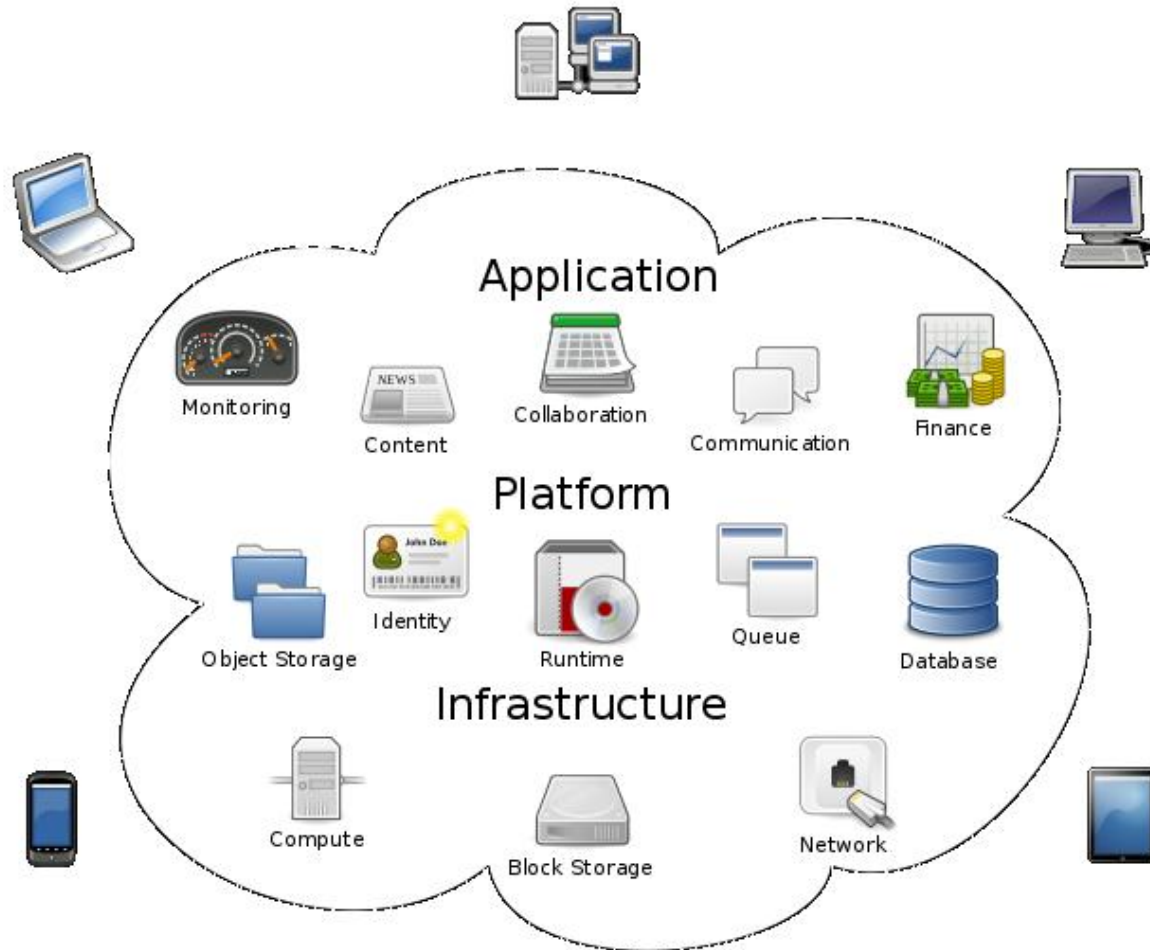


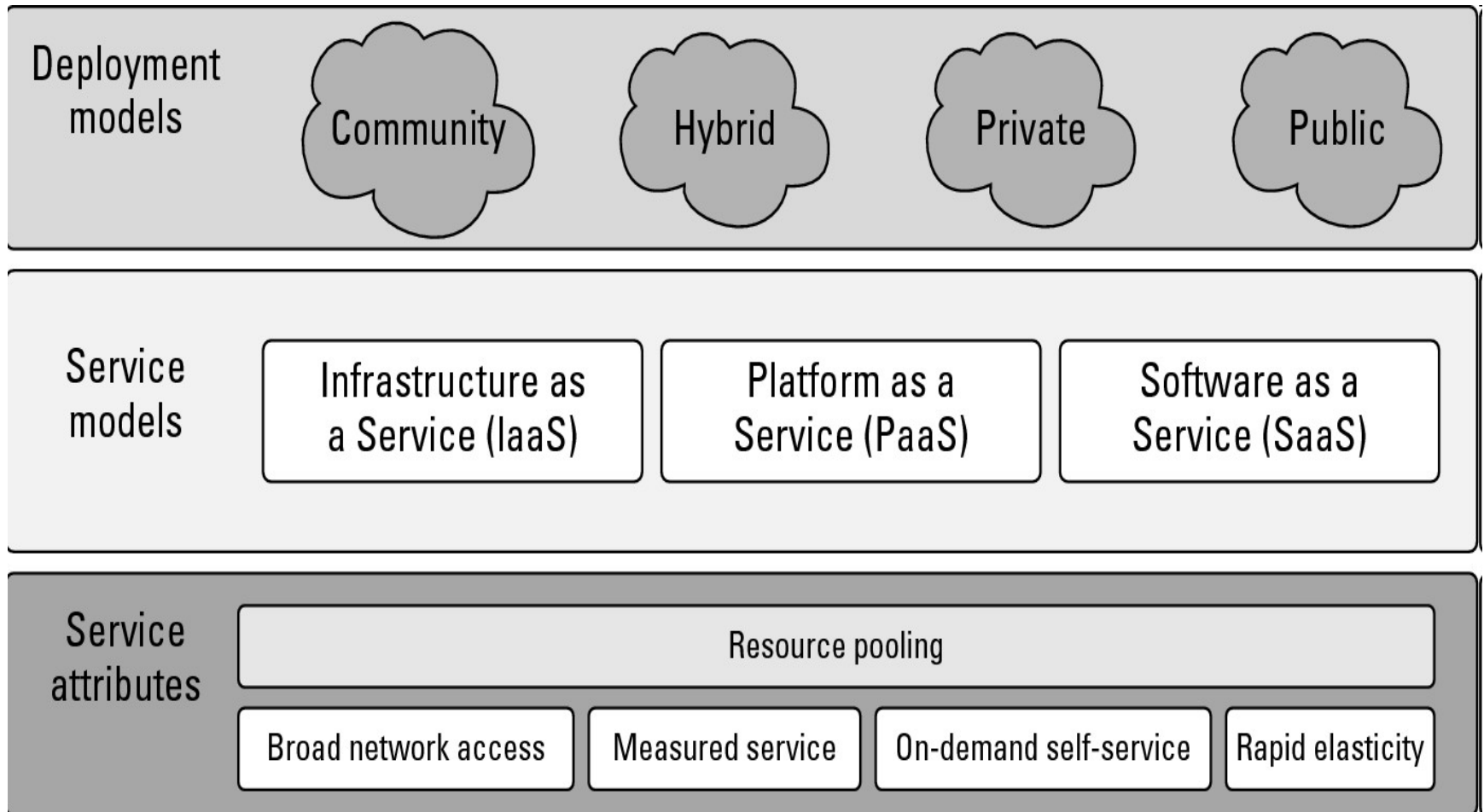
Računarski oblaci (cloud computing)



