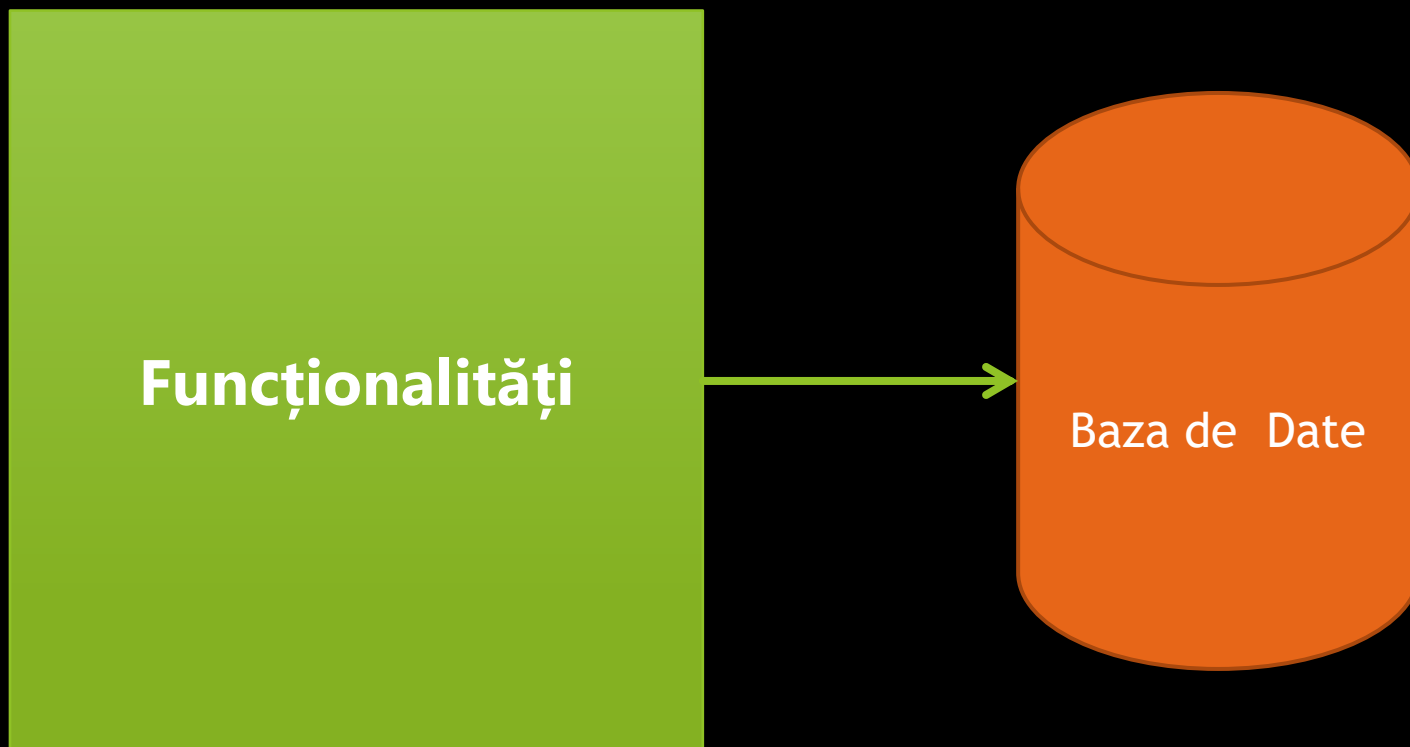


# Arhitecturi software bazate pe OData

