

Travail de Bachelor 2012

## Filière Informatique de gestion

### EVOLUTION DE L'IHM D'UNE PLATEFORME REALISEE EN SILVERLIGHT



Etudiante : Sonia Ragosta Foti

Professeur : Jean-Pierre Rey

[www.hevs.ch](http://www.hevs.ch)

## Remerciements

Je tiens chaleureusement à remercier mes amis de la « classe des temps partiels » qui m'ont aidée et soutenue durant ces 4 années.

Je tiens également à remercier mon mari et mes filles pour m'avoir supportée pendant les périodes d'examens, M. Simon Crettaz pour la relecture de ce document, mais surtout M. Jean-Pierre Rey, mon professeur responsable, pour sa disponibilité et pour m'avoir encadrée dans mon travail.

## Résumé

Dans le cadre du projet de recherche MEU (Management Energétique Urbain) en collaboration avec le CREM, l'EPFL et de l'HES-SO, une plateforme à l'usage des villes a été réalisée en Silverlight et C#.

Cette plateforme représente un outil d'aide à la décision. Elle va aider les utilisateurs, grâce à différents outils de calculs, à élaborer des stratégies énergétiques des zones urbaines existantes ou futures.

Lors du démarrage du travail de bachelor, au mois de mars, le futur de Silverlight n'était pas assuré. Des rumeurs couraient sur le web. Microsoft allait peut-être cesser son développement. La mort de Silverlight impliquerait des modifications du projet MEU.

Le but du travail de bachelor « Evolution de l'IHM d'une plateforme réalisée en Silverlight » était d'analyser les éléments existants et comprendre la direction qu'allait prendre le projet pour faire face aux changements.

Pour porter à bien cette analyse, les recherches se sont déroulées de façon itérative et une séance était fixée à la fin de chaque Sprint. Les séances servaient à faire le bilan des résultats obtenus et à préciser le ou les buts de la prochaine étape. En effet le planning initial a subi des modifications au cours des semaines.

Voilà les différentes itérations:

- Compréhension du fonctionnement de Silverlight
- Compréhension de l'architecture de développement de l'application existante
  - Modèle MVVM (Model – View – ViewModel)
- Positionnement de Microsoft face à Silverlight et HTML5
- Metro Style Apps
- Applications (Windows 8, HTML5, OData)

Les recherches nous portent à ces conclusions.

Microsoft n'envisage pas d'abandonner le développement de Silverlight. La version 5 a été présentée en fin d'année 2011 et selon sa politique de support, Microsoft assurera le support de Silverlight jusqu'au 12 octobre 2021<sup>1</sup>.

La Société Microsoft a cependant décidé de prendre une tout autre direction : elle a décidé de s'attaquer au monde du mobile.

La présentation de Windows 8 montre cette évolution.

Les développeurs peuvent choisir soit les applications Desktop, le développement traditionnel avec Silverlight et toutes les fonctionnalités existantes et améliorées, soit les applications Metro Style, le développement mobile avec la possibilité de choisir le langage de programmation y compris HTML5 et JavaScript.

Le projet actuel MEU aura une évolution technique due aux nouveautés de Silverlight. Les développeurs n'ont pas la nécessité de changer le langage de programmation. Ils peuvent l'améliorer en utilisant les nouvelles fonctionnalités de Silverlight et éventuellement d'envisager de porter le projet MEU sur tablette en créant une application Windows Metro Style.