Pojam računarskog oblaka

- NIST definicija:
- model korišćenja računara koji omogućava (mrežni) pristup po potrebi deljenom skupu prilagodljivih računarskih resursa (mrežama, serverima, skladištima podataka, aplikacijama i servisima) koji se mogu brzo obezbediti i osloboditi uz minimum upravljačkog napora i interakcije sa provajderom.
- NIST = Američki nacionalni institut za standarde i tehnologiju

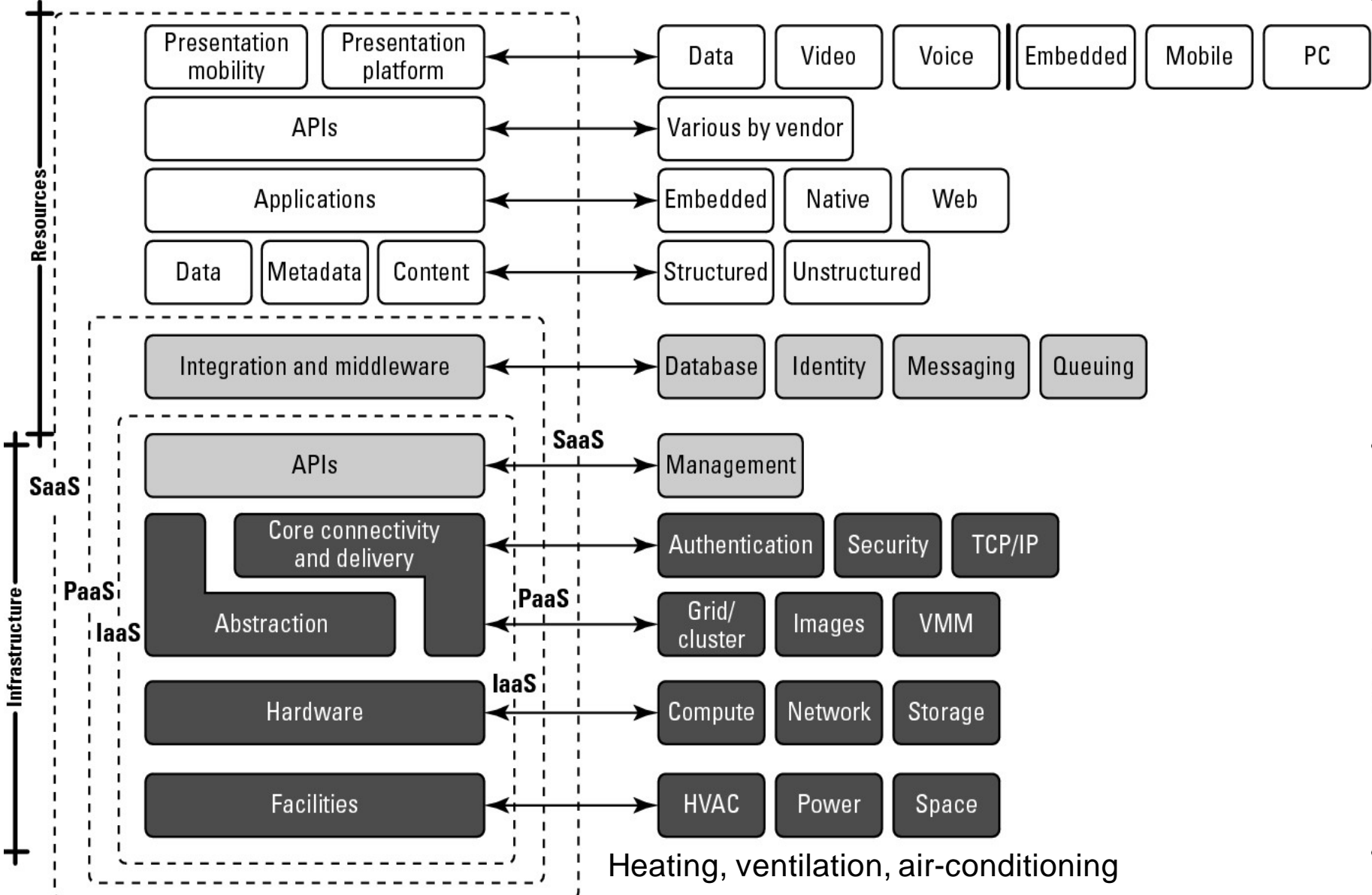
NIST kategorizacija oblaka



Modeli servisa

- Arhitektura oblaka je slojevita, od hardvera ka aplikacijama za krajnjeg korisnika.
- Na dnu steka je hardverska računarska i mrežna infrastruktura. Kako se ide na gore u steku, sledeći model servisa obuhvata i mogućnosti modela nižeg nivoa.
- Prema nivou na kome se pružaju usluge korisnicima razlikuju se sledeći tipovi oblaka:
 - infrastrukturni (IaaS – infrastructure as a service),
 - platformski (PaaS) i
 - aplikativno-softverski (SaaS)

Modeli servisa



IaaS model servisa

- Infrastrukturni servisi oblaka, poznati pod imenom "Infrastruktura kao servis" (IaaS), čine dostupnom računarsku infrastrukturu – tipično virtuelizovano okruženje, kao servis. Iz skupa servera, spoljne memorije i mrežnih komponentata uzimaju se resursi prema trenutnoj potrebi. Korisnik slobodno bira operativne sisteme i aplikacije koje će koristiti.

IaaS model servisa

- Primeri provajdera IaaS servisa:
 - Amazon Web Services Elastic Compute Node EC2
 - GoGrid
 - Rackspace
 - VMWare
 - (open source) Eucalyptus, OpenStack
- Na Amazon-u (EC2), kao tipičnom primeru IaaS provajdera, klijent dobija računar u vidu slike virtuelne mašine, virtuelnog diska, a on zatim instalira operativni sistem po želji i aplikacije na taj virtuelni sistem.

