Gossiping meets Cloud Computing

Ozalp Babaoglu
Department of Computer Science
University of Bologna

- I will address the following in the context of cloud computing:
  - Achieving full decentralization
  - Achieving accountability and trust
- The first has to do with gossiping. The second does not

Towards full decentralization

- During the period 2003 - 2007, extensive use of "gossiping" at Bologna to construct fast, robust, decentralized solutions for
  - Aggregation
  - Overlay topology building and management
  - Heartbeat synchronization
  - Cooperation in selfish environments

- In 2006 we started thinking about an architecture built out of components that we had designed for *dynamically allocating* and *sharing* huge collections of commodity resources among many peer-to-peer applications
- Not unlike "multiplexing" a distributed infrastructure in a totally decentralized manner to create a p2p "timesharing" system
- Wrote up the idea as a position paper:

  O. Babaoglu, M. Jelasity, A-M Kermarrec, A. Montresor, M. van Steen. *Managing clouds: a case for a fresh look at large unreliable dynamic networks*, ACM SIGOPS Operating Systems Review, 2006

- Recently, we have revisited the idea and proposed the architecture as the basis for *peer-to-peer cloud computing*
- Extreme point in the spectrum of cloud computing architectures from centralized-to-federated-to-p2p
- The architecture inherits characteristics of p2p systems:
  - Total decentralization
  - Self organization
- "Poor man's" cloud computing platform

1 THE DATA CENTER THAT POWERS THE iCLOUD: Apple's 500,000 square foot Maiden, NC, facility was completed last year.

2 THE QUESTION MARK: Apple is building a 4.8 megawatt biogas-powered power generator right next to the data center. This looks like the spot it will go.

3 THE SOLAR ARRAY: This 100 acre solar power farm will generate 20 megawatts of power.

Peer-to-Peer Clouds

- In our *Managing Clouds* paper, we used the "cloud" metaphor to highlight *granularity* and *fluidity*:
  - huge number of water droplets or ice particles,
  - individually insignificant but aggregated significant,
  - in a state of flux with constantly changing boundaries,
  - yet, maintaining an identifiable shape

Peer-to-Peer Clouds

- What are these "water droplets" in practice?
- Range from set-top boxes to ADSL/Broadband modems to laptops to multi-core PCs
  - have onboard computing and storage resources,
  - are owned and operated by different individuals,
  - are physically located at individuals' homes,
  - remain "mostly on" but can be powered off or unplugged from the network
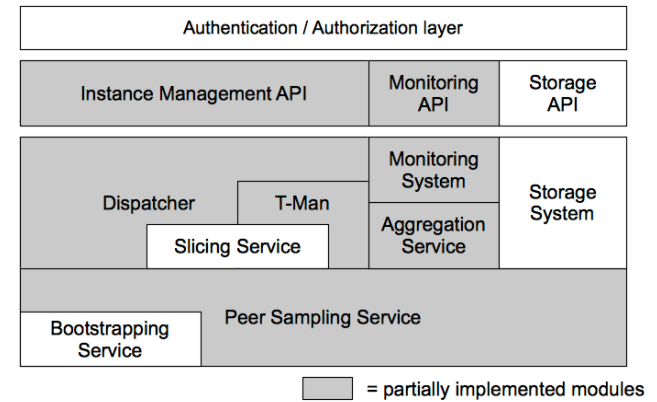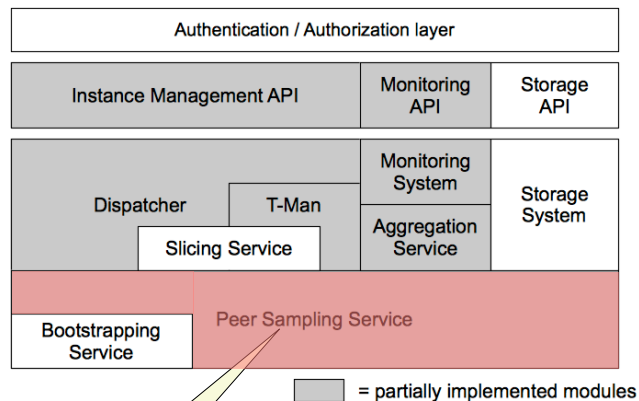
- The infrastructure we envision is similar to a classical p2p system and, as a basis for cloud computing, offers
  - Very low initial investment costs,
  - Distributed power consumption,
  - Distributed heat generation/dissipation,
  - Distributed network connectivity
- The challenge is to maintain a coherent abstraction over this *large-scale, distributed*, *unreliable* and *dynamic* infrastructure in a totally decentralized and self-organizing manner
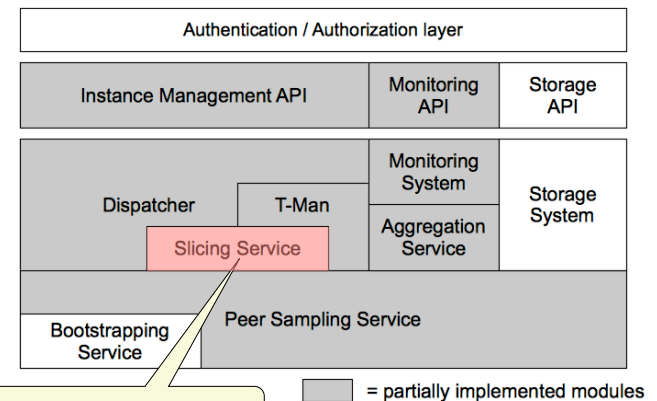

