

APLICATII DISTRIBUITE IN CLOUD

Curs introductiv

Scopul cursului



- Introducere in ce este Cloud
- Arhitectura aplicatiilor distribuite in Cloud
- Notiuni de deployment
- Crearea unei aplicatii scalabila la nivelul internetului

Nota

- 70% proiect de echipa, 30% examen.
- Proiect in grupe de maxim 3.
- Examen practic, la calculator.
- Cursurile sunt comprimate pentru a lasa loc de lucru la laborator.

Terminologie

- Performanta
- Scalabilitate
- Scalare orizontala/verticala
- Date structurate/nestructurate
- BigData

Performanta



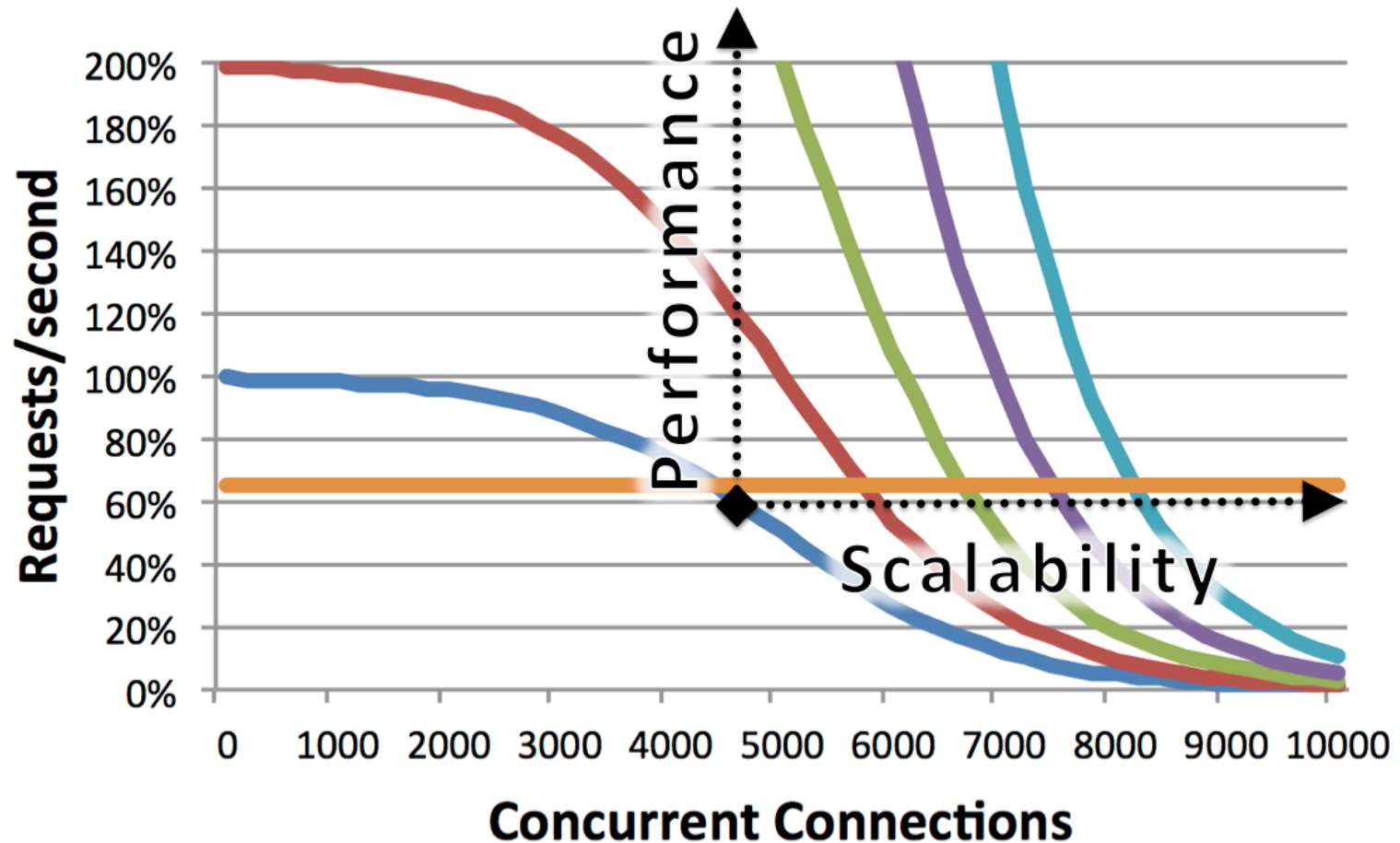
Sistem mai **performant**: capacitate ridicata de a procesa o singura operatie in timp mai scurt.