PaaS model servisa

- Platformski oblaci uključuju pored hardvera i sloj softvera - integrisan operativni sistem, softver srednjeg nivoa (tipa aplikativnih servera) i razvojne alate.
- Korisnik može sam da bira i razvija aplikativni softver i konfiguraciju operativnog okruženja.
- Primeri PaaS provajdera:
 - Windows Azure
 - Google AppEngine
 - GoGrid CloudCenter

SaaS model servisa

- Softver kao servis omogućava da se kompletna aplikacija ponudi u vidu servisa po potrebi. Jedna instanca softverske aplikacije izvršava se na oblaku i opslužuje veći broj korisnika i klijentskih organizacija (osobina **multitenancy**; tenant=stanar, zakupac).
- Multitenant (MT) aplikacija je suprotna aplikacijama koje imaju posebnu instancu (ili hw server) za svaku klijentsku organizaciju.
- Kod MT arhitekture, aplikacija se projektuje da virtuelno podeli podatke i konfiguraciju pa klijenti imaju privid da rade sa svojom instancom.

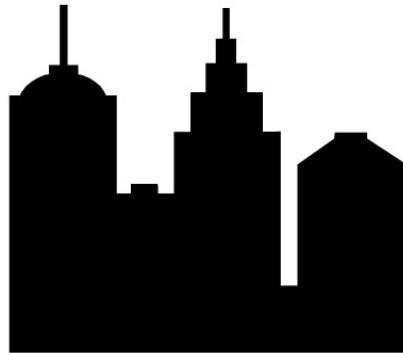
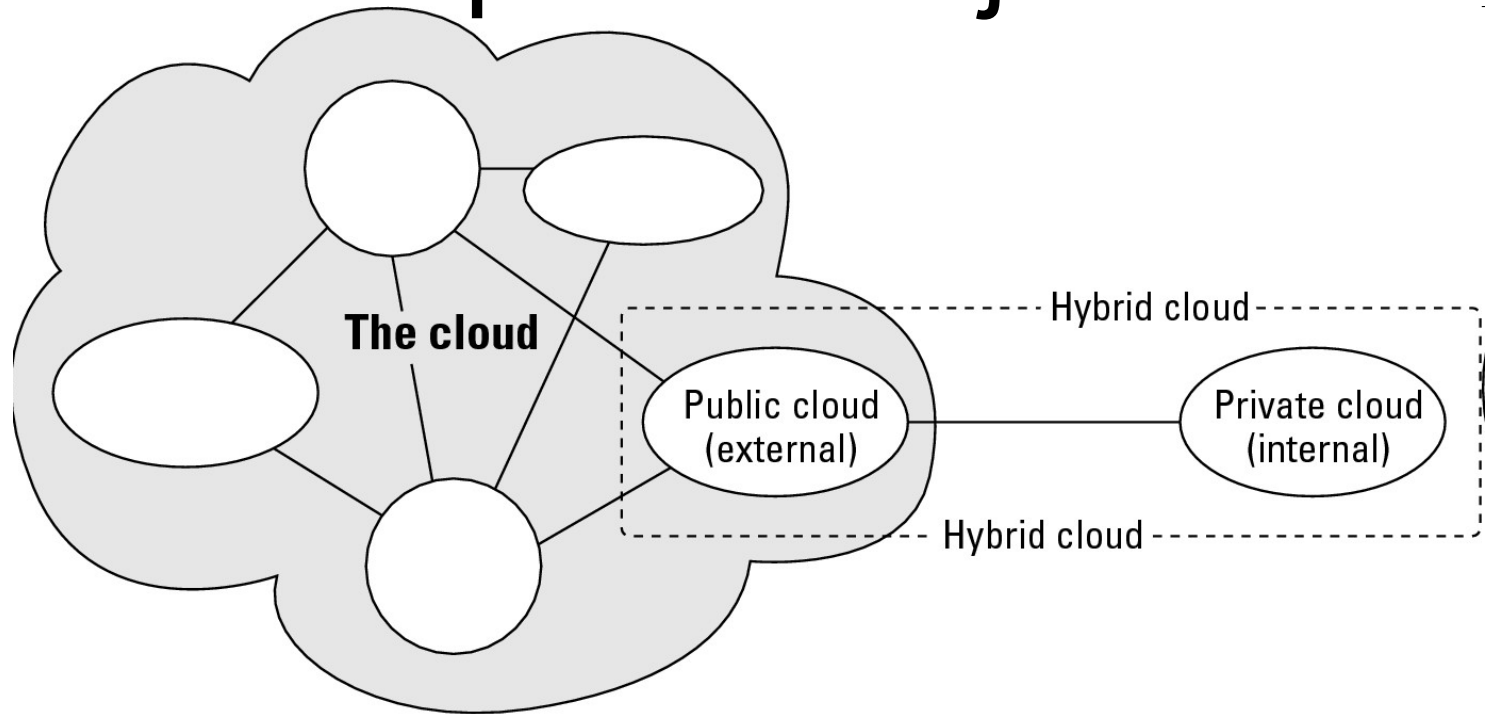
SaaS model servisa

- Primeri SaaS:
 - GoogleApps
 - Oracle on Demand
 - Salesforce.com
 - SQL Azure
- Kod SaaS, klijent koristi aplikaciju kada mu je potrebna, nije odgovoran za instalaciju niti održavanje a provajderu plaća prema obimu korišćenja

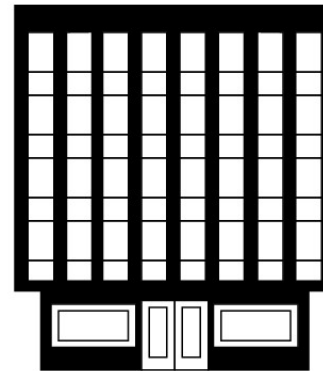
Modeli raspoređivanja oblaka

- **Javni oblak:** javna infrastruktura dostupna za javnu upotrebu ili alternativno velikoj industrijskoj grupi, u vlasništvu organizacije koja prodaje oblačne servise.
 - **Privatni oblak:** privatna infrastruktura koristi se isključivo za potrebe jedne organizacije. Oblakom se može upravljati od strane te organizacije ili trećeg lica. Privatni oblaci mogu fizički biti ili unutar firme ili van firme.
 - **Hibridni oblak:** hibridni oblak kombinuje više oblaka (privatnih, javnih, komunalnih) gde ti oblaci zadržavaju svoje identitete, ali su vezani zajedno kao celina. Hibridni oblak može ponuditi standardizovan ili specifičan pristup podacima i aplikacijama, kao i prenosivost aplikacija.
 - **Komunalni oblak:** komunalni oblak je onaj u kome je oblak bio organizovan da služi zajedničkoj svrsi ili funkciji. To može biti za jednu organizaciju ili za nekoliko organizacija, koje dele zajedničke probleme kao što su njihovi ciljevi, pravila poslovanja, bezbednost, potreba za poštovanjem propisa i tako dalje. Komunalnim oblakom mogu upravljati konstitutivne organizacije ili treće lice.

