

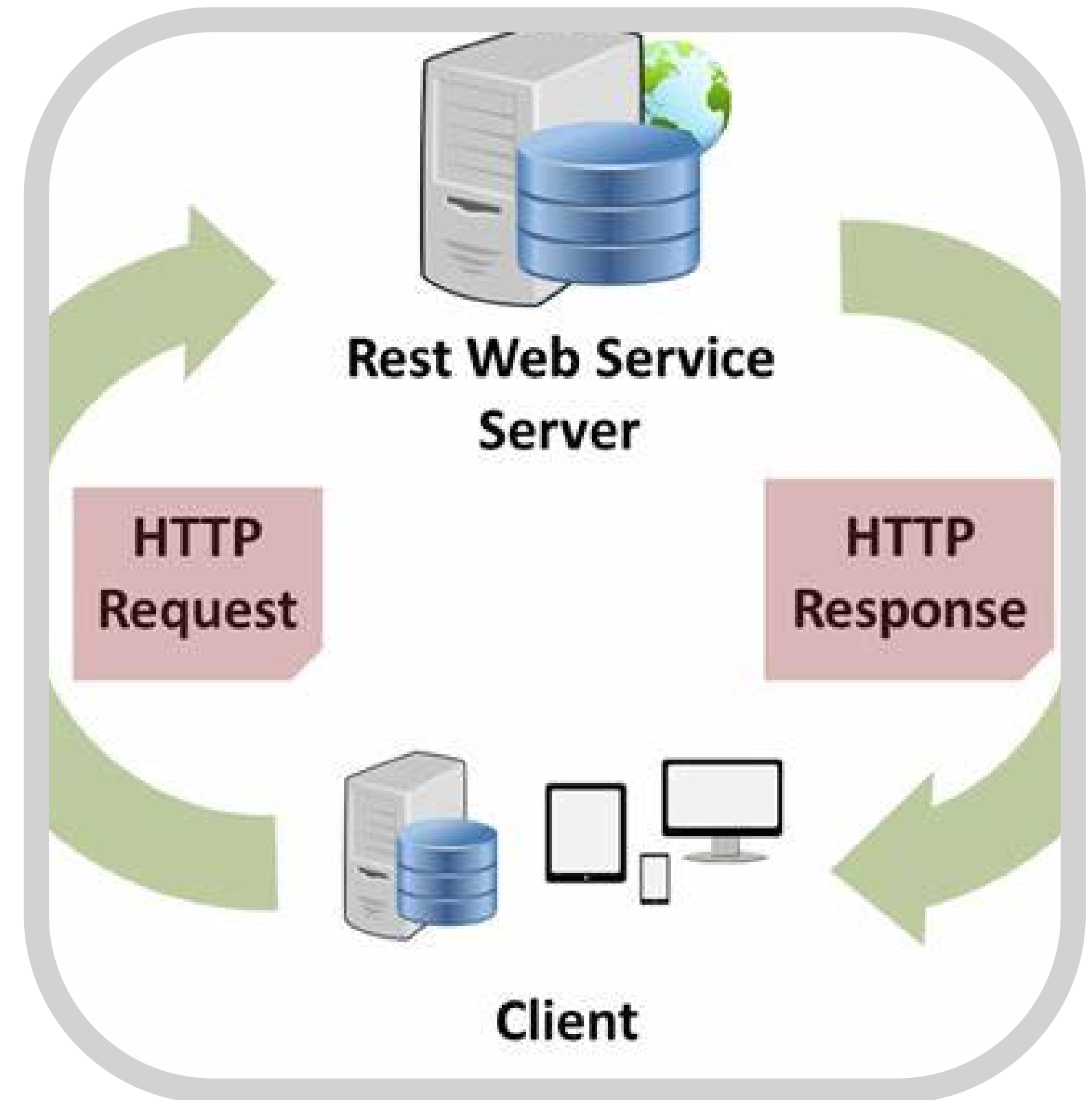


PREZENTARE GENERALĂ A ARHITECTURII REST

Realizat de Deac Laura
Grupa 244

CE ESTE REST?