Scalabilitate



Sistem **scalabil**: capacitatea de a procesa foarte multe operatii in paralel fara ca performanta sa scada drastic

Scalabilitate si Performanta



Scalabilitate si performanta



Scalabilitate \neq Performanta

Sunt ortogonale!

Scalabilitate orizontala/verticala

Pentru a scala un sistem pe

- ▣ verticala: imbunatatim hardware-ul din care sunt compuse masinile sistemului. (eg. pui un procesor mai mare, hdd mai rapid)
- ▣ orizontala: adaugam masini identice cu cele deja in sistem. (eg. adaugam inca 2 masini care sa proceseze cereri web)

Date structurate/nestructurate

- Structurate: Date tabelare a caror forma este descrisa de o schema. Indexate. Interogabile.
 - ▣ Datele stocate intr-o baza de date mysql.
- Nestructurate: Restul.
 - ▣ Pozele din vacanta.
 - ▣ Loguri ale aplicatiei
- Obs:
 - ▣ Satisfac nevoi diferite, au avantaje si dezavantaje.
 - ▣ Alegerea lor depinde de natura problemei

BigData

Zeci de TB!

Exemple:

- Cartile scrise de omenire de-a lungul anilor.
- Conversatiile telefonice ale pamantului in cursul unei zile.

Cloud

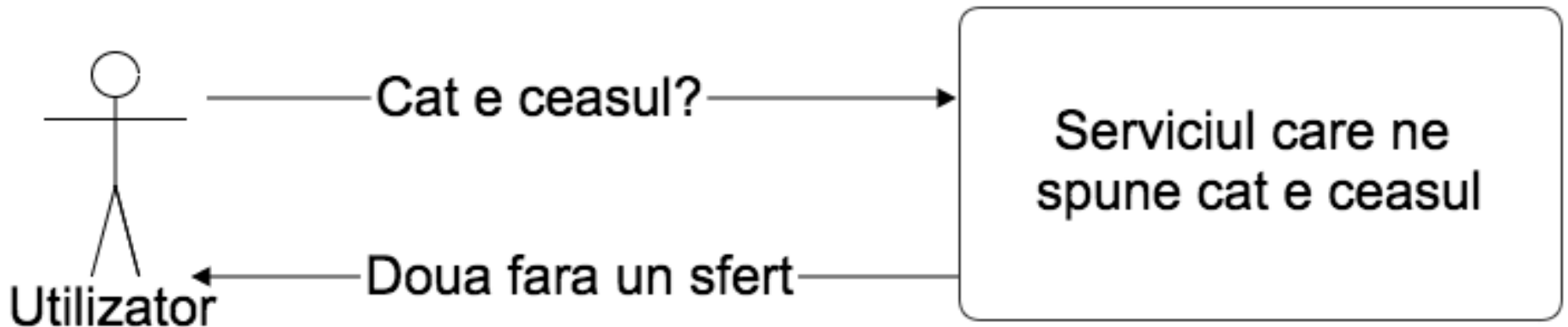
- Infrastructure inchiriată + servicii peste
- Exemple:
 - ▣ Amazon Web Services (AWS)
 - ▣ Windows Azure
 - ▣ Heroku
 - ▣ OpenStack (doar servicii cloud. open source)
- Vom studia AWS