Modeli raspoređivanja



Off premises
(external)



On premises
(internal)

Osnovne karakteristike računarskog oblaka prema NIST

- **Samousluga na zahtev:** klijent može da obezbedi kompjuterske resurse bez potrebe za interakcijom sa osobljem provajdera oblaka.
- **Široki mrežni pristup:** Pristup resursima u oblaku je dostupan preko mreže korišćenjem standardnih metoda na način koji obezbeđuje platformski nezavisan pristup klijentima svih vrsta.
Ovo uključuje mešavinu različitih operativnih sistema, i debele i tanke klijente kao što su laptop računari, mobilni telefoni, tableti.
- **Resursno objedinjavanje:** provajder oblaka stvara resurse koji su objedinjeno koriste u sistemu koji podržava “multitenant” upotrebu, gde se fizički i virtualni resursi dinamički dodeljuju ili preraspodeljuju po potrebi. Klijenti generalno ne znaju i nemaju kontrolu nad tačnom lokacijom resursa (procesori, memorija, spoljna memorija, mreža).
- **Brza elastičnost:** Resursi se mogu brzo i elastično, prema potrebi dodeliti ili preraspodeliti automatski ili manuelno. Sa stanovišta klijenta, računarski oblaci treba da izgledaju neograničeni i resursi se mogu kupiti u bilo koje vreme i u bilo kojoj količini.
- **Izmerena usluga:** korišćenje sistemskih resursa oblaka se može meriti, kontrolisati, i prijavljivati kako kupcu tako i provajderu. Klijentu može da se naplati na osnovu poznatih metrika poput prostora za skladištenje, broja transakcija, mrežnog ulaza/izlaza ili propusnog opsega, procenta iskorišćenje procesorske snage i tako dalje.

Prednosti oblaka

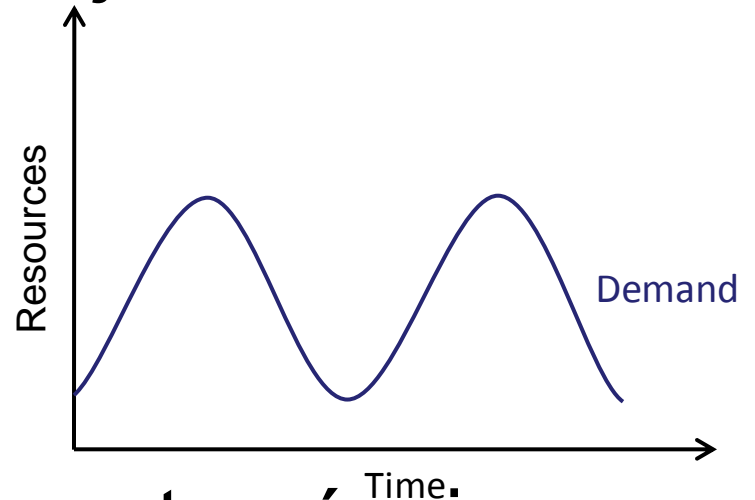
- Potencijali računarskih oblaka leže u efikasnijem korišćenju računarskih resursa nego u klasičnim pristupima, manji ukupan trošak posedovanja minimizovanjem investiranja u IT infrastrukturu, bez potrebe da se brine o servisnim zakrpama, upgrade-ima itd. Bitno se pojednostavljuje dostavljanje i primena IT servisa, što omogućava organizacijama da brže reaguju na poslovne zahteve. Kontrola nad obimom i kvalitetom usluge obezbeđuje se ugovorima (engl. Service Level Agreements). Korisnici imaju slobodu izbora servisa od bilo kod provajdera i pristup u svako doba i sa svakog mesta

Rizici oblaka

- Rizici prelaska na oblak (na postojećem nivou razvoja) ogledaju se u **tehničkim** (migriranje postojećih složenih inf. sistema nije automatizovan niti jednostavan proces), oblačna aplikacija ne mora imati onoliko funkcija koliko posebno razvijana niti toliko prilagođena konkretnom korisniku
- **pravni i administrativni** (npr. u nekim oblicima poslovanja postoje ograničenja gde podaci mogu da se čuvaju i kome da budu dostupni),
- **ekonomski** (ukoliko se neracionalno koriste, oblaci mogu biti skuplji od klasičnog posedovanja inf. sistema), i
- vezani za **sigurnost i dostupnost** servisa (SLA ugovori tipično štite provajdera da u slučaju prekida usluge plaća penale samo u vrednosti usluge, a ne u vrednosti poslovne štete koja je time izazvana)

Ekonomičnost i elasticitet oblaka

- Mnoge web aplikacije imaju ciklične krive opterećenja
 - Dnevno, nedeljno, mesečno, ...



- Postoje i česta impulsna opterećenja
 - Smrt Majkla Džeksona:
 - Google je pomislio da je meta napada, 30min google news je upućivao na stranu za prijavu greške: "Your query looks similar to automated requests from a computer virus or spyware application."

Data Centers

- Microsoft Dublin Datacenter (\$500M)
 - 27,300 m²
 - 22.2 Megawatt
- Chicago Data Center (\$500M)
 - 65,000 m²
 - 60 Megawatt
 - Containers with up to 2500 servers
 - 112 cont. 224.000 servers

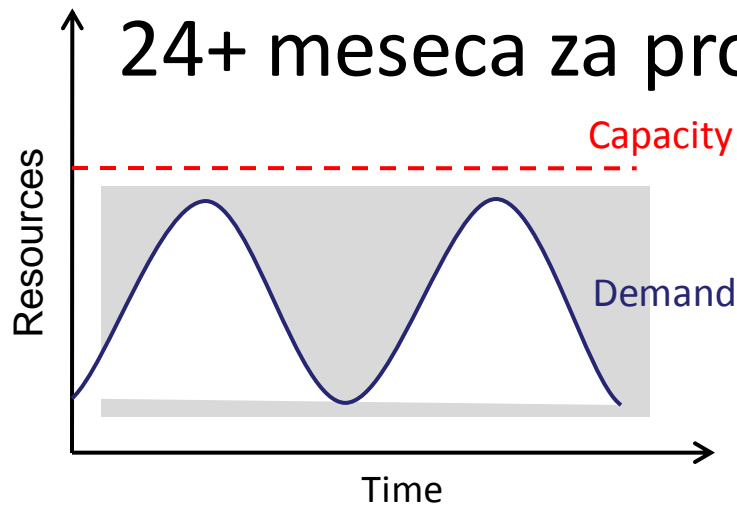


Microsoft's Chicago Data Center
October 2008
Photo by McShane Perry Studios, Chicago