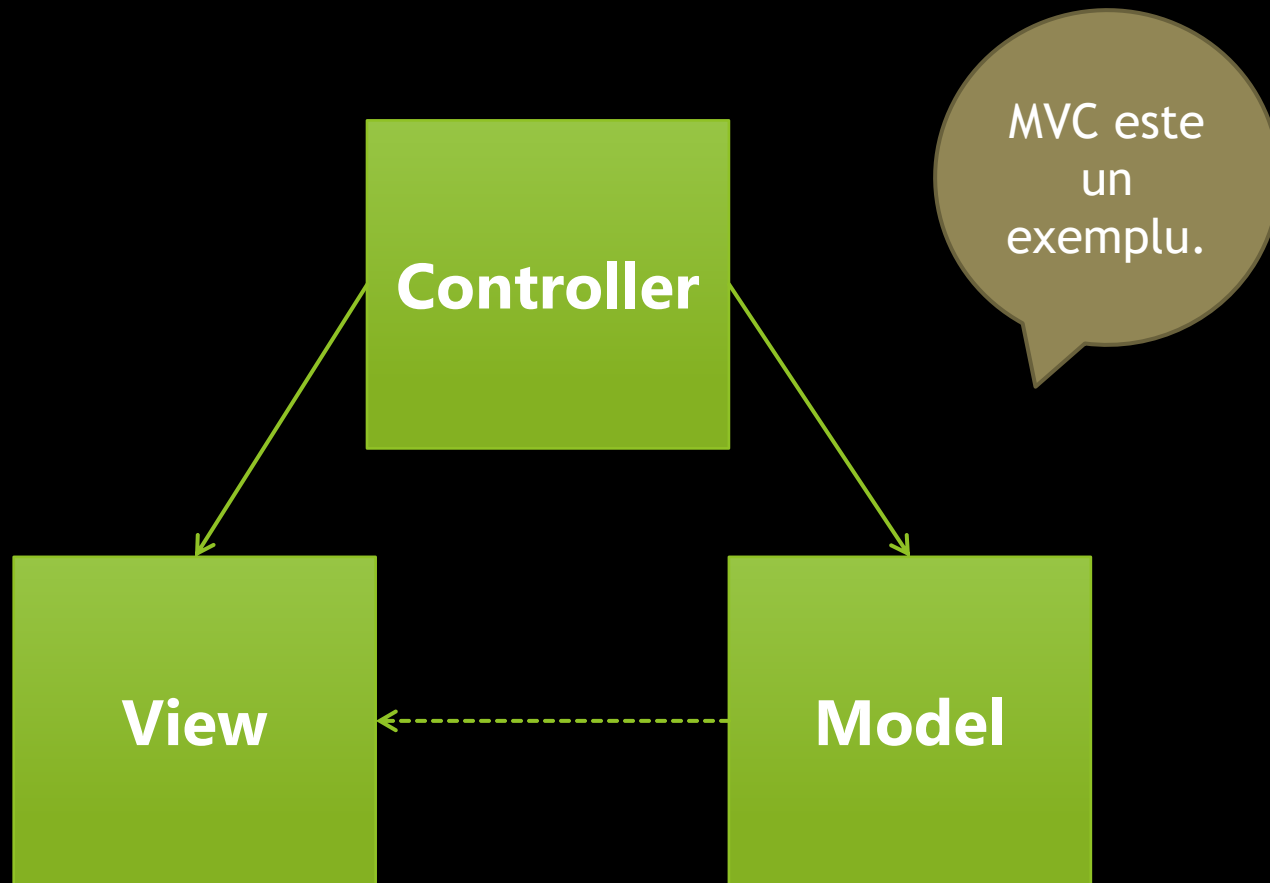
Conf.Dr. Cristian KEVORCHIAN

[ck@fmi.unibuc.ro](mailto:ck@fmi.unibuc.ro)

