

Universitatea “Politehnica” din București  
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

*Arhitectura kernelului sistemului de operare Windows Phone 8.  
Legătura cu Windows 8*

## **Proiect SOA**

*Master Ingineria Informației și Sistemelor de Calcul  
Anul I*

Conducător științific

*Dr.Ing. Ștefan STĂNCESCU*

Student

*Ing. George – Daniel ȘERBĂNESCU*

2014

# Cuprins

## 1. CONCEPTE ȘI ARHITECTURA

- 1.1. ARHITECTURA SISTEMULUI DE OPERARE
- 1.2. LEGATURA CU VERSIUNILE ANTERIOARE
- 1.3. STIVA PLATFORMEI
- 1.4. TIPURI DE APLICAȚII

## 2. ARHITECTURA KERNELULUI

- 2.1. DESCRIERE
- 2.2. PĂRȚI COMUNE CU WINDOWS 8

## 3. PROCESE. EXECUTARE IN FUNDAL( BACKGROUND )

- 3.1. SERVICIILE SISTEMULUI DE OPERARE
- 3.2. SECURITATE

## 4. CONCEPTUL XAML PENTRU WINDOWS PHONE OS

- 4.1. XAML. DESCRIERE
- 4.2. DEPENDENȚA PROPIETĂȚILOR
- 4.3. EVENIMENTE PENTRU WINDOWS PHONE OS
- 4.4. DATA BINDING
- 4.5. MODEL-VIEW-VIEWMODEL

## 5. CONCLUZII

## BIBLIOGRAFIE

# Capitolul 1. CONCEPTE ȘI ARHITECTURA

## 1.1. ARHITECTURA SISTEMULUI DE OPERARE

Când Windows Phone 7 a fost lansat în toamna anului 2010, a reprezentat o îndepărtare semnificativă, nu numai de la sistemele anterioare Microsoft de operare mobile, dar de asemenea de la orice alt sistem de operare mobil (OS) de pe piață.

## 1.2. LEGATURA CU VERSIUNILE ANTERIOARE

Chiar dacă distinctiv UX nu s-a modificat semnificativ între Windows Phone 7 și Windows Phone 8, au existat schimbări dramatice la suprafață. Pentru prima dată, Windows Phone este construit pe aceeași tehnologie ca și omologul său PC. În această secțiune, voi descrie cele două părți esențiale ale acestei schimbări cu impact pentru dezvoltatori: părțile de bază pentru Windows partajate, precum și adoptarea de Windows Runtime.

## 1.3. STIVA PLATFORMEI

Nici un capitol de arhitectură nu ar fi complet fără schema bloc, figura de mai jos prezintă componentele logice de bază ale platformei Windows Phone 8.