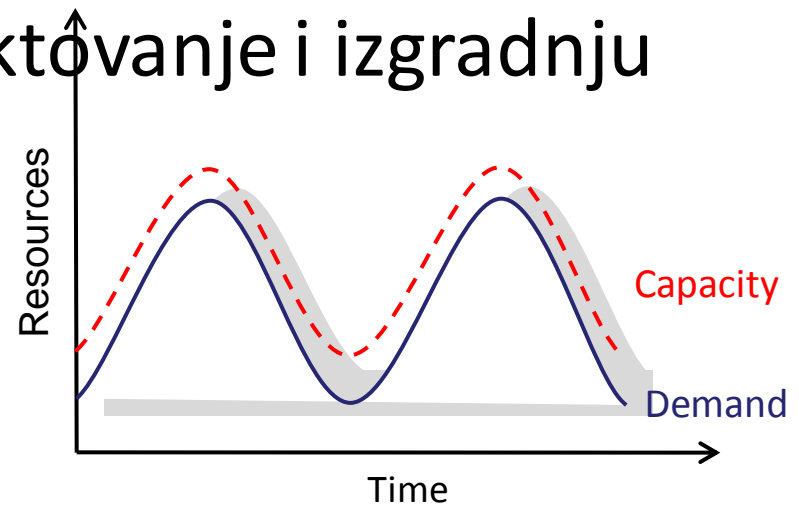


Ekonomičnost i elasticitet oblaka

- Plaćanje prema upotrebi umesto dimenzionisanja prema vršnom opterećenju
 - Veliki data centar košta $> \$150M$ and treba 24+ meseca za projektovanje i izgradnju



Static data center



Data center in the cloud

 neupotrebljeni resursi

Troškovi oblaka

- Amazon Elastic Compute Cloud (EC2)
- “Compute unit” rental: ~~\$0.10-0.80~~ 0.085-0.68/hour
 - 1 CU \approx 1.0-1.2 GHz 2007 AMD Opteron/Intel Xeon core

	Platform	Units	Memory	Disk
Small - \$0.10 \$0.085/hour	32-bit	1	1.7GB	160GB
Large - \$0.40 \$0.35/hour	64-bit	4	7.5GB	850GB – 2 spindles
X Large - \$0.80 \$0.68/hour	64-bit	8	15GB	1690GB – 4 spindles
High CPU Med - \$0.20 \$0.17	64-bit	5	1.7GB	350GB
High CPU Large - \$0.80 \$0.68	64-bit	20	7GB	1690GB
High Mem X Large - \$0.50	64-bit	6.5	17.1GB	1690GB
High Mem XXL - \$1.20	64-bit	13	34.2GB	1690GB
High Mem XXXL - \$2.40	64-bit	26	68.4GB	1690GB

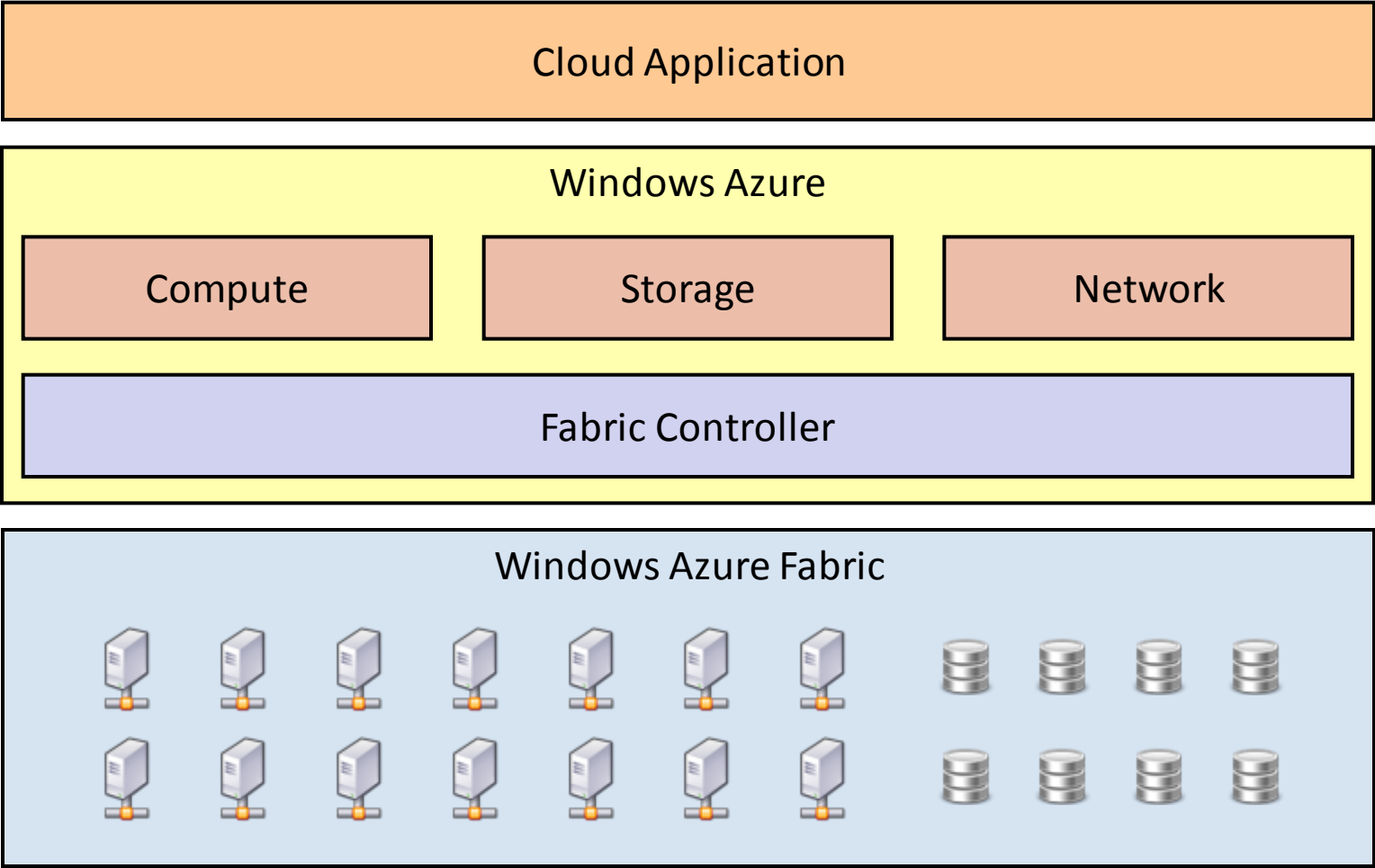
Northern VA cluster

- No up-front cost, no contract, no minimum
- Billing rounded to nearest hour (also regional, spot pricing)
- New paradigm(!) for deploying services?, HPC?