Cum arata o aplicatie internet

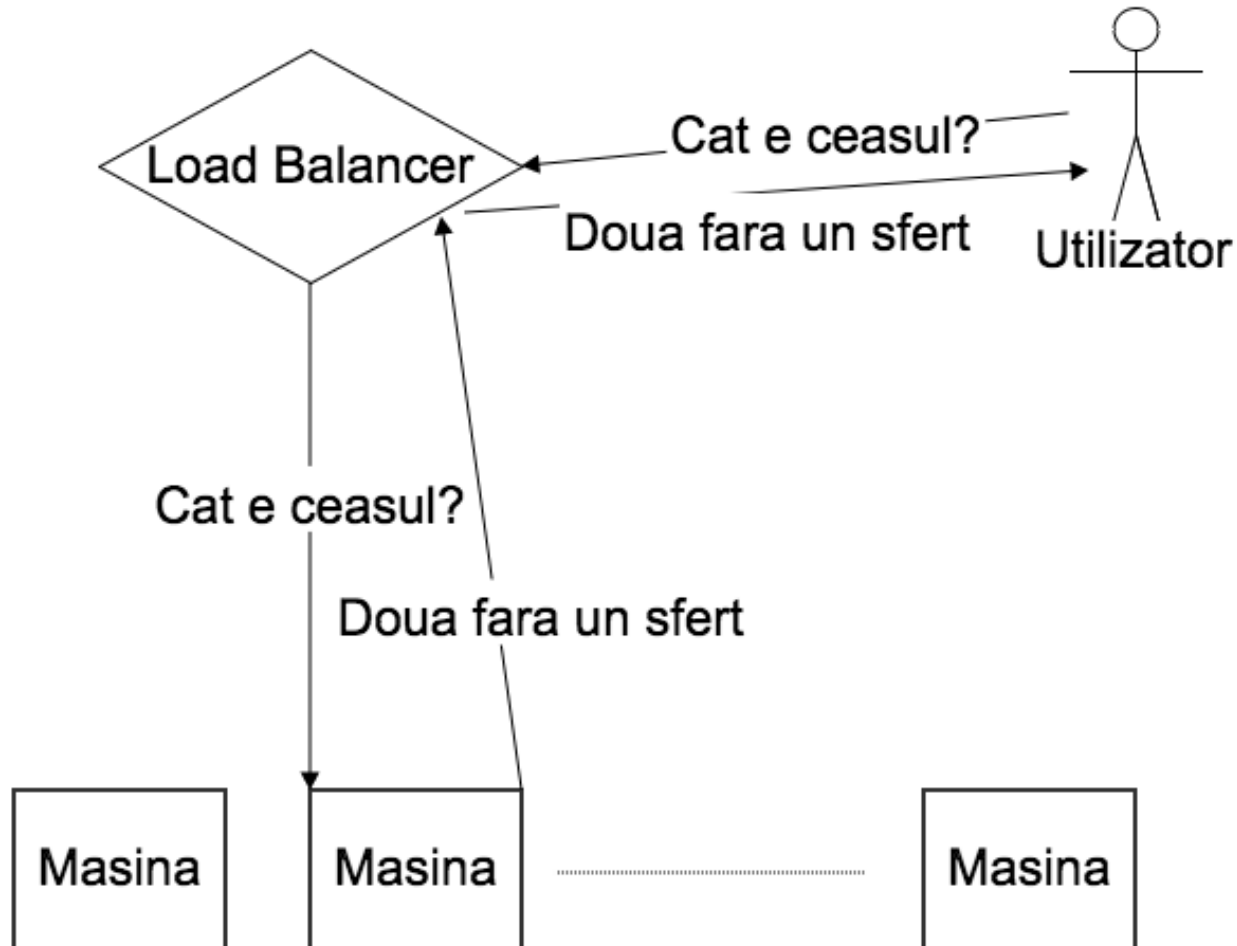
- Oricat de mult hardware ai pune pe masina, nu poate servi 1.000.000 de cereri pe secunda
- Solutie: Scalare pe orizontala, mai multe masini care raspund la fel pentru aceeaasi cerere.