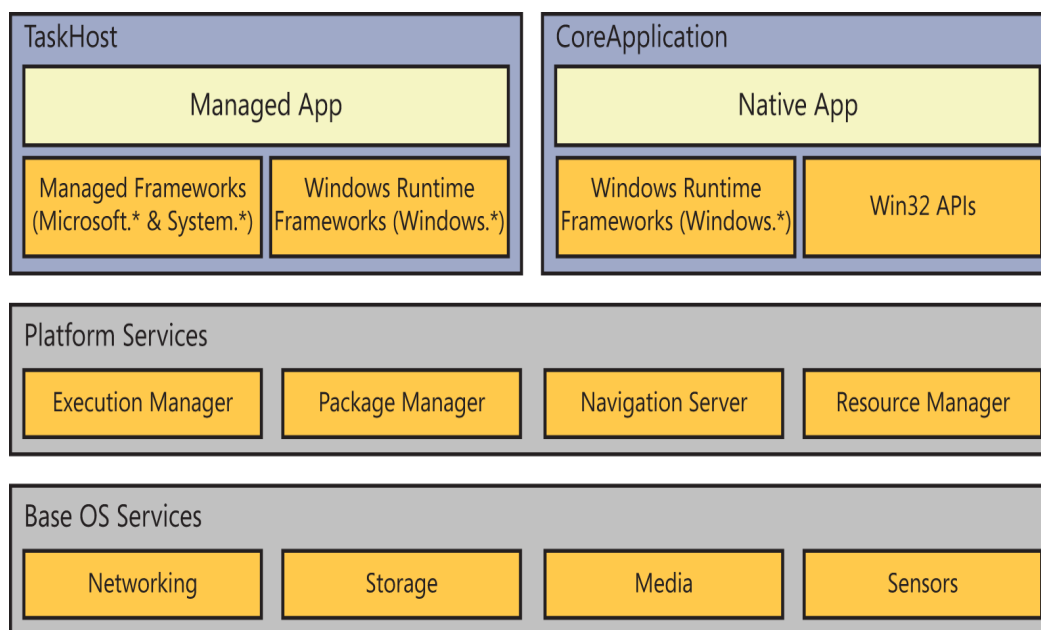


Figura 1. Stiva componentelor logice ale sistemului de operare

În partea de sus a stivei sunt plasate două modele distincte de aplicații. Caseta cu eticheta "TaskHost" reprezintă modelul de aplicație XAML, care a fost modelul primar de la lansarea Windows Phone 7. La dreapta ei este o cutie cu eticheta "CoreApplication," un nou model de aplicație pentru Windows Phone, care este un subset al noului model app Windows 8. În lansarea Windows Phone 8, acest model de aplicație acceptă doar aplicații native pure folosind Direct3D pentru UI.

## 1.4. TIPURI DE APLICAȚII

Până în prezent, am fost referindu-se la aplicații pentru Windows Phone generic, ca și în cazul în care acestea au fost construite și a funcționează în esență același mod. De fapt, Windows Phone 8 suportă mai multe tipuri de aplicații diferite, specifice .

Tipul aplicației	Descriere	Limbaje suportate	UI framework	API-uri suportate
XAML	Cel mai comun tip de aplicație pentru Windows Phone 7.x. Aceste aplicații sunt scrise exclusiv în XAML și cod gestionat.	C#, Visual Basic	XAML	Microsoft .NET, Windows Phone API, Windows Runtime API
Mixed	Aceste aplicații urmează structura	C#,	XAML, Direct3D	.NET Windows

Tipul aplicației	Descriere	Limbaje suportate	UI framework	API-uri suportate
mode	<p>app XAML, dar permite includerea de cod nativ înfășurat într-o componentă pentru Windows Runtime. Acest lucru este bine-potrivite pentru aplicații pentru care doriți să reutilizați o bibliotecă nativ existent, mai degrabă decât rescrierea-l în cod gestionat. De asemenea, este util pentru cazurile în care doriți să scrie cea mai mare a aplicației în cod nativ (inclusiv grafica Direct3D), dar, de asemenea, nevoie de acces la cadrul XAML UI și unele dintre caracteristicile care sunt disponibile numai pentru XAML aplicații, cum ar fi capacitatea de a crea și manipula ecranul de pornire gresie.</p> <p>Cel mai potrivit pentru jocuri, aplicații pur native folosind Direct3D oferă posibilitatea de a extrage maximum de hardware de bază a telefonului. De asemenea, deoarece acestea se bazează pe modelul App pentru Windows, acestea oferă cel mai mare grad de cod de partajare între Windows și Windows Phone.</p>	Visual Basic, C/C++	(via DrawingSurface)	Phone API, Windows Runtime API, Win32/COM API (within Windows Runtime components)
Direct3D	<p>Cel mai potrivit pentru jocuri, aplicații pur native folosind Direct3D oferă posibilitatea de a extrage maximum de hardware de bază a telefonului. De asemenea, deoarece acestea se bazează pe modelul App pentru Windows, acestea oferă cel mai mare grad de cod de partajare între Windows și Windows Phone.</p>	C/C++	Direct3D	Windows Runtime API Win32/COM API

Table 1. Tipuri de aplicații pentru Windows Phone 8

## Capitolul 2. ARHITECTURA KERNELULUI

### 2.1. DESCRIERE

Deși API-urile Win32/COM sunt prezentate doar în caseta CoreApplication în Figura anterioară, ele sunt de fapt executate de aplicații administrate, precum și, atâta timp cât acestea sunt amplasate într-o componentă pentru Windows Runtime.

Cele două modele de aplicații se bazează pe un set comun de servicii al platformei de bază.