Programski model računarskog oblaka

- Definiše način povezivanja i korišćenja virtuelnih komponenata, uključujući smeštaj i instanciranje slika virtuelnih mašina i podataka sa skladišnog prostora u oblaku

Primer: arhitektura Windows Azure



Primer: arhitektura Windows Azure

- ***Fabric*** (tkanje?) mreža povezanih čvorova:
 - Servera
 - Brzih svičeva, rutera, balansera opterećenja
 - fiber-optičkih veza
- ***Azure Fabric Controller*** je servis koji prati, održava i dodeljuje mašine.
- **Osnovni (core) servisi Windows Azure-a**
 - ***Compute***: Middleware za skalabilne servise na Windows Serveru 2008.
 - ***Storage***: Upravljanje visoko dostupnim podacima (relacioni i ne relacioni).
 - ***Network***: Resursi za komuniciranje sa eksternim aplikacijama (Service Bus, servisna magistrala).

Glavni koncepti upravljanja oblakom

- API se tipično nudi u vidu RESTful web servisa, ali provajderi nude i različite biblioteke koje enkapsuliraju osnovni API
- **Representational State Transfer** (REST) je vrsta klijent-server arhitekture bez pamćenja stanja u kojoj je svaki servis identifikovan jedinstvenim URLom. URL se vraća korisniku na upit korišćenjem http protokola. Informacije o trenutnom stanju servisa i mogućim operacijama nad tim stanjem kodirane su u odgovoru nekim od Internet formata, kao što su html, xml ili JSON (java script object notation)

Primer

- Na primer list containers funkcija blob service REST API-ja u Azure Development Emulatoru:
- Upit (http GET):
- <http://127.0.0.1:10000/devstoreaccount1/guestbookpics/?comp=list>

Primer

- Odgovor servisa:
- ```
<EnumerationResults ContainerName="http://127.0.0.1:10000/devstoreaccount1/guestbookpics/">
<Blobs>
<Blob><Name>image_0B58FC53-D654-497D-B491-
9ECE24A1359B.jpg</Name><Url>http://127.0.0.1:10000/devstoreaccount1/guestbookpics/image_0
B58FC53-D654-497D-B491-9ECE24A1359B.jpg</Url><LastModified>Thu, 15 Mar 2012 11:37:29
GMT</LastModified><Etag>0x8CED0B723447460</Etag><Size>4931</Size><ContentType>image/
jpeg</ContentType><ContentEncoding/><ContentLanguage/></Blob>
<Blob><Name>image_55547A85-5439-460D-8B5F-
6F08006D9757.png</Name><Url>http://127.0.0.1:10000/devstoreaccount1/guestbookpics/image_55
547A85-5439-460D-8B5F-6F08006D9757.png</Url><LastModified>Thu, 15 Mar 2012 12:04:36
GMT</LastModified><Etag>0x8CED0BAECFB3840</Etag><Size>65854</Size><ContentType>ima
ge/png</ContentType><ContentEncoding/><ContentLanguage/></Blob>
<Blob><Name>thumbnail_image_0B58FC53-D654-497D-B491-
9ECE24A1359B.jpg</Name><Url>http://127.0.0.1:10000/devstoreaccount1/guestbookpics/thumbnail
_image_0B58FC53-D654-497D-B491-9ECE24A1359B.jpg</Url><LastModified>Thu, 15 Mar 2012
12:03:15
GMT</LastModified><Etag>0x8CED0BABC8A66E0</Etag><Size>12099</Size><ContentType>ima
ge/jpeg</ContentType><ContentEncoding/><ContentLanguage/></Blob>
<Blob><Name>thumbnail_image_55547A85-5439-460D-8B5F-
6F08006D9757.png</Name><Url>http://127.0.0.1:10000/devstoreaccount1/guestbookpics/thumbnail
_image_55547A85-5439-460D-8B5F-6F08006D9757.png</Url><LastModified>Thu, 15 Mar 2012
12:04:56
GMT</LastModified><Etag>0x8CED0BAF8A7F570</Etag><Size>0</Size><ContentType>image/pn
g</ContentType><ContentEncoding/><ContentLanguage/></Blob>
</Blobs>
<NextMarker/>
</EnumerationResults>
```

# Programiranje oblaka

- Provajderi infrastrukture oblaka nude različite programske biblioteke za pristup oblaku koje enkapsuliraju osnovni http protokol. Npr. Amazon EC2 nudi biblioteke za Javu, PHP, Python, Ruby i .NET programske jezike. Pored toga često postoje alati komandne linije i web interfejsi za upravljanje oblakom.

# Programiranje oblaka

- Važna karakteristika oblaka je da nudi automatsko balansiranje opterećenja i skaliranje korisnikovih servisa, često realizovano kroz merenje opterećenja i prema potrebi instanciranje novih primeraka servisa. To obično uvodi ograničenja u realizaciji poslovnih servisa, npr. da se ne pamti stanje klijenata koji se konektuju na servis u instanci, nego isključivo na skladišnom delu. Instance tipično međusobno mogu da komuniciraju i nekim protokolom slanja poruka osim web servisa.

# Servisi sa stanjem (Stateful)

- Host za servis obezbeđuje instancu servisa za svakog klijenta u toku celog trajanja sesije.