Authentication / Authorization layer

Instance Management API | Monitoring API | Storage API

Dispatcher | T-Man | Monitoring System | Storage System
Slicing Service | Aggregation Service

Bootstrapping Service | Peer Sampling Service

☐ = partially implemented modules

Authentication / Authorization layer

Instance Management API | Monitoring API | Storage API

Dispatcher | T-Man | Monitoring System | Storage System
Slicing Service | Aggregation Service

Bootstrapping Service | Peer Sampling Service

☐ = partially implemented modules

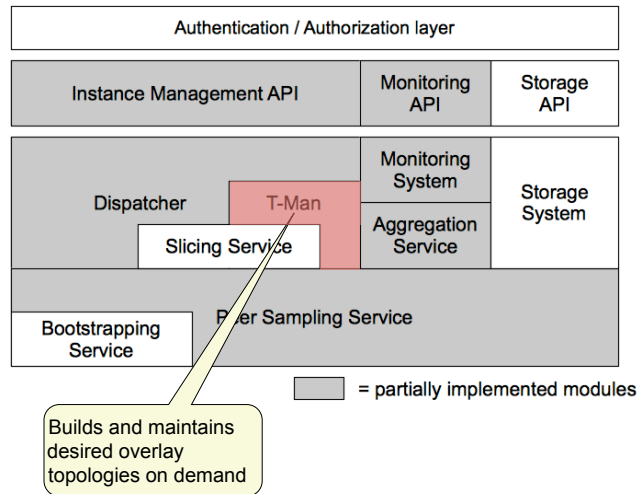Membership service. Guarantees connectivity among the participating nodes through an unstructured overlay network

---

Authentication / Authorization layer

Instance Management API | Monitoring API | Storage API

Dispatcher | T-Man | Monitoring System | Storage System
Slicing Service | Aggregation Service

Bootstrapping Service | Peer Sampling Service

☐ = partially implemented modules

Selects a subset of the nodes satisfying a set of attributes (e.g., top 5% of fastest nodes)

P2P Cloud - Architecture

Authentication / Authorization layer

Instance Management API | Monitoring API | Storage API

Dispatcher | T-Man | Monitoring System | Storage System
Slicing Service | | Aggregation Service |

Bootstrapping Service | Peer Sampling Service

☐ = partially implemented modules

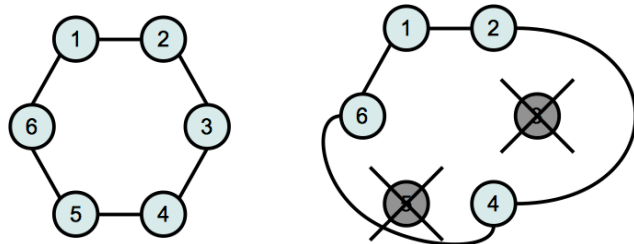Builds and maintains desired overlay topologies on demand