- **Manager de pachete** - este responsabil pentru instalarea / deinstalarea de aplicații și menținerea tuturor metadatelor de-a lungul ciclului de viață al aplicației. Se păstrează nu numai evidența pentru aplicațiile instalate și licențiate, persistă, de asemenea, informații despre orice interacțiune a care utilizatorul ar putea fi fixate la ecranul de pornire și punctele de extensibilitate pentru care o aplicație s-ar putea sau a fost înregistrată, astfel încât acesta este poată fi apărut în în loc potrivit din sistemul de operare.
- **Manager de execuție** - controlează logica asociată cu durata de viață a execuție aplicației. Se creează procesul de hosting pentru aplicația să ruleze în ridică și evenimentele asociate cu pornire app / oprire / dezactivare. Se efectuează o sarcină similară pentru procesele de fundal, care include, de asemenea, programarea corectă a acestor sarcini.
- **Server de Navigare** - gestionează toate interacțiunile dintre aplicații din backgroundul terminalului mobil. Când se atinge o icoana a unei aplicații pe ecranul de Start, când se navighează de la "Start app" până la aplicația aleasă serverul de navigare este responsabil pentru retransmiterea semnalului către managerul de execuție pentru a decide momentul când aceasta poate fi pornită. De asemenea, atunci se apăsă și se ține apăsată tasta Back și se alege o aplicație care a fost deschisă anterior, serverul de navigare este responsabil pentru a spune serverului de execuție care aplicație trebuie reactivată.
- **Managerul de Resource** - este responsabil cu asigurarea faptului că telefonul este întotdeauna rapid și receptiv prin monitorizarea utilizării resurselor de sistem (în special CPU și memorie) de către toate procesele active și aplicarea unui set de constrângeri pe ele. În cazul o aplicație sau un proces de fundal depășește limita de resurse alocată, acesta este terminat pentru a menține starea de bună funcționare generală a telefonului.

## 2.2. PĂRȚI COMUNE CU WINDOWS 8

### Core comun

De departe cea mai semnificativă schimbare de arhitectură din Windows Phone 8 este adoptarea unui nucleu comun cu Windows OS. Nucleul comun conține două componente distincte. În partea de jos este sistemul Core Windows, funcțiile de bază ale sistemului de operare Windows, inclusiv ( printre altele ), nucleul NT, sistemul de fișiere NT ( NTFS ), și stiva de rețea. Acest nucleu minim este rezultatul multor ani de rafinament arhitectural, al cărui scop a fost de a oferi o bază comună, care ar putea alimenta mai multe dispozitive, inclusiv smartphone-uri.

Deasupra Core System este Mobile Core, un set de funcționalități pentru Windows care nu este parte a sistemului de bază, dar care este încă relevant pentru un smartphone. Aceasta include componente cum ar fi multimedia, CoreCLR și Trident, motorul de randare pentru Internet Explorer. Figura de mai jos ilustrează unele din componentele comune pe care se bazează Windows și Windows Phone. Este de remarcat faptul că Mobile Core este doar o entitate arhitecturală distinctă în Windows Phone. Ferestrele conțin aceleași componente ca Mobile Core, dar ele sunt parte dintr-un set mai mare de funcționalități. Acest lucru este reprezentat printr-o linie punctată în jurul componentelor de bază mobile în partea cu Windows 8 a diagramei.