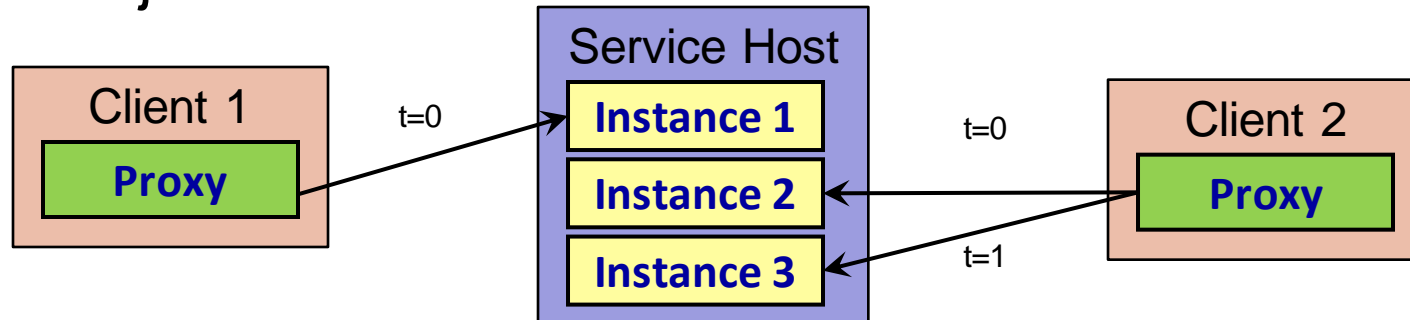
- Tehnologije
  - EJB: Stateful Session Beans
  - Windows communication foundation WCF: Per-Session Services, ASP.NET WebForms
  - CORBA
  - Java RMI

# Problemi sa stateful servisima

- Svaka instanca servisa zauzima serverske resurse
- Samo ograničen broj klijenata može biti opslužen.
- Šta se dešava sa instancom servisa klijent neočekivano “umre”?
  - (distribuirano sakupljanje smeća, istek vremena sesije...)
- Šta je stanje sesija, kada ima prekida veze?
  - Ponovno uspostavljanje veze može biti složeno.
- Transakcije bi trebalo da budu vezane za servisne operacije, a ne za sesije.
  - Kada transakcija uradi rollback, stanje baze i stanje sesije nisu u sinhronizmu.

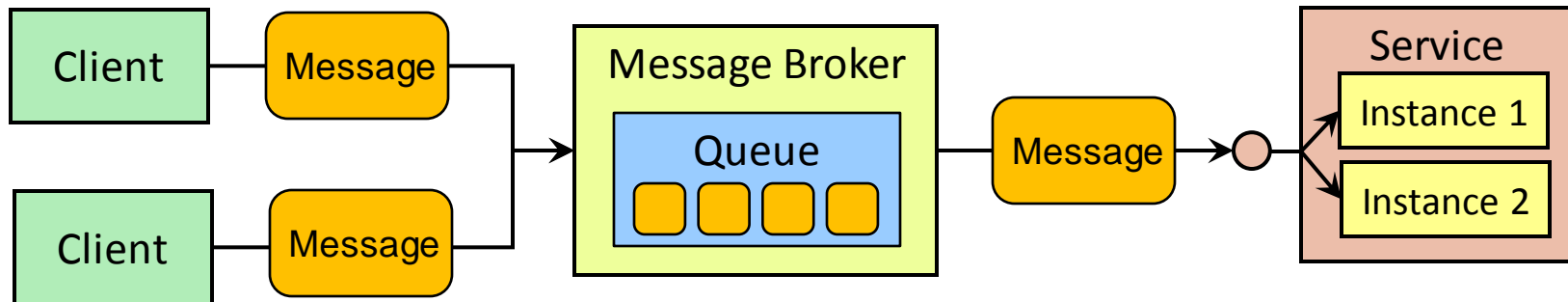
# Servisi bez stanja (stateless)

- Instanca servisa kreira se za svaki poziv servisne operacije.



- Nema stalne konekcije između klijenta i servisa
  - Resursi se mogu deliti među klijentima →
  - Poboljšana mogućnost skaliranja.
- Tehnologije
  - EJB: Stateless Session Beans
  - ASP.NET-MVC
  - Bazični PHP
  - HTTP, REST (Representational State Transfer)
  - WCF: Per-Call Services

# Asinhrono procesiranje poruka



- Poruke se čuvaju u perzistentnom redu i procesiraju se asinhrono (od zahteva klijenta).
- Prednosti
  - Klijenti nisu blokirani dok se vrši obrada.
  - Klijent i servis ne moraju istovremeno da budu aktivni.
  - Distribucija opterećenja na serveru (load balancing).
- Programski model je dosta različit od sinhronog
  - Klijent ne dobija rezultat niti povratnu informaciju u slučaju greške.

# Primer korišćenja Azure Service Management APIja za promenu broja instanci aplikacione uloge

- Koristi se PHP varijanta APIja:
- Inicijalizacija upravljačkog objekta:

```
require_once('Microsoft/WindowsAzure/Management/Client.php');
```

```
$client = new Microsoft_WindowsAzure_Management_Client(
 SUB_ID,
 CERT,
 CERT_KEY
);
```



# Primer korišćenja Azure Service Management APIja za promenu broja instanci aplikacione uloge

- Promena broja instanci određene uloge:

\$client->

```
setInstanceCountBySlot(AZURE_SERVICE,
'production', 'WebRole', <NUMBER OF NEW
ROLE COUNT>);
```

# Primer korišćenja Azure Service Management APIja za promenu broja instanci aplikacione uloge

- Čekanje na kraj operacije:

```
$status = $client->getOperationStatus($lastRequestId);
```

```
while ($status->Status == 'InProgress') {
 echo '.';
 $status = $client->getOperationStatus($lastRequestId);
 sleep(1);
}
```

# Skladištenje podataka u oblacima

- Virtuelne aplikacije normalno nemaju perzistentno skladište podataka, odnosno podaci se gube kada se izvršavanje instance okonča.
- Međutim, oblaci tipično nude više varijanti skladištenja, razmotrićemo to na primeru **Amazon Web Servisa**:
- **Simple Storage Service (S3)** je web servis kojem se pristupa RESTful http protokolom koji nudi smeštaj objekata od maksimum 5 gigabajta u kolekcije (buckets) i direktan pristup objektima (mogu dobiti i privremenu web adresu). S3 se odlikuje visokom pouzdanošću i skalabilnošću.
- Drugi smeštajni servis je **Elastic Block Storage** veličine do 1TB. Alocirani delovi se mogu asocirati sa jednom instancom virtuelne mašine u jednom trenutku i ponašaju se kao perzistentni diskovi. Na njima se mogu smeštati i klasične relacione baze (u tom slučaju, naravno, skalabilnost je ograničena).
- **SimpleDB** je skalabilno i pouzdano rešenje baze podataka nerelacionog tipa (leksikografsko indeksiranje) i sa odloženom konzistencijom (podaci se prosleđuju na više računara što ima konačno vremensko trajanje). Sličan je i **DynamoDB** (NoSQL baza).
- **Amazon RDS** je servis klasične relacione baze sa određenom mogućnošću skaliranja,