Sub-Clouds through Slicing

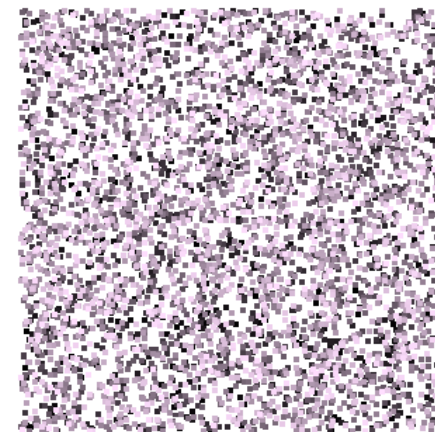- Slicing builds sub-clouds as disjoint ring overlays on top of the unstructured membership layer
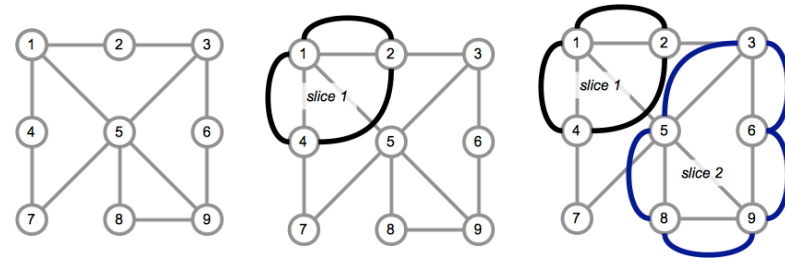
T-Man

- Sub-clouds are maintained in the presence of churn using the T-Man overlay protocol
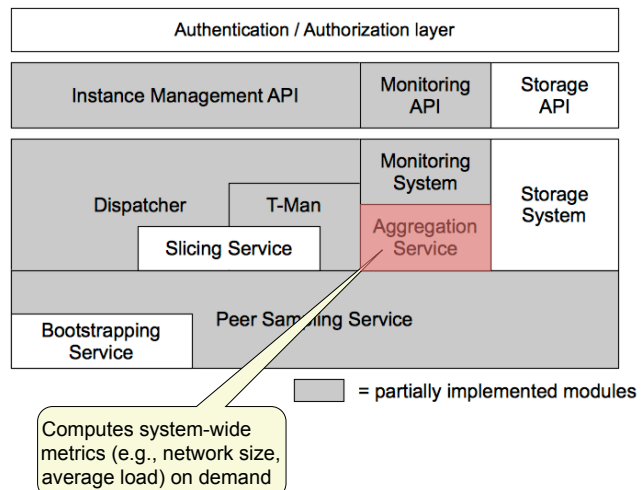
Building a ring with T-Man

P2P Cloud - Architecture

Authentication / Authorization layer		
Instance Management API	Monitoring API	Storage API

Dispatcher	T-Man	Monitoring System	Storage System
	Slicing Service	Aggregation Service	
Bootstrapping Service	Peer Sampling Service		

□ = partially implemented modules

Computes system-wide metrics (e.g., network size, average load) on demand

Status Report

- Prototype of major pieces as a middleware implemented in Java with JRMI

  http://cloudsystem.googlecode.com/

- Emulates the Amazon EC2 "Infrastructure-as-a-Service" API

- Machine virtualization, storage, authentication and authorization issues currently ignored

