

## Tema 2 P.S.S.C.

### *„Dezvoltarea de sisteme software extensibile folosind Managed Extensibility Framework”*

Managed Extensibility Framework (MEF) este una dintre cele mai noi platforme ce permite dezvoltarea de aplicații .Net extensibile. Platforma a fost creată din necesitatea sistemelor software complexe de a putea fi extinse cu noi funcționalități, respectiv cu noi module, într-o manieră simplă și eficientă. MEF îndeplinește cerințele companiilor de livrarea unui produs performant în timp util.

Managed Extensibility Framework are la bază trei concepte care descriu complexitatea și utilitatea acestei platforme.

#### **1. Extensibilitate**

Acest concept permite adăugarea și/sau înlocuirea unor funcționalități existente, fără a aduce modificări acelui sistem. Cel mai important aspect de menționat aici este faptul că extensibilitatea se realizează la execuție. Extensiile sunt integrate și injectate în sistem la rularea aplicației, și nu este nevoie de re-compilare, pentru ca extensiile să poată fi detectate și utilizate ca părți componente ale acelei aplicații.

*Librăria* System.ComponentModel.Composition găzduiește atributele ce vor fi folosite în realizarea compunerii și câteva clase destinate tratării diferitelor excepții.

#### **2. Detecție**

Acest concept face referire la modul în care detectăm *plug-in*-urile ce vor fi încărcate în sistemul nostru extensibil, decuplat. Premisa de bază este următoarea: la crearea unei clase, dorim ca acea clasă să realizeze ceva specific, un serviciu.

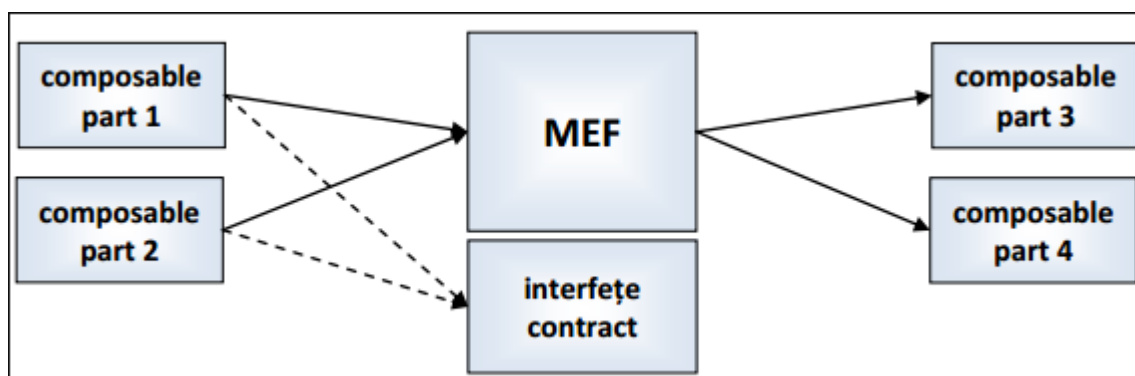
Detecția, reprezintă mecanismul de detectare și furnizare a implementărilor astfel încât celelalte componente ale sistemului să le poată utiliza, la nevoie.

#### **3. Metadata**

Este tehnologia MEF prin care se pot furniza informații adiționale despre componente. Prin intermediul metadata, aplicația poate să filtreze și să examineze componentele, fără a cunoaște detalii despre implementarea

acestora. Metadata oferă o abordare contextuală de a determina ce componente sunt potrivite pentru mediul curent, filtrându-le în mod corespunzător și apoi, încărcându-le.

### Arhitectură bazată pe MEF

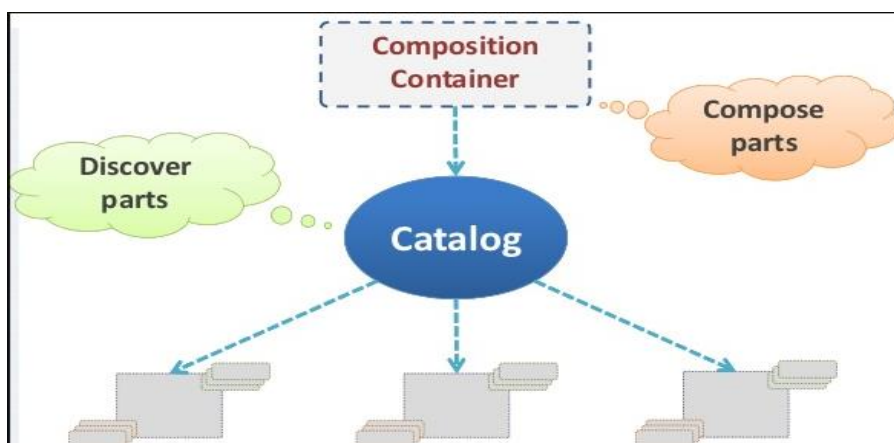


Sistemul software este compus din așa numitele părți componente (părți ce pot fi compuse). O astfel de parte oferă servicii altor părți și la rândul ei consumă servicii oferite de alte părți.

Principalele:

- **Export** - reprezintă un serviciu oferit de o parte
  - aceeași parte poate exporta mai multe servicii
  - se realizează prin decorarea unei clase sau a unei proprietăți cu atributul *Export*;
- **Import** - reprezintă un serviciu consumat de o parte
  - o parte poate consuma unul sau mai multe servicii
  - se realizează prin decorarea proprietăților cu atributul *Import*
- **Contract** - un contract este un identificator pentru un export sau un import.
- **Compunere** - părțile sunt compuse de către MEF care le instanțiază și apoi face legătura între importuri și exporturi.

### Arhitectura MEF la nivel general



**Catalogul** reprezintă mecanismul de detecție automată a părților, la execuție, astfel MEF oferă *out-of-the-box*, patru tipuri de cataloage: *Type catalog*, *Assembly catalog*, *Directory catalog*, *Aggregate catalog*.

**Container-ul** este responsabil pentru gestionarea compunerii, asociind exporturile cu importurile în mod corespunzător. Cel mai comun tip de *container* este *CompositionContainer*, iar compunerea propriu-zisă se efectuează la apelul metodei *ComposeParts*