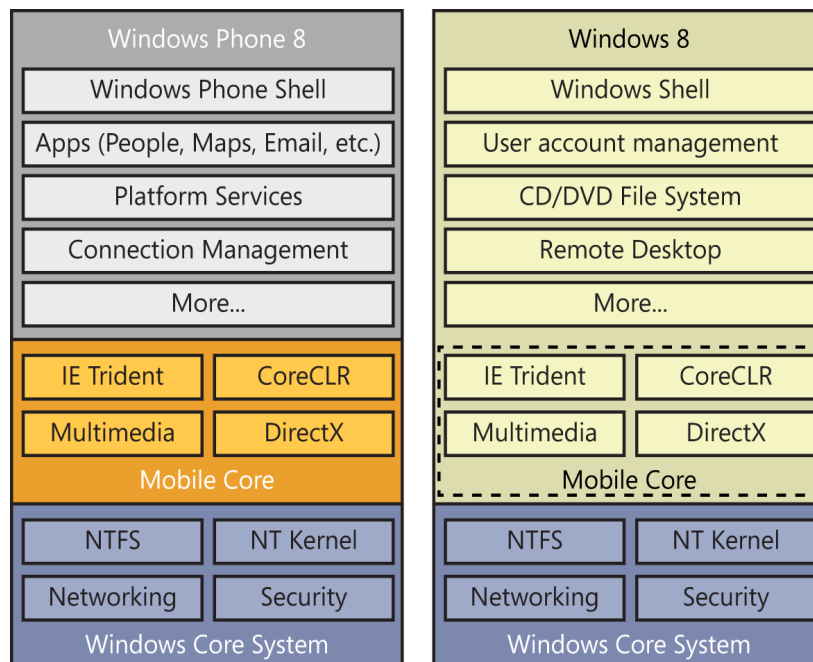


Figura. Windows 8 și Windows Phone 8 au un core comun

## Capitolul 3. PROCESE. EXECUTARE IN FUNDAL( BACKGROUND )

Cand vine vorba de execuție a proceselor de fundal pe un dispozitiv mobil, utilizatorii au obiective conflictuale . Pe de o parte , se dorește ca aplicațiile să continue furnizarea de funcționalitate chiar și atunci când un utilizator nu interacționează direct cu terminalul mobil - streaming de muzică de pe Internet , actualizarea iconițelor live cu cele mai recente date meteorologice , sau furnizarea de instrucțiuni de navigare prin GPS. Pe de altă parte , se dorește ca durata de viață a bateriei telefoanelor să fie valabilă pe parcursul unei zi de funcționare.

Se încearcă echilibrarea acestor cerințe contradictorii in ceea ce privește o abordare de scenariu concentrat pentru prelucrarea proceselor din background. Mai degrabă decât a permite rularea pur și simplu de aplicații în mod arbitrar în fundal pentru a efectua toate aceste funcții , platforma oferă un set de caracteristici orientate multitasking concepute pentru a satisface nevoile ( și constrângerile ) de scenarii specifice .

### 3.1. SERVICIILE SISTEMULUI DE OPERARE

#### **Servicii de sistem de operare de fundal**

Windows Phone oferă un set de servicii de fundal, care pot efectua sarcini comune în numele aplicațiilor.

#### **Serviciul de transfer de fundal**

Serviciul de transfer de fond (BTS) face posibilă efectuarea de transferuri HTTP care utilizează aceeași infrastructură robustă ca sistemul de operare pentru a efectua operații cum ar fi descărcarea de muzică. BTS se asigură că descărcări sunt persistente chiar dacă dispozitivul repornește și că acestea nu au impact asupra traficului în rețea a aplicațiilor din prim-plan.

#### **Sarcinile programate**

Sarcinile programate oferă soluția cea mai generică pentru prelucrare de fundal din Windows Phone . Există două tipuri de sarcini planificate pe care o aplicație le poate crea, fiecare dintre acestea fiind programate și gestionate de sistemul de operare , în funcție de anumite condiții :

- **Sarcini periodice** - rulează pentru o scurtă perioadă de timp la un interval



regulat de aproximativ 25 secunde la fiecare 30 minute ( atat timp cât telefonul nu este în modul de economisire a baterie ) . Acestea sunt destinate pentru îndeplinirea de sarcini mici care beneficiază de execuție frecvente . De exemplu , o aplicatie pentru vreme care descărca cea mai recentă prognoză de la un serviciu web și apoi să actualizeze datele aplicației.

- **Sarcini sarcini mari consumatoare de resurse** - pot rula pentru o perioadă mai lungă , dar nu sunt rulate la intervale previzibile. Deoarece acestea pot avea un impact mai mare asupra performanțelor dispozitivului, se executa numai atunci când dispozitivul este conectat, aproape complet încărcată , la Wi - Fi , și nu în utilizarea activă . Agenții consumatori de resurse în mod intensiv sunt destinați unor operațiuni mai solicitante , cum ar fi sincronizarea unei baze de date cu un server de la distanță .

### **Execuție în fundal pentru urmărirea locației**

În cazul redării muzicii de fundal, este foarte puțin cod care trebuie rulat după ce configurația inițială este completă. Infrastructura de redare audio controlează redarea unui sunet real, iar utilizatorul efectuează în general sarcini , cum ar fi play, pauză , și schimbări de fișiere audio de folosind built -in universal Volume Control ( UVC ) , mai degrabă decât redeschiderea aplicației în sine . În mare parte tot ceea ce trebuie să facă aplicația este să furnizeze URL-uri pentru conținut audio și metadate ( sau conținut audio de streaming ) la serviciul audio .