**Mots - clés :** HTML5, JavaScript, Metro Style, MVVM, OData, Silverlight, Visual Studio, Windows 8, WinRT

---

<sup>1</sup> <http://support.microsoft.com/lifecycle/?c2=12905>

# Table des matières

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
<b>2</b>	<b>COMPRÉHENSION DU FONCTIONNEMENT DE SILVERLIGHT .....</b>	<b>3</b>
2.1	RIA .....	3
2.2	ARCHITECTURE RIA - SILVERLIGHT .....	5
2.3	SILVERLIGHT .....	6
2.3.1	<i>Silverlight 1</i> .....	6
2.3.2	<i>Silverlight 2</i> .....	6
2.3.3	<i>Silverlight 3</i> .....	7
2.3.4	<i>Silverlight 4</i> .....	7
2.3.5	<i>Silverlight 5</i> .....	8
2.4	APPLICATION SILVERLIGHT .....	8
2.4.1	<i>Les langages de programmation</i> .....	8
2.4.2	<i>Les applications</i> .....	9
<b>3</b>	<b>COMPRÉHENSION DE L'ARCHITECTURE DE DÉVELOPPEMENT DE L'APPLICATION EXISTANTE... ..</b>	<b>13</b>
3.1	ARCHITECTURE SILVERLIGHT .....	13
3.2	ARCHITECTURE MVVM .....	15
3.2.1	<i>View</i> .....	17
3.2.2	<i>ViewModel</i> .....	18
3.2.3	<i>Model</i> .....	18
3.2.4	<i>Fonctionnement</i> .....	19
3.3	DATA BINDING .....	20
3.4	COMMANDS .....	22
3.5	MVVM FRAMEWORKS .....	24
3.5.1	<i>PRISM</i> .....	24
3.5.2	<i>MVVM Light Toolkit</i> .....	25
3.5.3	<i>Simple MVVM Toolkit</i> .....	26
3.5.4	<i>Caliburn.Micro</i> .....	26
<b>4</b>	<b>POSITIONNEMENT DE MICROSOFT FACE À SILVERLIGHT ET HTML5 .....</b>	<b>27</b>
4.1	L'HISTOIRE DE HTML5 .....	27
4.2	LA COMPOSITION DE HTML5 .....	29
4.3	LES NOUVELLES FONCTIONNALITÉS .....	30
4.4	LE DÉVELOPPEMENT D'APPLICATIONS MOBILES .....	34
4.5	L'ARCHITECTURE DE HTML .....	35

4.6	POSITIONNEMENT DE MICROSOFT .....	37
4.6.1	Plateforme Windows 8 et les outils .....	38
4.6.2	Et Silverlight ? .....	39
<b>5</b>	<b>METRO STYLE APPS .....</b>	<b>40</b>
5.1	L'HISTOIRE DE METRO STYLE .....	40
5.2	METRO STYLE APPS .....	42
5.2.1	WinRT APIs .....	44
5.2.2	Applications .....	45
<b>6</b>	<b>APPLICATIONS .....</b>	<b>46</b>
6.1	ENVIRONNEMENT DE TRAVAIL .....	46
6.1.1	Windows 8 .....	46
6.1.2	Visual Studio 2011 .....	47
6.1.3	Visual Studio Express 2012 RC .....	48
6.1.4	Visual Studio Professional 2012 RC .....	51
6.1.5	Windows Phone .....	53
6.2	SILVERLIGHT / JAVASCRIPT : LES DIFFÉRENCES .....	54
6.2.1	Interface graphique .....	54
6.2.2	Gestion de l'évènement .....	55
6.2.3	Affichage .....	56
6.3	PEUT-ON PASSER FACILEMENT LES APPLICATIONS D'UN LANGAGE À L'AUTRE? .....	56
6.4	ODATA .....	59
<b>7</b>	<b>PLANIFICATION DE MICROSOFT .....</b>	<b>61</b>
7.1	WINDOWS 8 GUIDANCE .....	61
7.2	MOBILE WEB GUIDANCE .....	62
<b>8</b>	<b>CONCLUSION .....</b>	<b>64</b>
<b>9</b>	<b>ATTESTATION .....</b>	<b>65</b>
<b>10</b>	<b>RÉFÉRENCES .....</b>	<b>66</b>
<b>11</b>	<b>ANNEXES .....</b>	<b>I</b>
11.1	SUPPORT DE MICROSOFT .....	I
11.2	APPLICATIONS .....	II
11.2.1	Pivot Viewer - Silverlight .....	II
11.2.2	MVVM .....	VI
11.2.3	LightSwitch .....	XII
11.3	CAHIER DES CHARGES .....	XVIII
11.4	PLANNING .....	XXVIII