# Aplicații Monolit



# Interconectarea Părților



# Nivele arhitecturale

## Nivelul de prezentare

**Model**

**View**

**Controller**

## Nivelul logicii de business

**Workflow-  
uri**

**Componente**

**Entități**

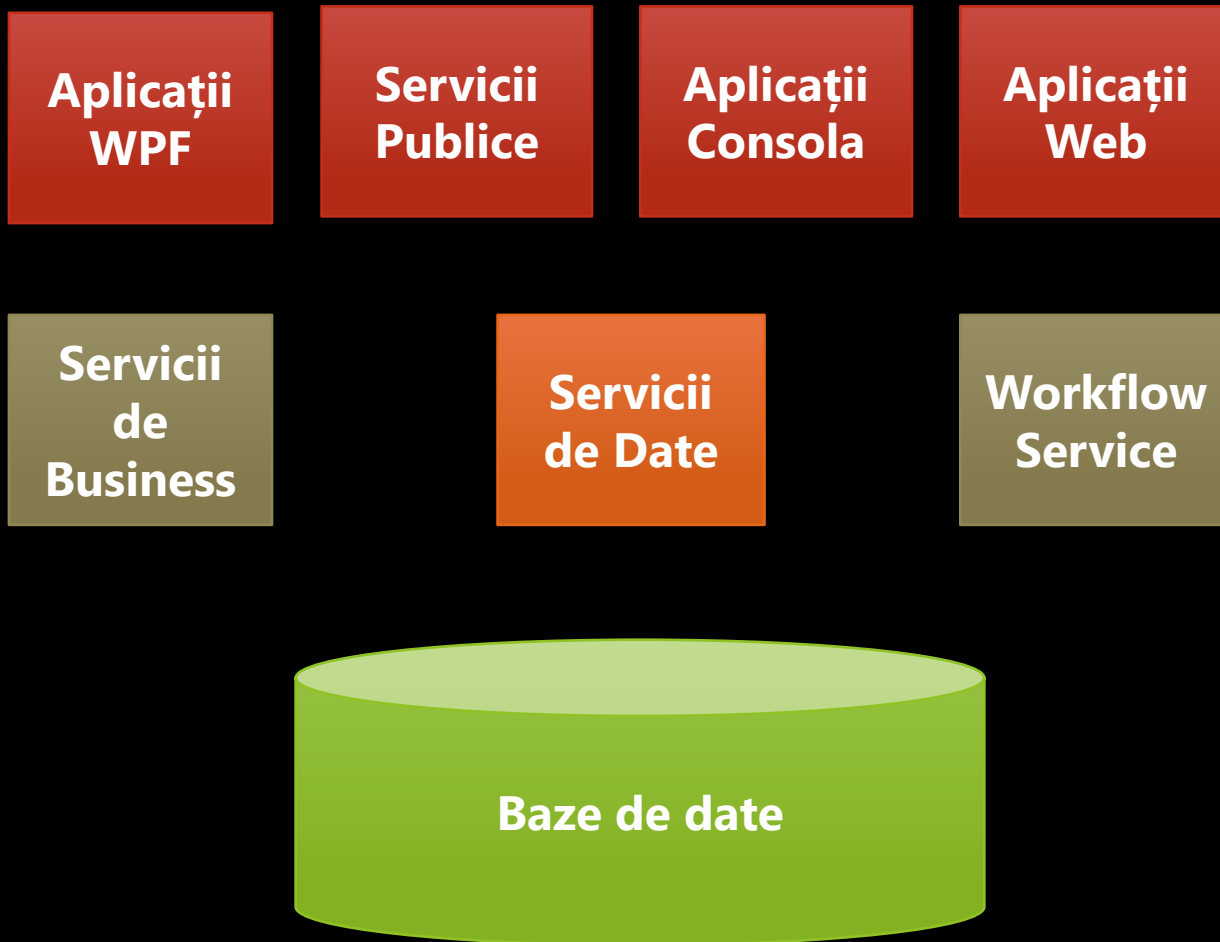
## Nivelul accesării datelor

**Componenta  
DAL**

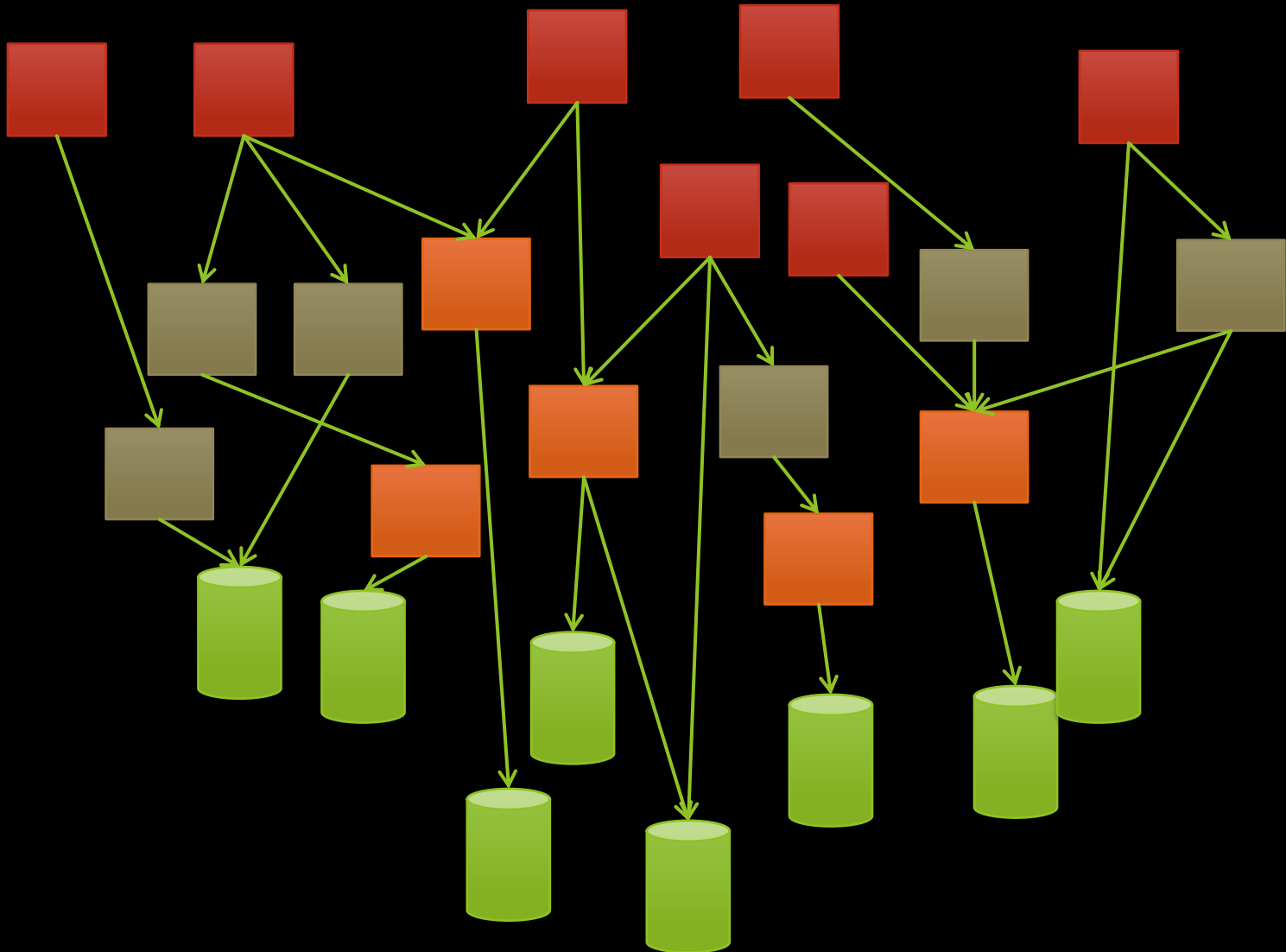
**Gateway**

**Armonizare a nivelelor  
arhitecturale**

# Aplicații "Enterprise"



# Ecosistemul "Enterprise"



# SOA(Service-Oriented Architecture)

- ▶ Contracte Standardizate
- ▶ Cuplare slabă
- ▶ Abstractizare
- ▶ Reutilizare
- ▶ Autonomie
- ▶ Fără păstrarea stării
- ▶ Posibilitate de identificare ușoară
- ▶ Compozabilitate

- ▶RPC(Remote Procedure Call)
- ▶REST(Representational State Transfer)



# REST

- ▶ Stil arhitectural
- ▶ Fără reținerea stării(Stateless)
  - ▶ Nu este întotdeauna adevărat
- ▶ Resurse