Exemplul 1

Un serviciu care iti spune cat este ceasul si comunica prin http.

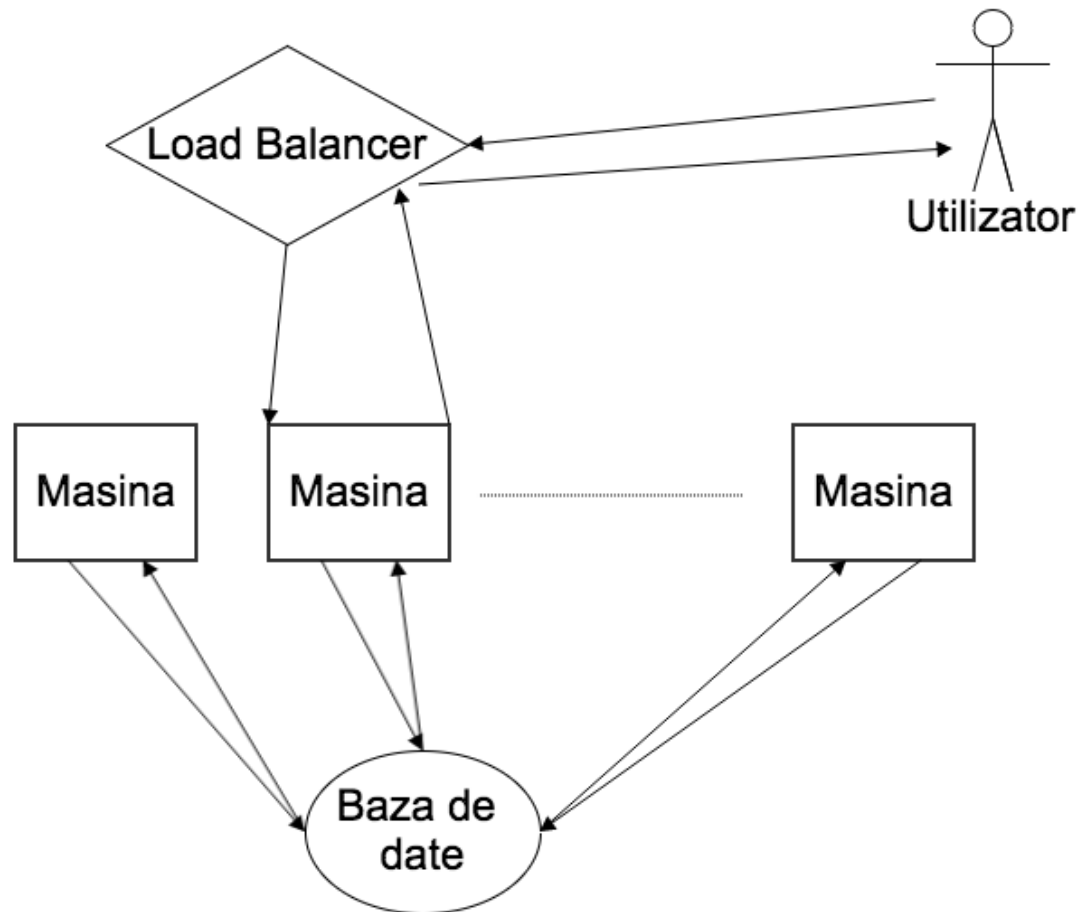