11.5	PV .....	XXXIX
11.5.1	<i>PV de la séance du 16.03.12 – N°01</i> .....	XXXIX
11.5.2	<i>PV de la séance du 22.03.12 – N°02</i> .....	XLI
11.5.3	<i>PV de la séance du 19.04.12 – N°03</i> .....	XLIII
11.5.4	<i>PV de la séance du 02.05.12 – N°04</i> .....	XLV
11.5.5	<i>PV de la séance du 16.05.12 – N°05</i> .....	XLVII
11.5.6	<i>PV de la séance du 30.05.12 – N°06</i> .....	XLIX
11.5.7	<i>PV de la séance du 13.06.12 – N°07</i> .....	LII
11.5.8	<i>PV de la séance du 27.06.12 – N°08</i> .....	LIII
11.6	RAPPORTS HEBDOMADAIRES.....	LV
11.6.1	<i>Semaines du 9 au 22 avril 2012</i> .....	LV
11.6.2	<i>Semaines du 23 avril au 6 mai 2012</i> .....	LVII
11.6.3	<i>Semaines du 7 mai au 27 mai 2012</i> .....	LIX
11.6.4	<i>Semaines du 28 mai au 10 juin 2012</i> .....	LXI
11.6.5	<i>Semaine du 11 juin au 17 juin 2012</i> .....	LXII
11.6.6	<i>Semaines du 18 juin au 8 juillet 2012</i> .....	LXIII
11.6.7	<i>Semaines du 9 juillet au 22 juillet 2012</i> .....	LXV

## Table des illustrations

FIGURE 1 : MEU - PLATEFORME.....	1
FIGURE 2 : MEU - ARCHITECTURE .....	2
FIGURE 3 : WORLD WIDE WEB .....	3
FIGURE 4 : NEXT - CERN .....	4
FIGURE 5 : RIA.....	4
FIGURE 6 : ARCHITECTURE RIA.....	5
FIGURE 7 : HISTORIQUE .....	6
FIGURE 8 : DEEP ZOOM .....	7
FIGURE 9 : BLUREFFECT - XAML .....	8
FIGURE 10 : BLUREFFECT - C# .....	9
FIGURE 11 : LECTURE D'UN MÉDIA - SÉLECTION.....	9
FIGURE 12 : LECTURE D'UN MÉDIA - SÉLECTION.....	10
FIGURE 13 : SAUVEGARDER UN MÉDIA.....	10
FIGURE 14 : OUVERTURE D'UN MÉDIA .....	11
FIGURE 15 : DRAG & DROP 1 .....	11
FIGURE 16 : DRAG & DROP 2 .....	12
FIGURE 17 : GALERIE D'IMAGES.....	12
FIGURE 18 : ARCHITECTURE MULTICOUCHE.....	13
FIGURE 19 : MVC – DIAGRAMME DE SÉQUENCE .....	14
FIGURE 20 : MVVM.....	15
FIGURE 21 : SÉPARATION : DESIGNER - DEVELOPER.....	15
FIGURE 22 : MVVM - DESCRIPTION .....	17
FIGURE 23 : MVVM - VIEW.....	17
FIGURE 24 : MVVM - VIEWMODEL .....	18
FIGURE 25 : MVVM - MODEL.....	18
FIGURE 26 : MVVM – DIAGRAMME.....	19
FIGURE 27 : DATA BINDING.....	20
FIGURE 28 : DATA BINDING - ONEWAY .....	20
FIGURE 29 : DATA BINDING - TWOWAY .....	21
FIGURE 30 : DATA BINDING - MODE .....	21
FIGURE 31 : DATA BINDING - INOTIFYPROPERTYCHANGED.....	22
FIGURE 32 : DATA BINDING - PROPERTYCHANGED.....	22
FIGURE 33 : COMMANDS - EXEMPLE .....	23
FIGURE 34 : COMMANDS - SCHÉMA.....	23
FIGURE 35 : PRISM 4.1 .....	24
FIGURE 36 : VIEWMODELBASE .....	25
FIGURE 37 : MVVM LIGHT TOOLKIT.....	25
FIGURE 38 : SIMPLE MVVM LIGHT TOOLKIT .....	26
FIGURE 39 : CALIBURN.MICRO.....	26
FIGURE 40 : TIM BERNERS-LEE.....	27
FIGURE 41 : LOGO HTML5 .....	28
FIGURE 42 : COMPOSITION DE HTML5 .....	29
FIGURE 43 : SIMPLIFICATION DE CODE - AVANT .....	30
FIGURE 44 : SIMPLIFICATION DE CODE - APRÈS .....	30
FIGURE 45 : NOUVELLES BALISES .....	31
FIGURE 46 : NOUVELLE BALISE - <AUDIO> .....	31
FIGURE 47 : NOUVELLE BALISE - <VIDEO> .....	32
FIGURE 48 : DRAG AND DROP.....	32
FIGURE 49 : APPLICATIONS OFFLINES.....	33
FIGURE 50 : GÉOLOCALISATION.....	34
FIGURE 51 : SMARTPHONE ET HTML5 .....	34
FIGURE 52 : PHONEGAP - EXEMPLE .....	35
FIGURE 53 : HTML – MVVM – LA VIEW.....	36
FIGURE 54 : HTML – MVVM – LE VIEWMODEL.....	36