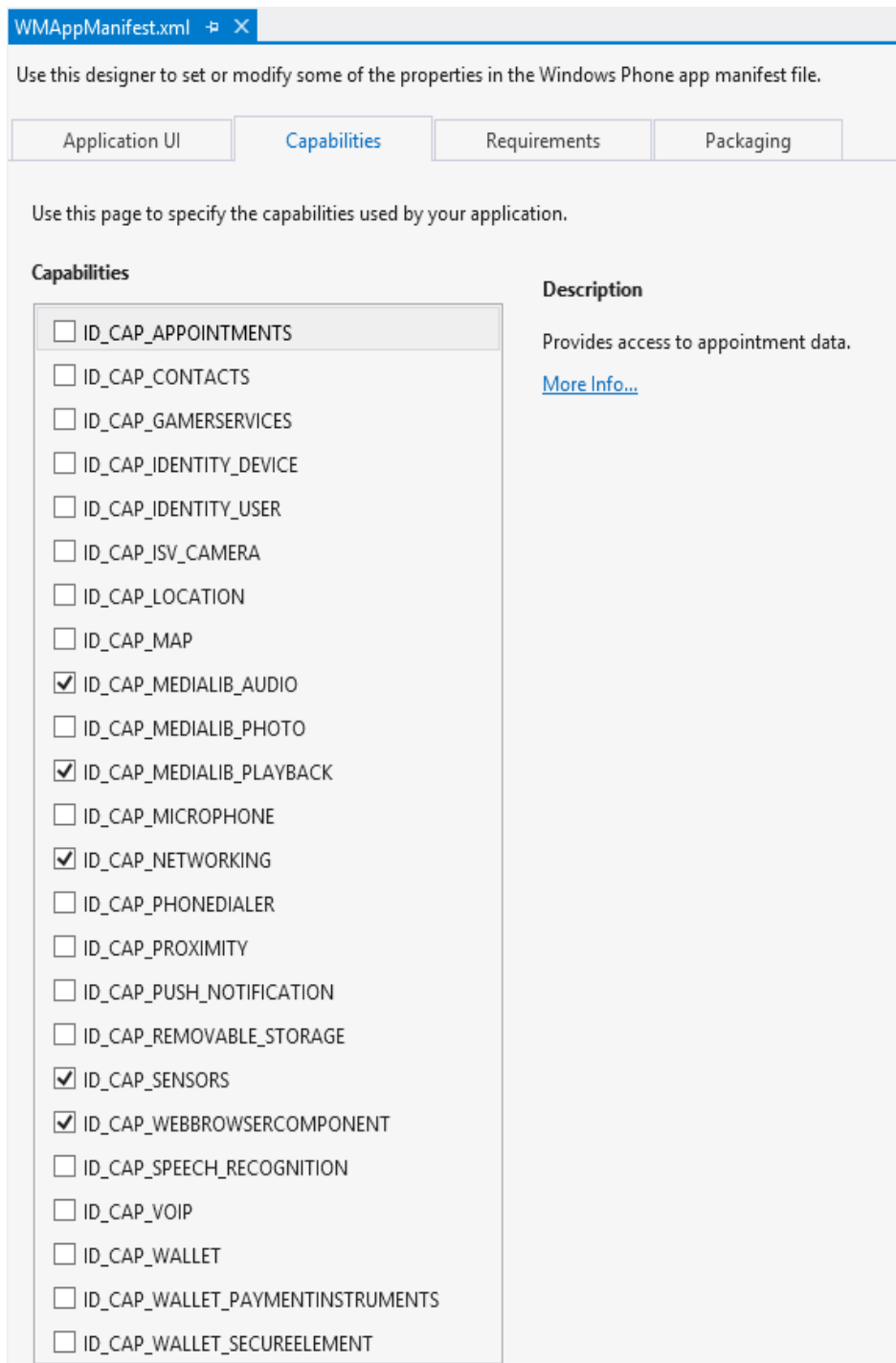
Acesta nu este cazul pentru urmărirea locației și , în special , aplicațiilor de navigare . Aceste aplicații , în general, au nevoie să primească și să proceseze informații actualizate despre locație la fiecare câteva secunde, pentru a determina dacă utilizatorul ar trebui să facă să vireze la stânga sau la dreapta . Acestea pot să ofere un expriență de utilizare complexă în cadrul aplicației , cum ar fi o hartă care arată traseul complet până la destinație și timpul / distanța de parcurs. Ca urmare , modelul de redare audio utilizând o sarcină de fond constrâns este mai potrivită în acest caz. În schimb , Windows Phone 8 introduce un concept cunoscut sub numele Executare continua în fundal ( CBE ) , care pur și simplu se referă la capacitatea unei aplicații curente de a continua să fie rulate, chiar dacă utilizatorul navighează mai departe .

## 3.2. SECURITATE

Smartphone-uri moderne sunt , de departe, elementele cele mai personale pe care oamenii le au în proprietate - în device sunt memorate sunt numele , numerele de telefon și adresele tuturor persoanelor din familie și ale prieteni , mii de fotografii, istoria de localizare, e-mail și informațiile financiare stocate în portofelul aplicațiilor mobile . Asigurându-se că toate aceste informații rămâne în condiții de siguranță în timp ce se deplasează cu telefonul între locații fizice și navighează pe o varietate de site-uri și aplicații necesită un model robust de securitate .