REST este un stil arhitectural software care definește setul de reguli care trebuie utilizate pentru crearea serviciilor web. Serviciile web care urmează stilul arhitectural REST sunt cunoscute sub numele de servicii web RESTful. Acesta permite sistemelor de solicitare să acceseze și să manipuleze resursele web folosind un set uniform și predefinit de reguli. Interacțiunea în sistemele bazate pe REST are loc prin Protocolul de Transfer al Hipertextului (HTTP) de pe Internet.



PROTOCOLUL HTTP

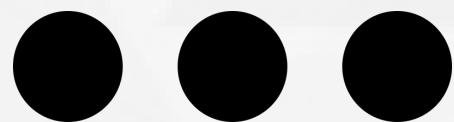
HTTP funcționează într-o abordare de cerere-răspuns.

Un sistem RESTful este compus dintr-un:

- client care solicită resursele.
- server care deține resursele.

Fiecare resursă are propriul URI, și acesta este doar o parte a cererii HTTP.

.



Este importantă metoda HTTP folosită pentru a efectua cererea. Există patru metode de bază:

GET - este folosit pentru a citi resursa.

POST - este folosit pentru a transfera date către server.

PUT - este folosit pentru a actualiza resursa existentă. Suprascrie întreaga resursă cu datele transferate.

DELETE - este folosit pentru a șterge o resursă existentă.

RESTRICȚIILE REST

REST se bazează pe unele constrângeri și principii care promovează simplitatea, scalabilitatea și lipsa stării în proiectare. Cele șase principii sunt:

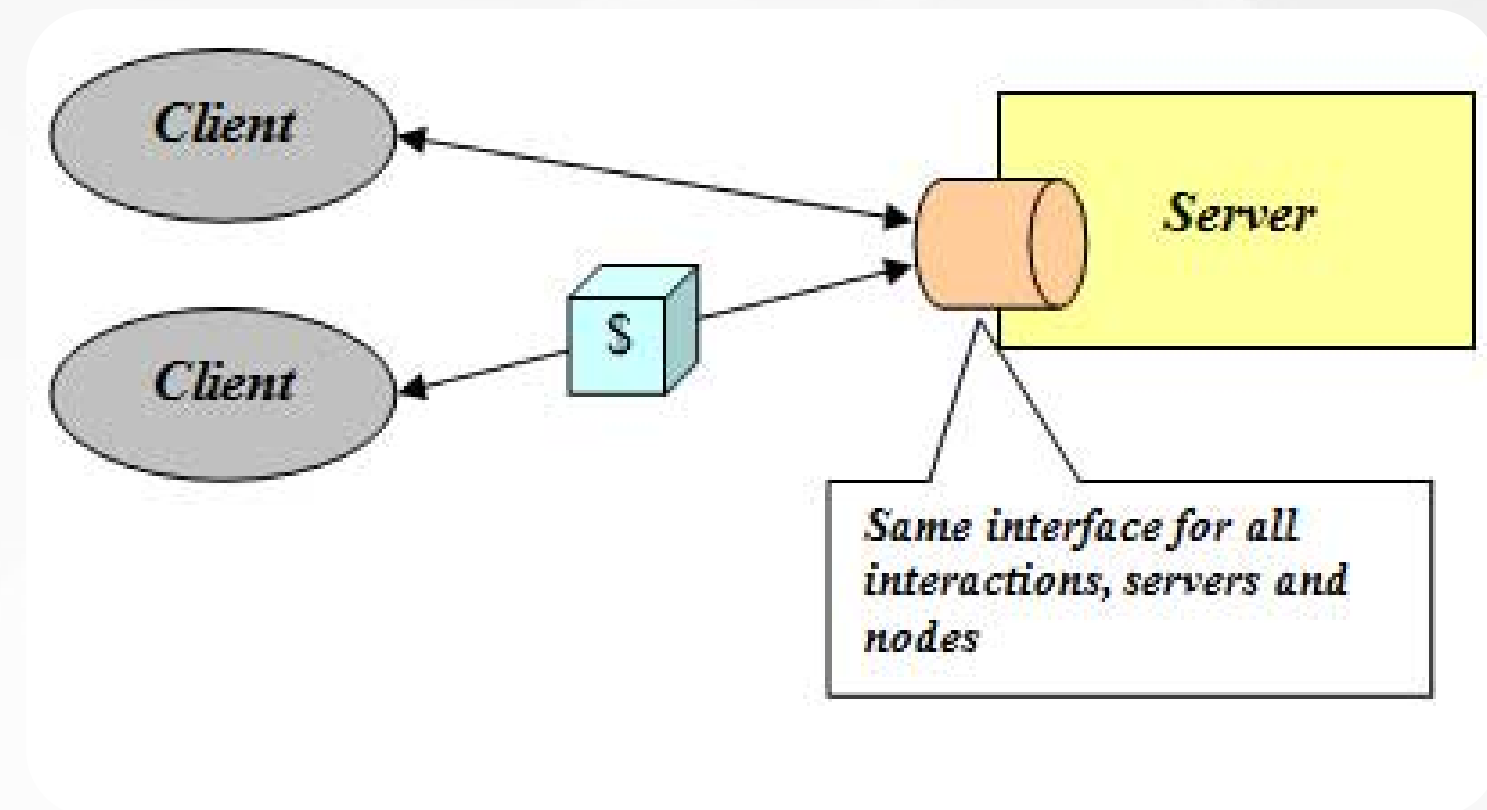


- INTERFAȚA UNIFORMĂ
- STATELESS
- CACHEABLE
- CLIENT-SERVER
- SISTEM STRATIFICAT
- COD LA CERERE

RESTRICTIILE REST

1 Interfața Uniformă

- O CONSTRÂNGERE CHEIE
- SUGEREAZĂ CĂ AR TREBUI SĂ EXISTE O MODALITATE UNIFORMĂ DE INTERACȚIUNE CU UN SERVER DAT, INDIFERENT DE DISPOZITIV.