FIGURE 55 : HTML – MVVM – RÉSULTAT .....	36
FIGURE 56 : PART DU MARCHÉ EN EUROPE (2011).....	37
FIGURE 57 : WINDOWS 8 - PLATEFORME .....	39
FIGURE 58 : METRO DESIGN LANGUAGE .....	40
FIGURE 59 : WINDOWS 8 – START PAGE .....	41
FIGURE 60 : WINDOWS 8 - SCREEN .....	42
FIGURE 61 : WINDOWS METRO STYLE – ORIENTATION PREFERENCES .....	42
FIGURE 62 : WINDOWS STORE.....	43
FIGURE 63 : WINDOWS 8 - PLATEFORME .....	44
FIGURE 64 : WINRT .....	44
FIGURE 65 : NAVIGATION – WINDOWS 8.....	46
FIGURE 66 : VISUAL STUDIO 2011 - MENU.....	47
FIGURE 67 : VISUAL STUDIO 2011 – MESSAGE ERREUR.....	47
FIGURE 68 : VISUAL STUDIO EXPRESS - MENU .....	48
FIGURE 69 : VISUAL STUDIO EXPRESS - MESSAGE .....	48
FIGURE 70 : VISUAL STUDIO 2011 – PAGE DEFAULT.JS.....	49
FIGURE 71 : VISUAL STUDIO EXPRESS – PAGE DEFAULT.JS .....	49
FIGURE 72 : VISUAL STUDIO 2011 – PAGE DEFAULT.JS - 2 .....	50
FIGURE 73 : VISUAL STUDIO EXPRESS – PAGE DEFAULT.JS - 2.....	50
FIGURE 74 : FICHIER JS .....	50
FIGURE 75 : LIGHTSWITCH – ARCHITECTURE.....	51
FIGURE 76 : LIGHTSWITCH – START SCREEN.....	52
FIGURE 77 : LIGHTSWITCH – ATTACH DATA SOURCE .....	52
FIGURE 78 : LIGHTSWITCH – ODATA_CONNECTION.....	52
FIGURE 79 : LIGHTSWITCH – UI .....	53
FIGURE 80 : VISUAL STUDIO EXPRESS FOR WINDOWS PHONE .....	53
FIGURE 81 : SILVERLIGHT – CODE XAML .....	54
FIGURE 82 : JAVASCRIPT – HTML5 .....	55
FIGURE 83 : SILVERLIGHT – GESTION DE L'ÉVÈNEMENT .....	55
FIGURE 84 : JAVASCRIPT – GESTION DE L'ÉVÈNEMENT .....	55
FIGURE 85 : SILVERLIGHT - SCREEN .....	56
FIGURE 86 : JAVASCRIPT - SCREEN .....	56
FIGURE 87 : MODIFICATION DES APIS .....	57
FIGURE 88 : ARCHITECTURES .....	58
FIGURE 89 : MULTIPLATEFORMES .....	58
FIGURE 90 : PORTABLE CLASS LIBRARY – CODE COMMUN .....	59
FIGURE 91 : ODATA - SDK.....	60
FIGURE 92 : ODATA - MULTIPLATEFORME .....	60
FIGURE 92 : ROADMAP .....	61
FIGURE 93 : VENTE TABLETTES.....	62
FIGURE 94 : VENTE SMARTPHONE.....	63
FIGURE 95 : FUTUR DU PROJET MEU .....	64