  O. Babaoglu, M. Marzolla, M. Tamburini. *Design and Implementation of a P2P Cloud System*. In *Proceedings of the 27th ACM Symposium on Applied Computing (SAC 2012)*, Trento, Italy, March 2012.
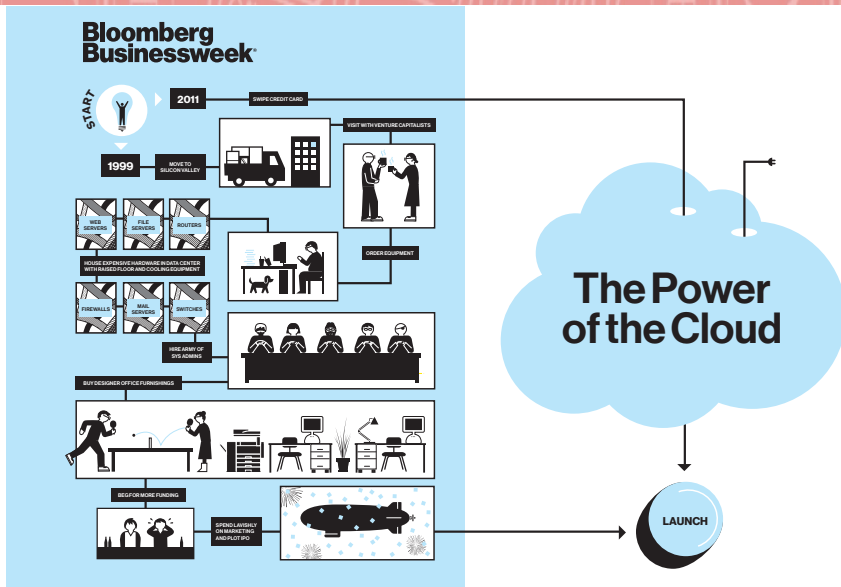
Towards accountability and trust

- By any measure, current commercial cloud offerings have to be judged as commercial successes
- Because cloud computing *empowers* startups

- Yet, of the 60,000 (estimated) customers of Amazon AWS, only a small fraction are startups
- A larger number of current cloud customers are governments, banks, pharmaceuticals companies and other large corporations that outsource only *small pieces* of their enterprise that deal with *less sensitive* data to the cloud
- What prevents corporations and government organizations from realizing the full potential of cloud computing?
- Lack of *accountability*, and as a consequence, lack of *trust*

Accountability and Trust

- Current service offerings are provided *as-is* and *as-available*
- Disclaimer clause of the Amazon AWS Customer Agreement:

*"WE AND OUR AFFILIATES AND LICENSORS **MAKE NO REPRESENTATIONS OR WARRANTIES OF ANY KIND**, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE REGARDING THE SERVICE OFFERINGS OR THE THIRD PARTY CONTENT, INCLUDING ANY WARRANTY **THAT** THE SERVICE OFFERINGS OR THIRD PARTY **CONTENT WILL BE UNINTERRUPTED, ERROR FREE OR FREE OF HARMFUL COMPONENTS**, OR THAT ANY CONTENT, INCLUDING **YOUR CONTENT** OR THE THIRD PARTY CONTENT, **WILL BE SECURE** OR **NOT** OTHERWISE **LOST OR DAMAGED**.*

*WE AND OUR AFFILIATES OR LICENSORS **WILL NOT BE LIABLE TO YOU FOR ANY** UNANTICIPATED OR **UNSCHEDULED DOWNTIME** OF ALL OR A PORTION **OF THE SERVICES** FOR ANY REASON, INCLUDING **AS A RESULT OF POWER OUTAGES**, SYSTEM **FAILURES OR** OTHER **INTERRUPTIONS**."*

A few Notable Cloud Outages of 2011

- "New Amazon Cloud Outage Takes Down Netflix, Foursquare"
- "Amazon EC2 Goes Dark In Morning Cloud Outage"
- "Google Docs Goes Dark In Evening Cloud Outage"
- "Lightning Causes Amazon, Microsoft Cloud Outages In Europe"
- "Another Cloud Outage Strikes Microsoft BPOS, Exchange Online"
- "Yahoo Mail Takes Big Hit In Cloud Outage"
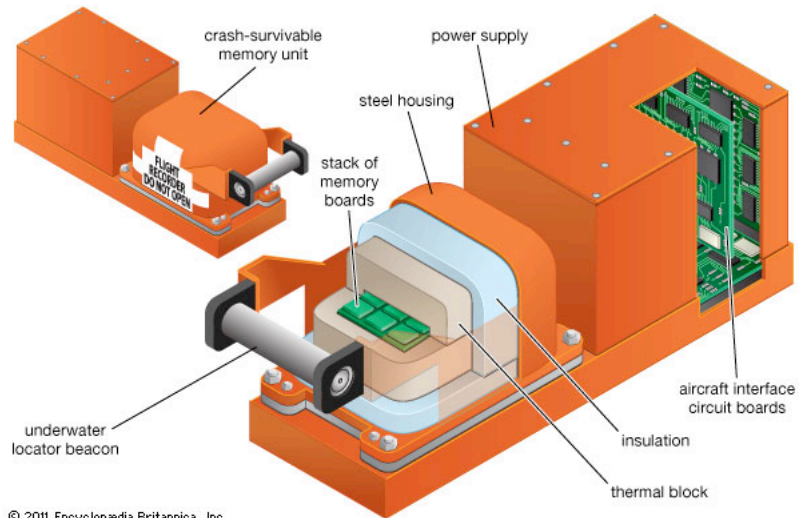- "GMail suffers massive outage affecting roughly 150,000 users"