# ROA(Resource-Oriented Architecture)

- ▶ Expune mai multe URI(Uniform Resource Identifier)
  - ▶ Una pentru fiecare resursă
- ▶ Fără cooki-uri reprezentând ID-ul
- ▶ Agregarea se realizează prin consolidarea URI
- ▶ URI facilitează operații de tip “cut & paste”

# OData

- ▶ Open Data Protocol
- ▶ REST-based
- ▶ content-type
  - ▶ AtomPub
  - ▶ JSON
  - ▶ XML

Resurse de date relaționale

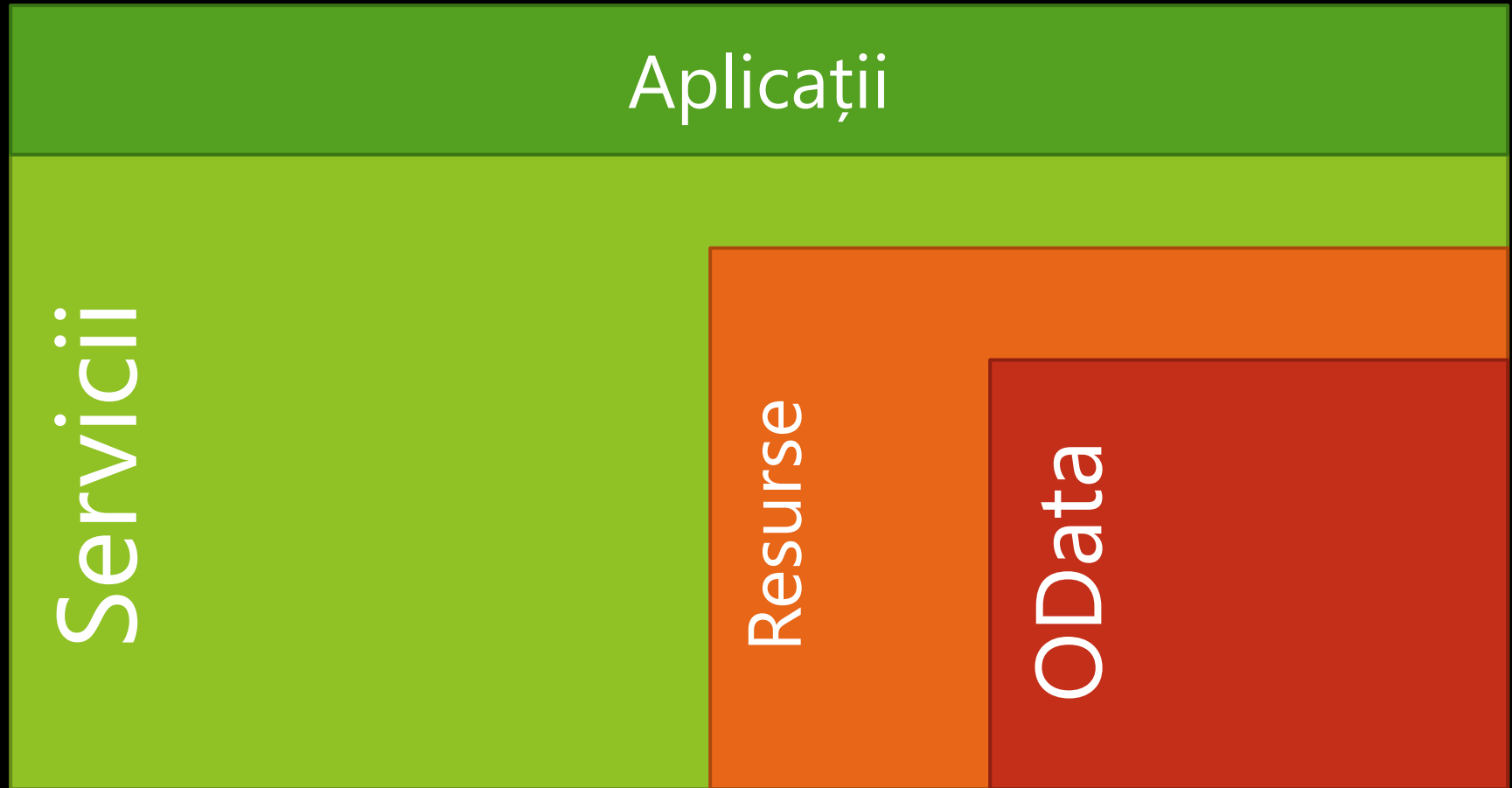
# Furnizori ODATA

**WCF Data  
Services**

**WCF RIA  
Services**

**SharePoint**

# Infrastructura "Enterprise"



# ROI (Resource-Oriented Infrastructure)



# Potențiale Servicii

## Nivelul de Prezentare

**Model**

**View**

**Controller**

## Nivelul Logicii de Business

**Workflow-  
uri**

**Componente**

**Entități**

## Nivelul de Accesare a Datelor

**Componente  
DAL**

**Gateway**

**Armonizare a nivelelor  
arhitecturale**



# SOA vs. ROA

- OData - standard bazat pe servicii web pentru operații asupra datelor prin utilizarea protocolului HTTP.
- **Limitare:** nu poate fi anticipată întreaga gamă de solicitări de date a de consumatorilor servicii web în context SOA
- ROA(Resources Oriented Architecture)-expune resurse în vederea accesării de către consumatorul de servicii. Similar cu abordarea bazata pe SQL a interogării datelor din baze de date.