Arhitectura serviciului



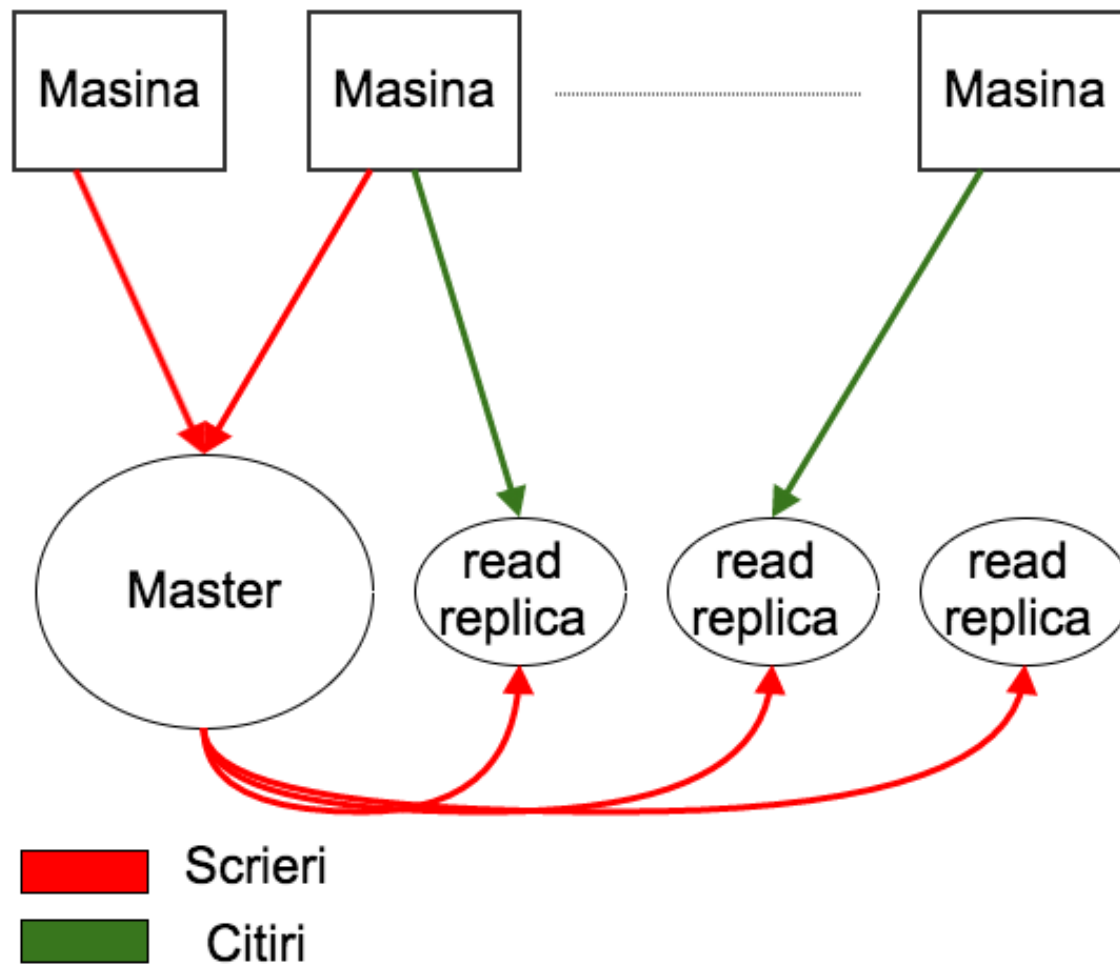
Exemplul 2

O aplicatie web care foloseste o baza de date.



