How to buy a Cloud

Vassilis Prevelakis

Institut für Datentechnik und Kommunikationsnetze
Technische Universität Braunschweig
Concepts

• Why open system of cloud resource provisioning?
  – provider independent.
  – allows combinations of different providers.
  – better allocation of resources.
  – cloud resources are perishable (use it or lose it).

• If its such a good idea, why aren’t we using it already?
  – Customer lock-in.
  – No common architecture for accounting and charging.
  – API issues
Motivation

- Assuming that the various resources are owned by different organizations, how can the customer construct a virtual computation platform from these resources?
- Voucher analogy
Cloud Exchange

• The Cloud exchange is an on-line site where
  – cloud providers place quotes for their available cloud resources
  – customers browse for offers for cloud resources

• Futures Market
  – advance bookings

• Spot Market
  – used to allocate resources that have not been previously reserved.

• Requirements
  – transparency (to ensure fairness).
  – assurances (to both buyer and seller).

• Supports overlay providers (delegation)
Example of Spot Market Operation
The Clearing House acts as a repository of all the offers for bandwidth issued by the various cloud providers (CPs).
Customer finalizes the path selection by downloading the offer credentials.
The customer issues a request by sending the offer credentials collected from the CRX Clearing House along with a credit-worthiness credential issued by his or her credit institution.
Configuration complete:
Storage servers from CP5,
Computation server from CP2 and
Communication links from ISP 1
CRX Architecture (5)

• Operation of the Futures Market
  – In the spot market, offers have immediate effect.
  – In the Futures market, offers take effect in the future.
    • thus CPs must be informed in advance.
  – Carry out “notional” negotiation (same as spot market).
  – Reservation Credential(s) sent to the user.
  – When resource is required (within the reserved period), user initiates reservation process.
    • in this case only the reservation credentials need be sent.
Implementation (1)

- Trust Management Framework
  - credentials contain public keys of *authorizer*, *licensee*, and *conditions* (code that describes what is contained in the credential)
  - Credential signed by the *authorizer*.

Keynote-Version: 2
Local-Constants: ALICE KEY = "rsa-base64:MCgCIQGB0f8..."
  CG KEY = "rsa-base64:MIGJAo..."
Authorizer: CG KEY
Licensees: ALICE KEY
Conditions: app domain == "CRX" && currency == "USD"
  && &amount < 5.01 && date < "20150824" -> "true"
Signature: "sig-rsa-sha1-base64:QU6SZtG9R3IXXAU9vRDBgu..."
Implementation (2)

- Trust Management Framework
  - Each entity trusts itself.
    - basic policy allows other entities to be trusted (conditionally)
    - additional credentials allow this trust to be extended (conditionally).
  - For a request to be granted it must be consistent with existing policy.
    - otherwise the request must supply credentials to extend the policy.
    - if not, the request will be denied.
  - Keynote library allows credentials to be verified and integrated into the policy automatically.
Example

- Vendor wants to give discount to “preferred” customers
  - needs to instruct vending machines to give 10% discount to authorized users
  - usual problems
    - list of users not known in advance, lots of vending machines, etc.
- Vendor issues a credential to each “preferred” customer
  - command example (pseudo code)
    if (date < 20151231 && price > $1) then price = price * 0.9
  - Authorizer is vendor (signs credential)
  - Licensee is customer’s public key
Trust Model (2 level)

From: Cola Co
To: Alice
price = price * 0.90

From: Alice
To: VM-12
dispence 1 Cola can

Credentials Alice will send to VM-12

Chain of trust from Alice to VM-12

Alice

Cola Co
(trusts Alice to get 10% discount)

Vending Machine 12
(trusts its owner, i.e. the Cola Co)
Trust Model

We can add as many levels as we want

Credentials Alice will send to VM-12

<table>
<thead>
<tr>
<th>From: Alice</th>
<th>To: VM-12</th>
</tr>
</thead>
<tbody>
<tr>
<td>dispence 1 Cola can</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>From: Cola Co</th>
<th>To: Alice</th>
</tr>
</thead>
<tbody>
<tr>
<td>price = price * 0.90</td>
<td></td>
</tr>
</tbody>
</table>

Cola Co (trusts Alice to get 10% discount)

Vending Machine 12 (trusts its owner, i.e. the Cola Co)

Chain of trust from Alice to VM-12
Example

• Customer establishes session with vending machine
  – any mechanism is acceptable
  – can deal with issues such as replay attacks, secrecy etc.
• Customer sends request with discount credential
• Vending machine
  – matches request signature with public key included in vendor credential
  – evaluates request
  – declares outcome (final sale price)
  – note that payment can be effected using trust management (micropayments) or other traditional form (e.g. coins)
Implementation (3)

From: Cloud Provider
To: Alice
Allow Alice to allocate ram < 12Gb on CPU-12

From: Alice
To: CPU-12
set RAM=5Gb, boot ...

Cloud Provider
(trusts Alice to manage ram < 12Gb on CPU-12)

CPU-12
(trusts its owner, i.e. the Cloud Provider)

Credentials Alice will send to CPU-12

Chain of trust from Alice to CPU-12
Problems

• Overbooking
  – CPs do not know in advance how many offers will be exercised

• Loss of Quality
  – What happens if a provider fails to deliver on their promises

• Revocation
  – What if one or more actors change their minds?
Conclusions

• Model accommodates both “instant” purchases of bandwidth and advanced purchases
  – CPs can plan ahead their resource allocation strategies
  – CPs can get better prices for unused capacity.
• The entire protocol is efficient requiring only a few exchanges between a buyer and various sellers to effect a reservation.
• CRX system can be deployed with minimum disruption.
Conclusions

• Credit Institution(s) link buyers and sellers
  – transactions can take place between buyer and seller without previous business relationship.
  – Allows cloud resource market to work freely with the buyer being able to select the seller offering the best value for money.

• Keynote-based micro-payment framework makes entire system efficient and scalable.
Conclusions

- The CRX model allows the presence of multiple entities for each role (\textit{i.e.}, we can have multiple Credit Institutions, Clearing Houses, buyers and sellers) operating within a single market. This increases the competition and overall reliability of the entire system.