Modelul de securitate Windows Phone se bazează pe noțiunea de camere de securitate, care sunt grupate în containere izolate, în care procesele sunt create și executate. Containerul este principalul securitate la care sunt acordate drepturi de acces în sistem . Sistemul acordă aceste drepturi , bazate pe principiul de securitate de lungă durată, care susține că o aplicație nu ar trebui să aibă doar drepturile care sunt strict necesare pentru a îndeplini funcțiile sale declarate . De exemplu , aplicația de e-mail nu ar trebui să aibă capacitatea de a porni în mod arbitrar aparatul foto și a face o fotografie .

În mod explicit fiecare cameră de securitate , dacă acesta conține cod deținut de Microsoft sau de către un dezvoltator de software extern , începe cu un set limitat de privilegii - suficient pentru a activa o aplicație de sine stătătoare , cum ar fi un calculator sau un joc simplu , dar nu suficiente pentru a permite gamă completă de scenarii consumatorilor așteptate de la un smartphone modern. Dacă o aplicație dorește să acceseze resurse care se află în afara camerei sale, cum ar fi trimiterea de trafic în rețea sau citirea de la contactele utilizatorului, acestea trebuie să îi fie acordat în mod explicit accesul prin capabilități. Capabilități de a acționa ca un set de mecanisme de control care furnizează accesul către resurse sensibile. Sistemul trebuie să acorde în mod explicit capabilități pentru o cameră .



*Figura. Se pot selecta capabilitățile explicit pentru o cameră în meniul de manifest al proiectului*

## Capitolul 4. CONCEPTUL XAML PENTRU WINDOWS PHONE OS

### 4.1. XAML. DESCRIERE

Extensible Markup Language Application (XAML) este un limbaj declarativ, în mod specific, poate inițializa obiecte și seta proprietăți ale obiectelor, folosind o structura de limbaj care arată relațiile ierarhice între mai multe obiecte, și folosind o convenție tip de suport care acceptă extinderea tipurilor. Se pot crea elemente de UI vizibile în marcaje XAML declarative. Se pot folosi clase create în C# separat pentru a răspunde la evenimente și manipula obiectele pe care le declară în XAML. Limbajul XAML sprijină schimbul de surse între diferite instrumente și roluri în procesul de dezvoltare, fără pierderi de informații, cum ar fi schimbul de surse XAML între Visual Studio și Microsoft Expression Blend.

Fișierele XAML sunt fișiere XML care au, în general, fișier XAML extensia de nume. Exemplul următor prezintă conținutul unui fișier de bază Windows Phone XAML.

```
<UserControl x:Class="MyWindowsPhone.Page"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" >
  <Grid Background="OldLace">
    </Grid>
</UserControl>
```

### 4.2. DEPENDENȚA PROPIETĂȚILOR

O proprietate dependență este o proprietate specializată, care are o valoare care se bazează pe valoarea altor factori de producție. O proprietate dependență contribuie la susținerea următoarelor caracteristici:

- legături de date
- stiluri
- Animații
- comportamentul PropertyChanged
- Valoarea implicită prin metadata
- Utilitate sistem general de proprietate și de metadata căutare

### 4.3. EVENIMENTE PENTRU WINDOWS PHONE OS

Un eveniment este un mesaj trimis de un obiect pentru a semnala apariția unei acțiuni. Acțiunea ar putea fi cauzată de interacțiunea cu utilizatorul, cum ar fi atingerea ecranului, sau ar putea fi declanșată de logica internă a unei clase. Obiectul care creează evenimentul se numește expeditor de eveniment. Obiectul care surprinde evenimentul și care răspunde este numit receptor de eveniment. Practic scopul de evenimentelor este de a comunica informații rapid de la un obiect la momentul execuției, și potențial de a furniza aceste informații la alte obiecte din aplicație.

În general vorbind , evenimente Windows Phone sunt evenimente CLR prin urmare, sunt evenimente care pot fi controlate în cod. Dacă se cunoaște modul de lucru de bază cu evenimente CLR este facila operarea cu unele dintre conceptele implicate . Dar nu trebuie neapărat să știe că de mult despre modelul eveniment CLR , în scopul de a îndeplini anumite sarcini de bază , cum ar fi atașarea handler-elor .

