowadays, SLAs are defined by the service provider and the client has no contribution in the contracting process.
  - Limited liability of the provider (e.g.: it is the clients responsibility to recognize SLA violations and report them to the provider.
- SLAs cannot make a good service out of a bad one, however, they can mitigate the risk of choosing a bad service.

SLA Management → Creation Phase-SLA Template Definition

- We defined the SLA using an ontology, OWL, DL.
- In order to develop the ontology:
  - defining the context in which the ontology is going to be used(client-centric, service model, requests).
  - defining top level concepts
  - refining the different concepts to include subconcepts and relations between the different concepts.
  - SLA Ontology

SLA Management → Creation Phase-SLA Instantiation

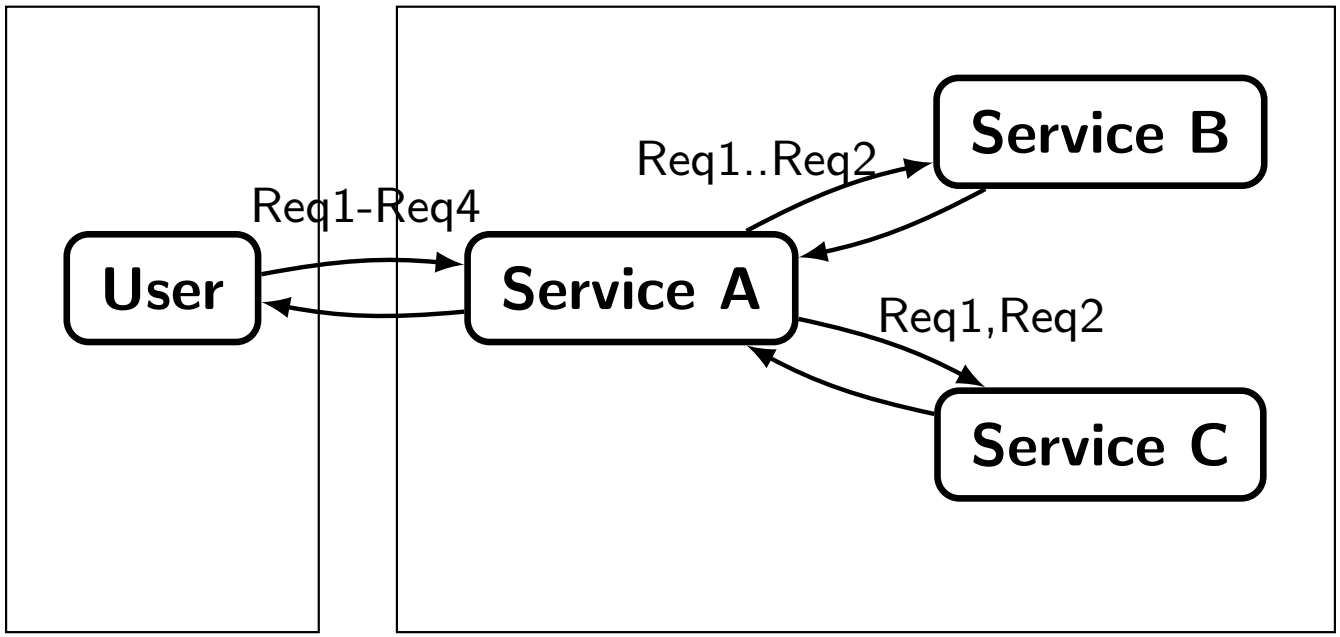


Figure: Example Scenario

SLA Management → Operation Phase-SLA Monitoring

- Monitor the compliance of the SLA to the service
- Proof that the SLA ontology is sufficient → Refine again.
- We focus on Availability, Performance and Security.
- We need to have an analysis component to measure availability and performance of a cloud service to be able to add this information to the SLA and check for the service level compliance to the SLA.

Conclusion

- Client Centric Cloud Computing
- Middleware based approach
  - Security and Privacy (Access Control, Identity Management, Intrusion Detection)
  - Client Needs (Content Adaptation, CTCI, Multi-Cloud Support)
  - SLA Management
- a short dive into SLA Management.

Contacts → Thank You for Your Attention

- Karoly Bosa (k.bosa@cdcc.faw.jku.at)
  - Access Control based on Service Plots
  - Client-to-Client Interaction in Cloud Computing
  - Supporting Multi-cloud Applications
- Roxana Chelemen (roxana.chelemen@cdcc.faw.jku.at)
  - Content Adaptation
- Harald Lampesberger (h.lampesberger@cdcc.faw.jku.at)
  - Intrusion Detections
- Mariam Rady (m.rady@cdcc.faw.jku.at)
  - SLA Management
- Boris Vleju (b.vleju@cdcc.faw.jku.at)
  - Dynamic Context-Aware Role-Based Access Control
  - Identity Management Machine (IdMM)