Exemplul 2

Scalarea bazei
de date



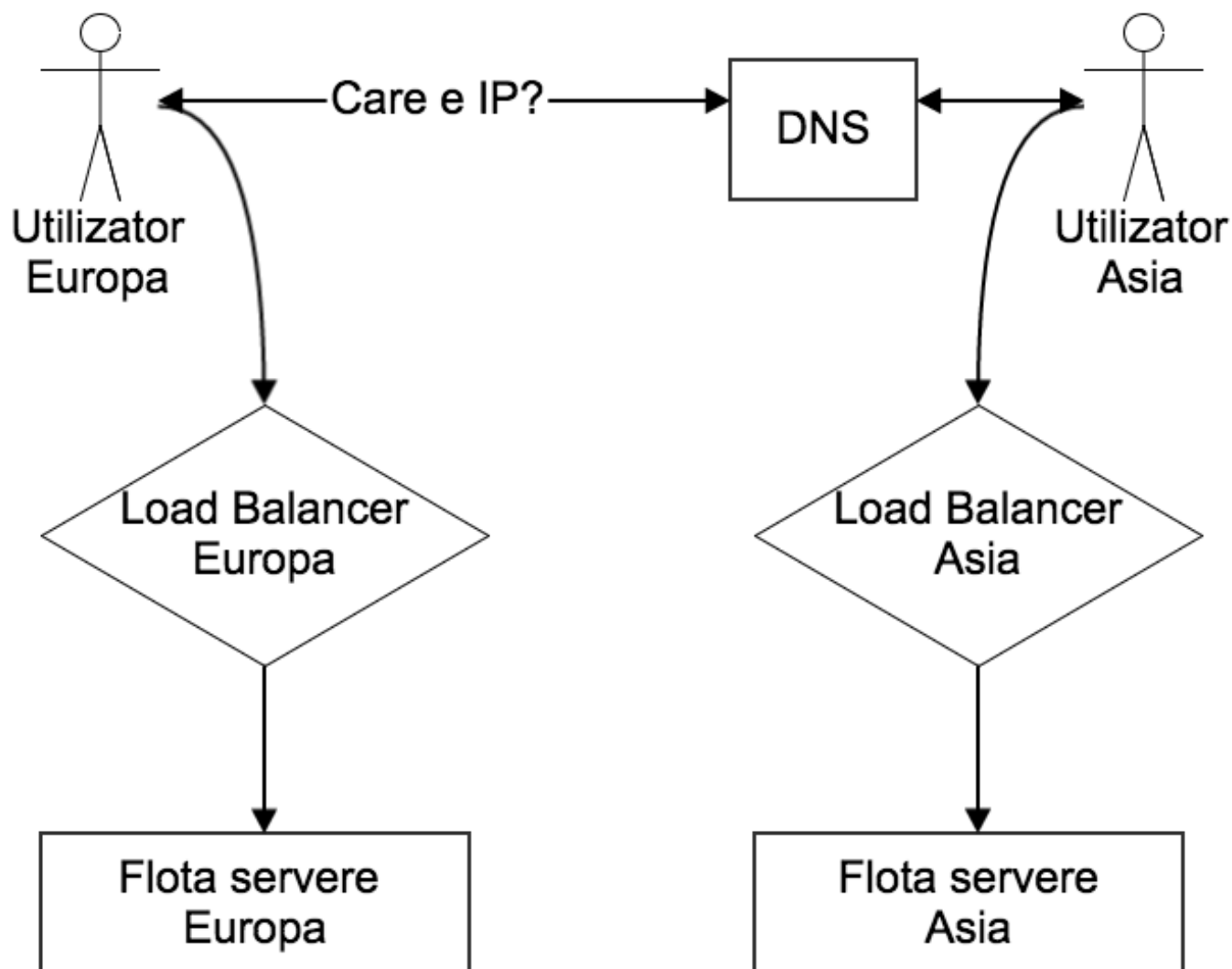
Exemplul 3



Aplicatie cu latenta redusa pe mai multe regiuni geografice:

- Fiecare regiune geografica are cate o instanta a aplicatiei dedicata.
- Rutarea se face la nivel de DNS.

Exemplul 3



Servicii AWS (ce vom folosi)

- ❑ EC2 – Elastic Compute 2
- ❑ ELB – Elastic Load balancer
- ❑ S3 – Simple Storage Service
- ❑ SQS – Message queue service
- ❑ RDS – Managed Relational Database Service
- ❑ DynamoDB – Predictable and scalable NoSQL store
- ❑ Route53 – DNS service
- ❑ Lista completa: <http://aws.amazon.com/products/>

Discutie



- Arhitectura unui sistem de conversie video.
- Arhitectura unui sistem de file sharing.

Alte caracteristici importante

- Viteza cu care sistemul de autoacomodeaza in functie de trafic.
- Disponibilitatea sistemului.
- Securitate.

Intrebari