Deoarece UI pentru Windows app tipic bazate pe telefon este definit în markup ( XAML ) , unele dintre principiile de conectare de evenimente UI de la elemente de marcaj la o entitate cod de execuție sunt similare cu alte tehnologii web , cum ar fi ASP.NET , sau de lucru cu un DOM HTML. În Windows Phone codul care asigură logica de execuție pentru o UI definit - XAML este adesea menționată ca cod - spatele sau cod - spatele dosar . În Visual Studio, această relație este reprezentată grafic , cu dosarul de cod - spatele fiind un fișier dependent și imbricate față de pagina XAML-ul la care se referă .

În plus față de evenimentele CLR care acționează în propriul model de obiect de execuție , Windows Phone are , de asemenea, cateva evenimente care pot invoca handlerle incarcate din scripturi la nivel de HTML cum ar fi OnError . Aceste evenimente sunt expuse și pot fi manipulate de către orice scenariu de lucru în HTML DOM . Ca un model de eveniment general pentru o aplicație , HTML scripting ar trebui să aibă în primul rând șansa de a face manipularea , și în cazul în care utilizatorul se ocupă de evenimente într-un HTML , aceste evenimente nu ar trebui trimise catre API .

---

## 4.4. DATA BINDING

Legăturile de date cu caracter obligatoriu oferă o modalitate simplă de aplicații pentru Windows Phone pentru a afișa și de a interacționa cu datele. Modul în care datele sunt afișate este separat de gestionarea datelor. O conexiune, sau cu caracter obligatoriu, între UI și un obiect de date permite datelor să circule între cele două. Când o conexiune este stabilită și se modifică datele, elementele UI care sunt legate la datele pot reflecta automat modificările. În mod similar, modificările efectuate de către utilizator într-un element de UI pot fi reflectate în obiectul de date. De exemplu, dacă utilizatorul editează valoarea într-un TextBox, valoarea de bază de date este actualizată în mod automat pentru a reflecta această schimbare.

Unele scenarii comune obligatorii includ obligatoriu un ListBox la o listă de titluri, TextBox un formular de intrare de la un obiect de date client, sau o imagine pentru a fotografia utilizatorului curent.

## 4.5. MODEL-VIEW-VIEWMODEL

Model-View-ViewModel (MVVM) modelul de proiectare este o modalitate foarte bună de a separa datele de la interfața de utilizator într-o aplicație. Schema de mai jos demonstrează o aplicație MVVM simplă care utilizează XAML pentru a conecta view-ul la model. Aplicația în acest eșantion este un "tracker joc", în care utilizatorul poate urmări realizările lor într-un joc video. Această probă folosește, de asemenea, un convertor de date personalizat, menține pagina de stat atunci când navigați la și de la app, și salvează datele aplicației izolat.

MVVM facilitează o separare clară a dezvoltării de interfața grafică ( fie ca limbaj de marcare sau cod GUI ), de la dezvoltarea de logica. View-ul Modelul de MVVM este un convertor de valoare ceea ce înseamnă că modelul de vizualizare este responsabil pentru expunerea obiectelor de date de la modelul în așa fel încât aceste obiecte sunt ușor de administrat și consumat . În acest sens , modelul de vedere este mai mult decât modelul de vedere , și se ocupă de cele mai multe , dacă nu toate de logica de afișare de vedere a ( deși delimitarea între ceea ce funcții sunt gestionate prin care strat este un subiect de discuție [ 5 ] și de explorare în curs de desfășurare ) . Vedere Modelul poate pune în aplicare , de asemenea, un model de mediator organizarea acces la logica backend în jurul set de cazuri de utilizare susținute de vedere .