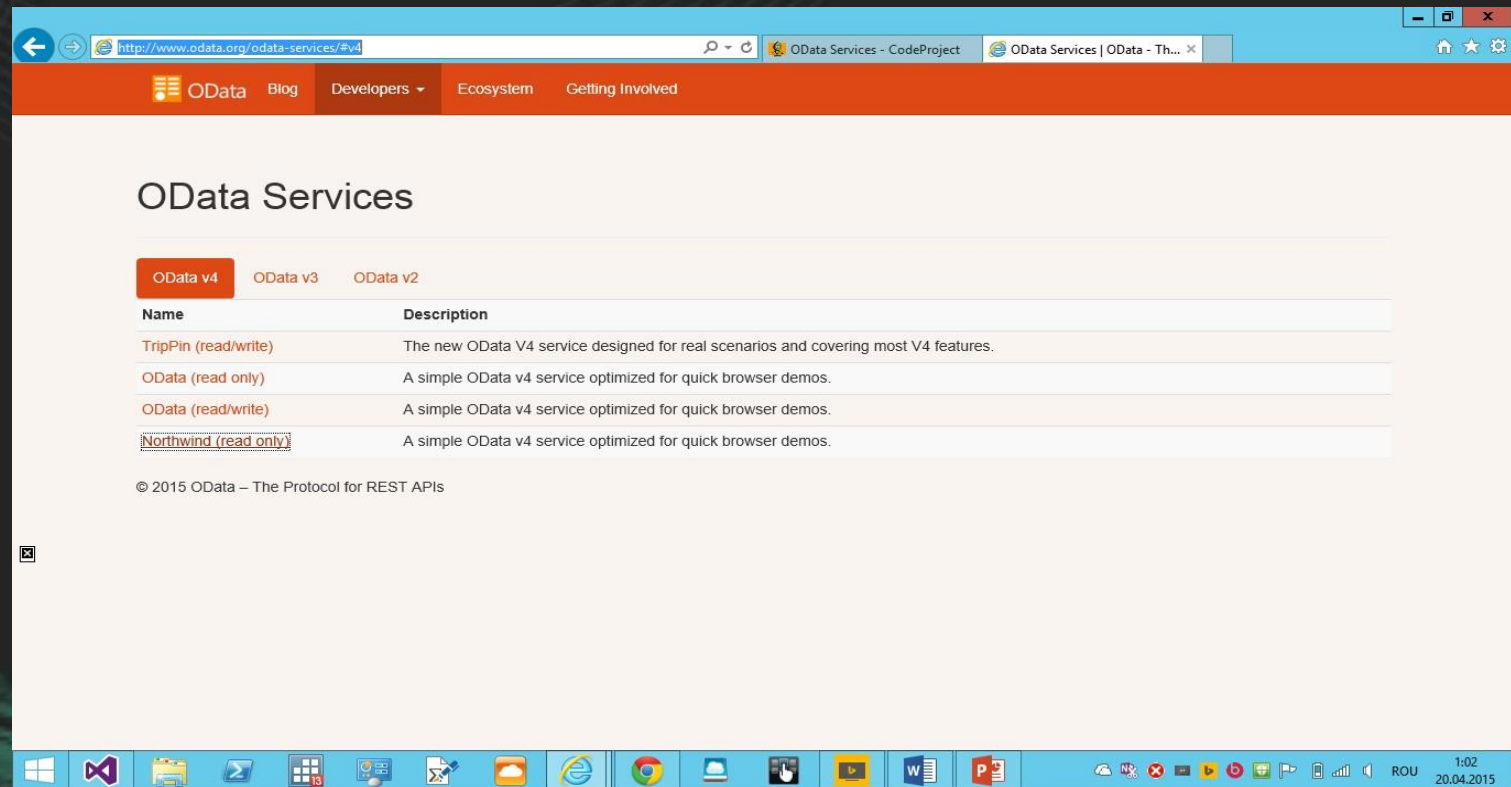
# OData

- Data (Open Data Protocol) este un standard internațional, care definește un set de bune practici pentru construirea și utilizarea API-urilor RESTful.
- Serviciile OData reprezintă servicii web care expun resurse, care pot fi accesate prin intermediul URL-ului.
- Protocolul OData specifică modul în care pot fi accesate date prin interogare via HTTP.
- Ideea este de a efectua interogări pe baza URL-ului cu parametri care să acceseze resursele expuse.

# Serviciile Odata expuse

Inventarul serviciilor Oudata expuse la URL:

- <http://www.odata.org/odata-services/>

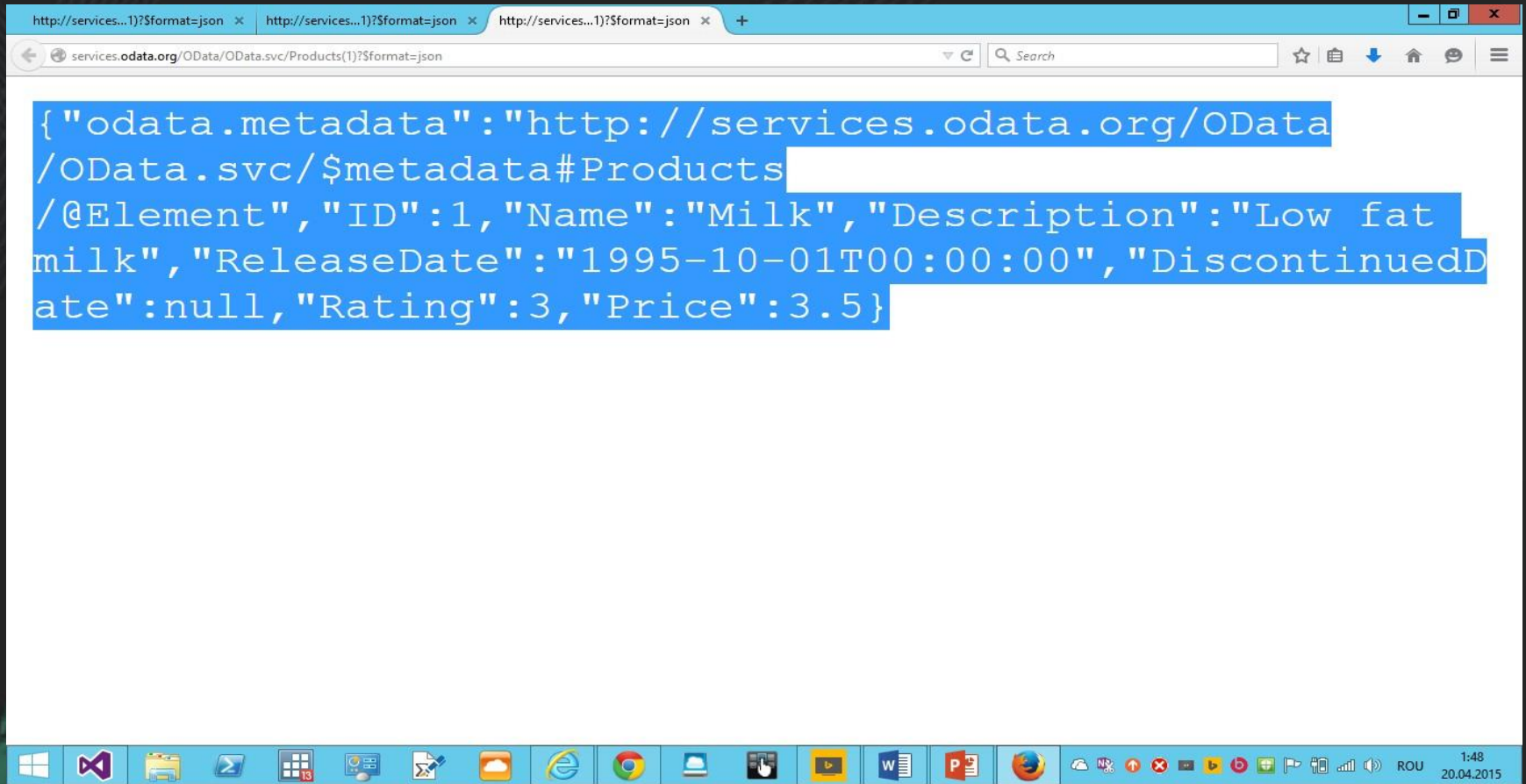


The screenshot shows a web browser window displaying the OData Services website. The page has a navigation bar with links for OData, Blog, Developers, Ecosystem, and Getting Involved. The main content area is titled "OData Services" and features a tabbed interface with "OData v4", "OData v3", and "OData v2". The "OData v4" tab is active, showing a table of services.

Name	Description
<a href="#">TripPin (read/write)</a>	The new OData V4 service designed for real scenarios and covering most V4 features.
<a href="#">OData (read only)</a>	A simple OData v4 service optimized for quick browser demos.
<a href="#">OData (read/write)</a>	A simple OData v4 service optimized for quick browser demos.
<a href="#">Northwind (read only)</a>	A simple OData v4 service optimized for quick browser demos.

© 2015 OData – The Protocol for REST APIs

[http://services.odata.org/OData/OData.svc/Products%281%29?\\$format=json](http://services.odata.org/OData/OData.svc/Products%281%29?$format=json)



# OData-Modelul(C#)

- ASP .NET Wep App->Empty(Web API)

- Models->class->Produs.cs

```