BAZAT PE
RESURSE

Manipularea
Resurselor
Prin
Reprezentări

MESAJE
AUTO-
DESCRIPTIVE

HYPERMEDIA
CA
HATEOAS

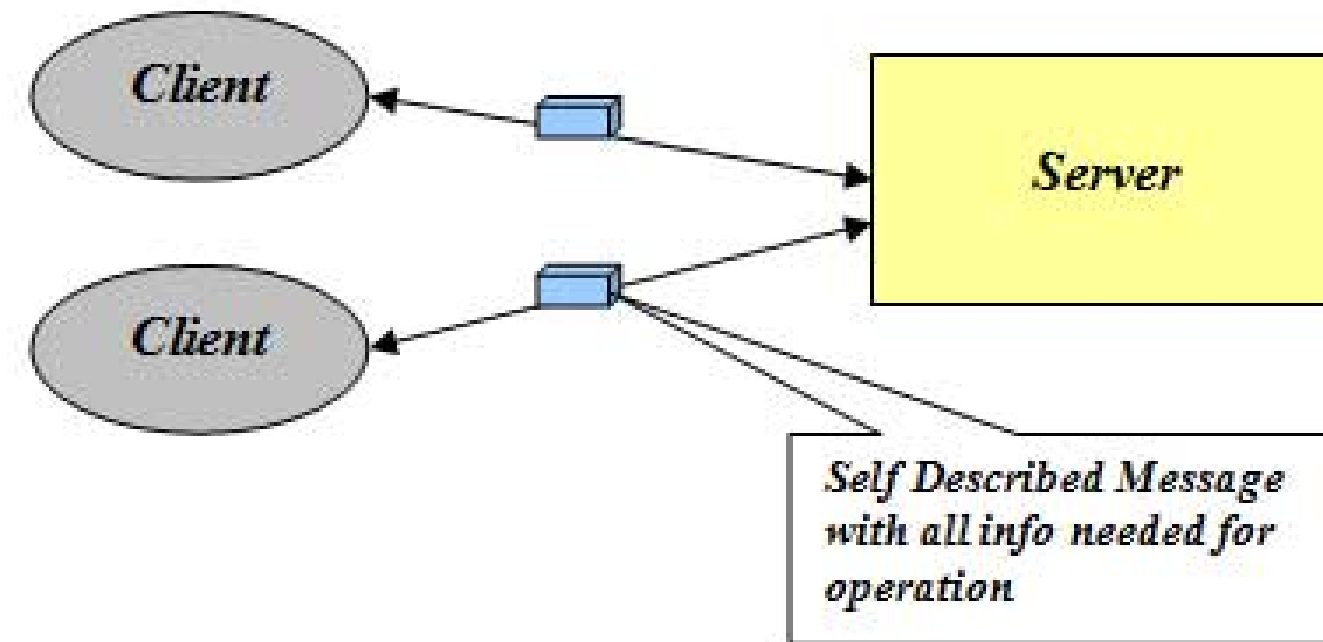
RESTRICȚIILE REST

2 Stateless

- Un API RESTful nu reține informații despre sesiunea utilizatorului și fiecare cerere trebuie să furnizeze date complete pentru a fi procesată, asigurând astfel o mai mare disponibilitate a API-ului.

3 Cacheable

- Serverul furnizează informații despre capacitatea de caching a răspunsului, ceea ce poate îmbunătăți performanța și reduce interacțiunile redundante client-server.