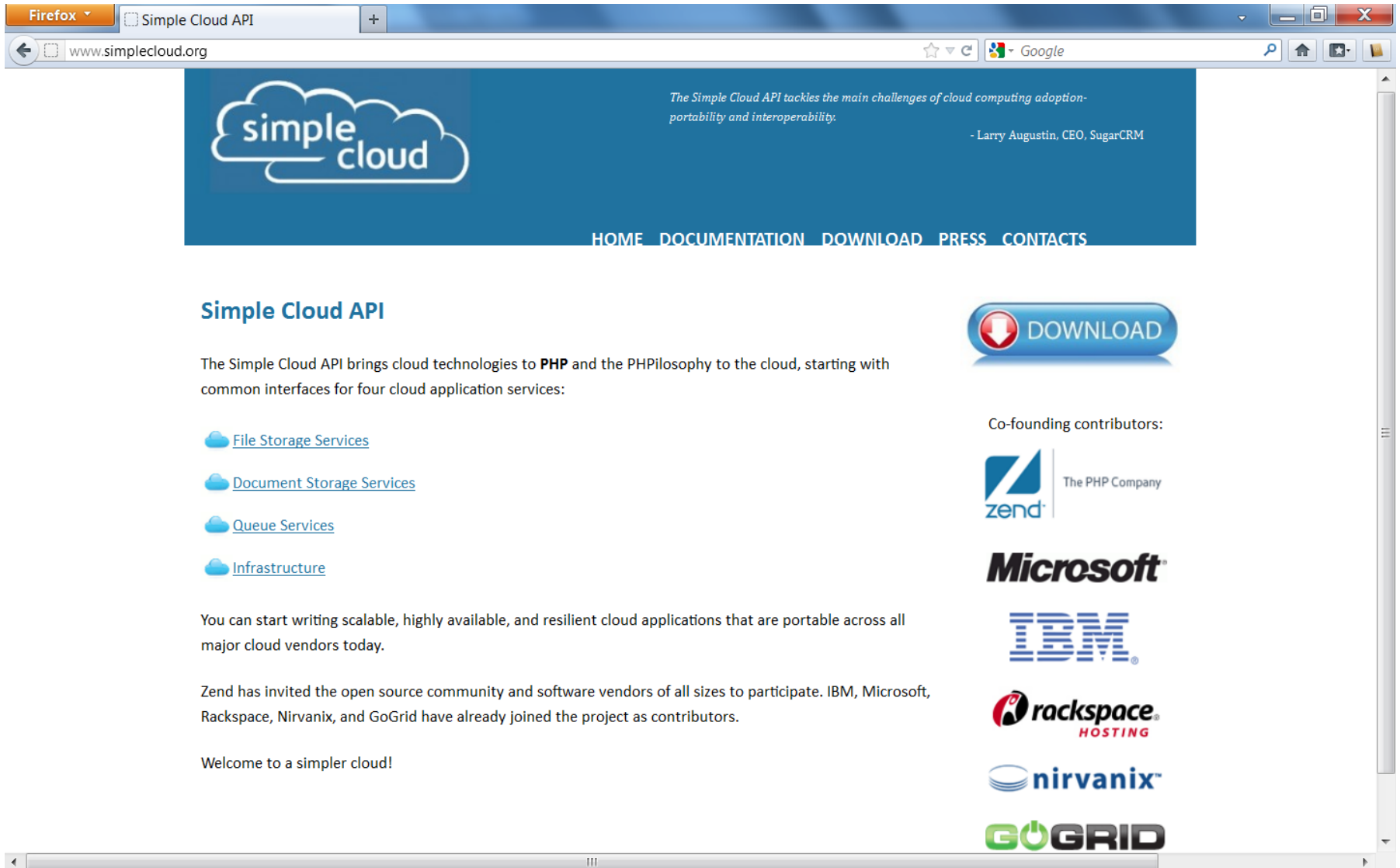
# Standardizacija API-ja

- Radovi u toku – veći broj tela za standardizaciju:
  - Distributed Management Task Force
  - Cloud Security Alliance
  - ETSI TC Cloud (Evropa)
  - National Institute of Standards and Technology

# Povezivanje na oblak (klijenti)

- Web browseri (html, ajax, flash, ...)
- Cloud operativni sistemi (Chromium OS, Jolicloud OS,...)
- Posebne aplikacije (tipično da obezbede sigurnu vezu, VPN itd) npr. gbridge za googletalk

# Standardizacija APIja



The screenshot shows a Firefox browser window with the address bar displaying "www.simplecloud.org". The page features a blue header with the "simple cloud" logo and a quote from Larry Augustin, CEO of SugarCRM. Below the header is a navigation menu with links for HOME, DOCUMENTATION, DOWNLOAD, PRESS, and CONTACTS. The main content area includes a "Simple Cloud API" section with a "DOWNLOAD" button, a list of services (File Storage, Document Storage, Queue, Infrastructure), and a list of co-founding contributors (Zend, Microsoft, IBM, Rackspace, Nirvanix, GoGrid).

Firefox Simple Cloud API +

www.simplecloud.org Google

**simple cloud**

*The Simple Cloud API tackles the main challenges of cloud computing adoption - portability and interoperability.*

- Larry Augustin, CEO, SugarCRM

HOME DOCUMENTATION DOWNLOAD PRESS CONTACTS

## Simple Cloud API

The Simple Cloud API brings cloud technologies to **PHP** and the PHPilosophy to the cloud, starting with common interfaces for four cloud application services:

- [File Storage Services](#)
- [Document Storage Services](#)
- [Queue Services](#)
- [Infrastructure](#)

You can start writing scalable, highly available, and resilient cloud applications that are portable across all major cloud vendors today.

Zend has invited the open source community and software vendors of all sizes to participate. IBM, Microsoft, Rackspace, Nirvanix, and GoGrid have already joined the project as contributors.

Welcome to a simpler cloud!

**DOWNLOAD**

Co-founding contributors:

- zend | The PHP Company
- Microsoft
- IBM
- rackspace HOSTING
- nirvanix
- GOGRID

# Trenutno stanje korišćenja tehnologije oblaka

- Istraživanje trž. u Vel. Britaniji – anketa u 250 preduzeća, mart 2010, [www.gfi.com](http://www.gfi.com)
- 86% srednjih (100-250 zaposlenih) kompanija koristi IT servise hostovane van firme, 69% malih (10-100), 39% najmanjih (do 10).
- 62% rukovodilaca koji donose IT odluke nije čulo za pojam računarskog oblaka!
- Za nekorišćenje tehnologije troškovi se pominju kao glavni razlog (44%), a sigurnost znatno manje (12%)
- Većina se odlučuje za hibridne modele raspoređivanja servisa