public class Produs
{
    public int ID {get; set;}
    public string Denumire {get; set;}
    public decimal Pret {get; set;}
    public string Categorie {get; set;}
}
```

*ID este cheia entitatii.*

***Se foloseste VS Scaffolding care utilizeaza “reflection” pentru a obtine tipul Produs.***



# OData - Controller-ul

- Un controller este o clasa ce realizeaza managementul cererilor HTTP.
- Se defineste cite un controller pentru fiecare entitate
- Se utilizeaza Scaffold-> **“Web API 2 OData Controller with actions using Entity Framework”**
- Se adauga un nou context



# Adăugăm EDM si Ruta

- ▶ App\_Start -> WebApiConfigs.cs
- ▶ Aceasta clasa cuprinde cod de configurare pentru Web API si are doua functionalitati:
  - EDM(un model abstract al datelor) pentru un endpoint OData
  - Se adaugă o ruta pentru acest endpoint al serviciului OData





- Package Manager->Enable-Migrations(Configuration.cs)  
... void Seed(...)  
{  
Context.Produse.AddOrUpdate(new Produs[] {  
    new Produs() {ID=1, Denumire="Costum Ceremonie",  
Pret=800,Categorie="Costume barbatesti"},,);  
}

## Generarea Bazei de Date(optional)



# Aplicație MVC Odata

DEMO

Muṭumesc!