4 Client-Server

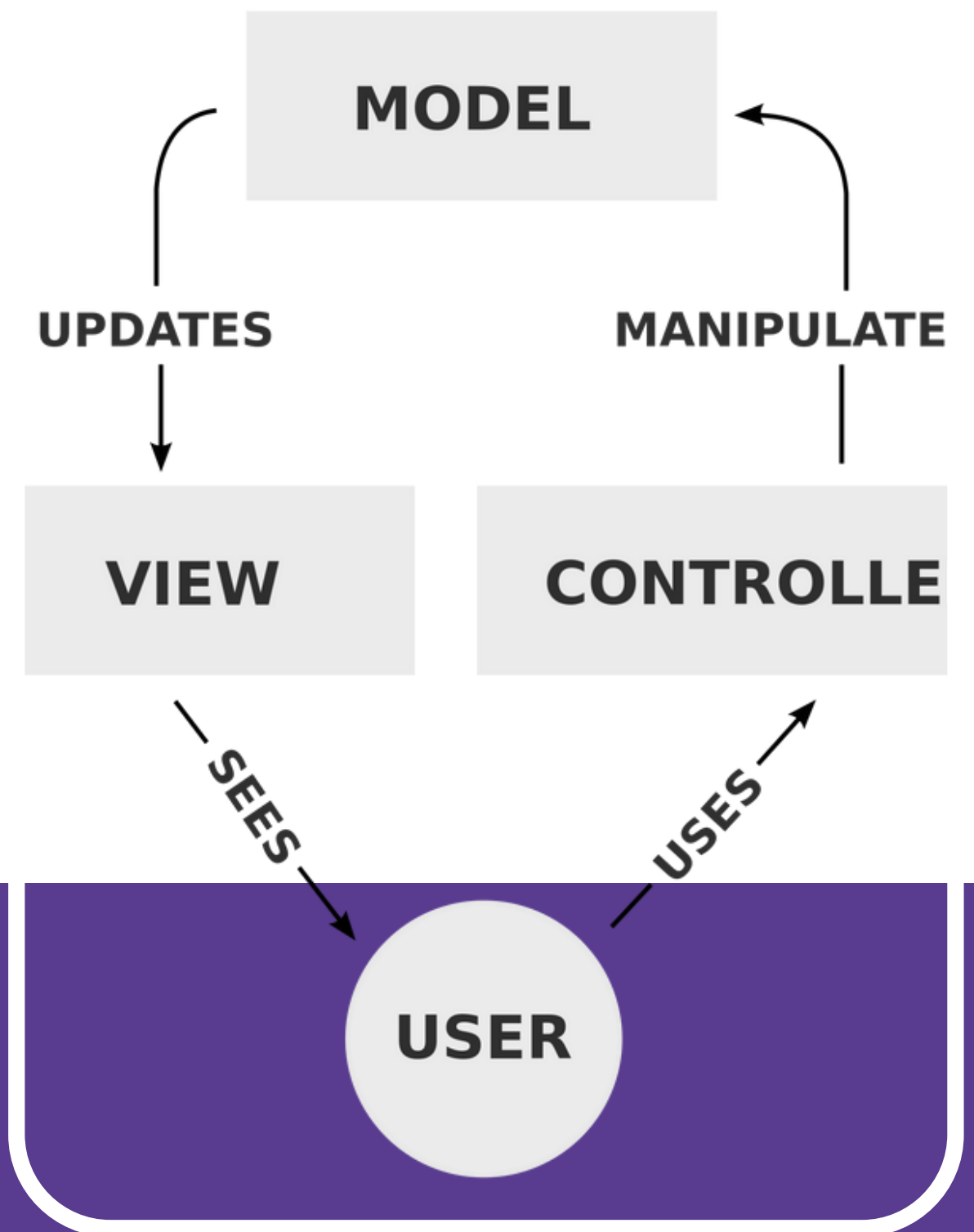
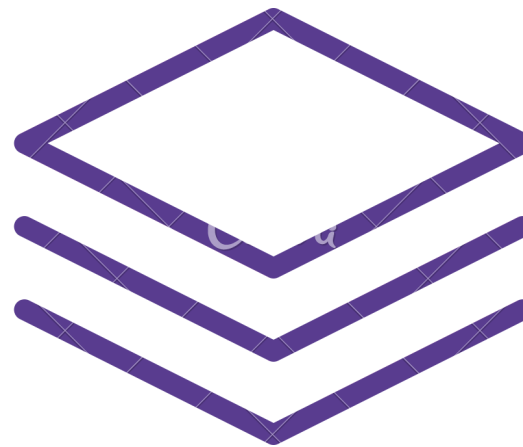
- Arhitectura REST este bazată pe o configurație client-server, unde clientul accesează resursele prin URI-uri, fără a cunoaște detalii despre procesarea cererii de către server, asigurând independența între cele două părți.

RESTRICȚIILE REST

5

Sistem stratificat

REST API-ul poate fi format din mai multe straturi care nu se afectează reciproc, permițând scalabilitatea și adăugarea de straturi suplimentare cum ar fi gateway-uri, proxy-urile.



6

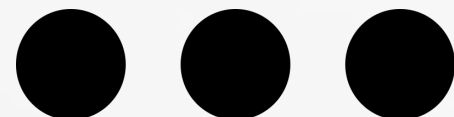
Cod la cerere

O restricție opțională care permite serverului să returneze cod executabil, în loc de date, pentru a oferi funcționalități specifice direct clientului, deși aceasta nu este o practică comună.



BENEFICIILE ALE ARHITECTURII REST

- Scalabilitate – Datorită separării între client și server, un produs poate fi scalat de către o echipă de dezvoltare fără prea multe dificultăți.
- Flexibilitate și portabilitate – Cererile pot simplifica migrarea datelor de la un server la altul și efectuarea modificărilor în bazele de date.
- Este simplu de construit și adaptat.
- Are un consum redus de resurse.
- Resursele pot fi reprezentate într-o varietate largă de formate de date (JSON, XML, etc.).



MULȚUMESC PENTRU ATENȚIE! 😊