Accountability and Trust

- Accountability in a cloud computing environment needs to address:
  - Who is responsible when data is lost, corrupted or disclosed?
  - Who is responsible when applications return no results, late results or erroneous results?
  - What are the legal implications of data and applications being held by third parties, possibly in multiple judicial domains?
  - How can disputes be settled impartially by third parties?

Civil Aviation

- Look to the **safety** aspects of Civil Aviation for inspiration
- Civil Aviation is a very complicated system of mutually suspecting agents set in a complex technological, economical, international, regulatory and legal context
- Yet it works surprisingly well and flying as a mode of transportation enjoys a high level of *trust* among its customers
- An important factor of this trust in flight safety rests with the requirement (by international law) that airlines render their flight operations *accountable*
- The famous "black box"

crash-survivable memory unit
power supply
steel housing
stack of memory boards
aircraft interface circuit boards
underwater locator beacon
insulation
thermal block

© 2011 Encyclopædia Britannica, Inc.

33

- Specification regulated by the *International Civil Aviation Organization* (ICAO)
  - Tamper proof
  - Withstand an acceleration of 3400 g's
  - Withstand extreme high and low temperatures
  - Withstand immersion to a depth of up to 6,000 meters
- Since the 1960's, mandatory for all commercial aircraft under internationally-agreed regulations
- Recorded data can be extracted and analyzed by the *Flight Data Analysis Service* of the *International Air Transport Association* (IATA)

© Babaoglu

34

- In the aftermath of an accident, the recovered *Flight Data Recorder* (together with the *Cockpit Voice Recorder*) are typically sufficient to attribute the cause of the accident to
  - Airline (pilot, cabin crew, ground personnel, maintenance, etc.)
  - Aircraft Manufacturer (design, manufacturing, materials, etc.)
  - Other parties (air traffic controller, another aircraft, etc.)
  - External Factors (weather, birds, volcanoes, etc.)
- Evidence typically stands in court and is the basis for legal settlements

© Babaoglu

35

- What we need is a "cloud flight recorder" (CFR)
- Integral part of a technical infrastructure along with a legal regulatory framework for making cloud computing *accountable*, and ultimately making cloud services *mutually trustworthy* for customers and providers

© Babaoglu

36

Actors

- Actors in a CFR-enabled cloud setting:
    - *P* - the cloud service provider
    - *U* - end user
    - *Q* - regulatory organization, equivalent to the ICAO
    - *R* - certified CFR provider
- The logs maintained by the CFR should be *self extracting* and *self describing* such that the equivalents of the *National Transportation Safety Board*, *Flight Data Analysis Service* and lawyers/judges are not needed

Technical Challenges

- How to extract from a service contract formal descriptions of
    - rules that state the rights, obligations and prohibitions of providers and customers,
    - specifications for a CFR as a list of events and their attributes that must be logged
- Requirements for a CFR logging facility:
    - *Fine-grained* to allow backtracking of "incidents"
    - *Tamper resistant*
    - *Trustworthy*
    - *Non-reputable*
    - *Non-intrusive*
    - *Closed* (does not rely on services outside the cloud itself)

Status

- For the time being, only a "paper design"
- Included in several funding proposals as an idea
- No results or publications