## WPF LOB Application Layers – M-V-VM

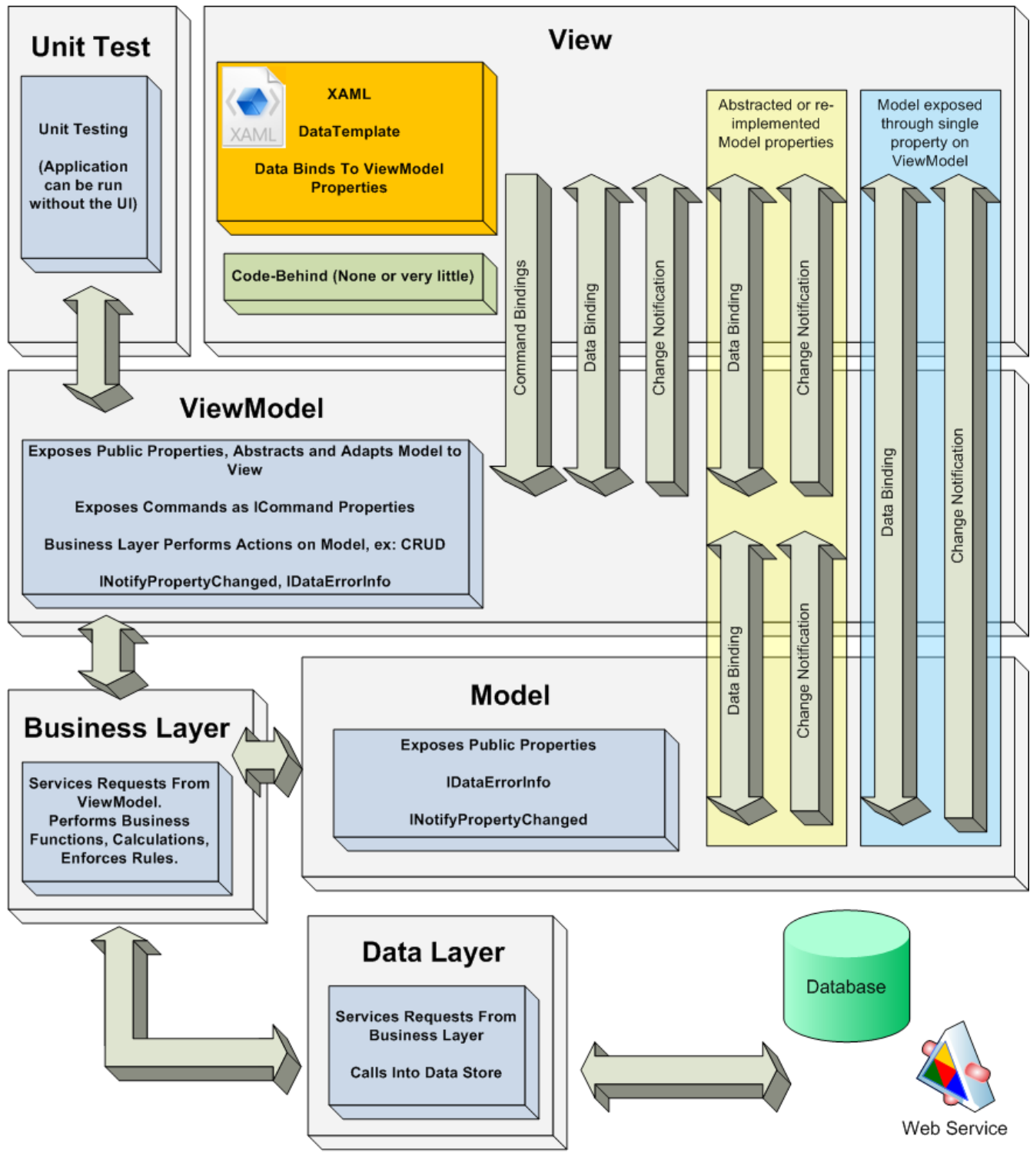


Figura. Exemplificarea modelului MVVM

## Capitolul 5. CONCLUZII

Windows Phone are un core comun cu Windows 8 care faciliteaza folosirea resurselor si integrarea hardware software pe terminalele mobile.

Chiar dacă distinctiv UX nu s-a modificat semnificativ între Windows Phone 7 și Windows Phone 8, au existat schimbări dramatice la suprafață. Pentru prima dată, Windows Phone este construit pe aceeași tehnologie ca și omologul său PC. În această secțiune, voi descrie cele două părți esențiale ale acestei schimbări cu impact pentru dezvoltatori: părțile de bază pentru Windows partajate, precum și adoptarea de Windows Runtime.

## BIBLIOGRAFIE

Windows Phone Recipes. Second Edition. Fabio Claudio Ferracchiati, Emanuele Garofalo. Ed. Apress

Essential Windows Phone. Shawn Wildermuth. Ed. Addison-Wesley

[http://chimera.labs.oreilly.com/books/123400000153/ch01.html#\\_light\\_and\\_simple](http://chimera.labs.oreilly.com/books/123400000153/ch01.html#_light_and_simple)

<http://j4ni.com/blog/?p=107>

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj26948\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj26948(v=vs.105).aspx)

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh20996\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh20996(v=vs.105).aspx)

<http://developer.windowsphone.com/en-us/develop>