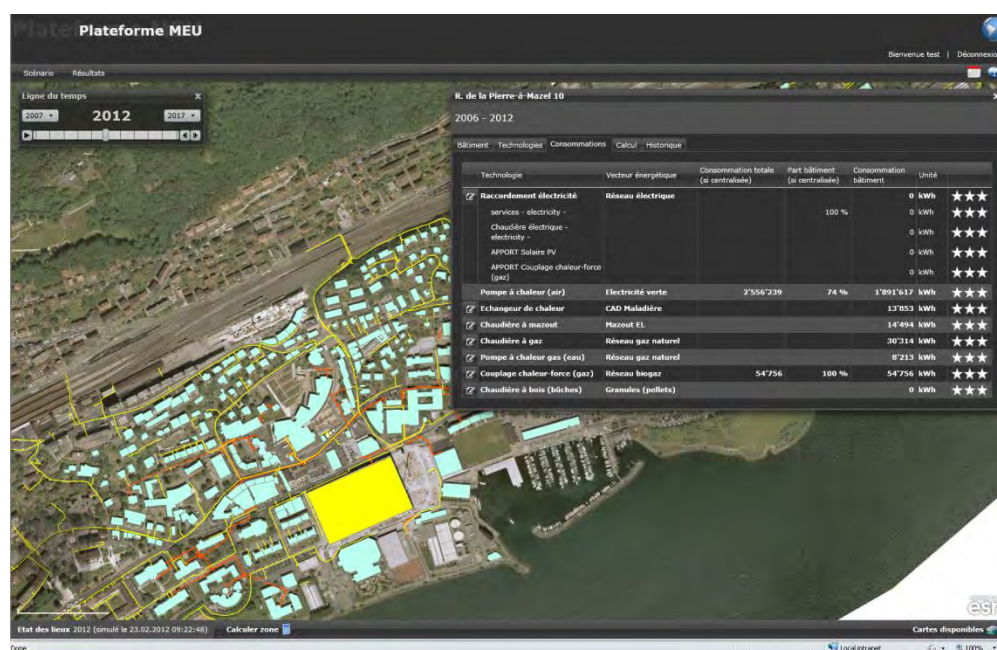
# 1 Introduction

Dans le cadre du projet de recherche MEU (Management Energétique Urbain) en collaboration avec le CREM, l'EPFL et de l'HES-SO, une plateforme à l'usage des villes a été réalisée en Silverlight et C#.

Cette plateforme représente un outil d'aide à la décision. Elle va aider les utilisateurs, grâce à différents outils de calculs, à élaborer des stratégies énergétiques des zones urbaines existantes ou futures.

L'utilisateur a devant lui une carte géographique et GIS (Geographic Information System). GIS est un système d'information géographique qui permet de visualiser, d'analyser, d'interpréter des données en temps réel. Toutes les fonctionnalités de GIS sont présentes dans l'environnement de ArcGIS.

Figure 1 : MEU - Plateforme



Source : Document reçu – mars 2012

La plateforme MEU permet de visualiser les situations les plus complexes et l'évolution au cours des années. L'utilisateur peut étudier différents scénarios et proposer rapidement la solution énergétique la plus avantageuse.

Cette plateforme fait appel à divers outils<sup>2</sup> tels que :

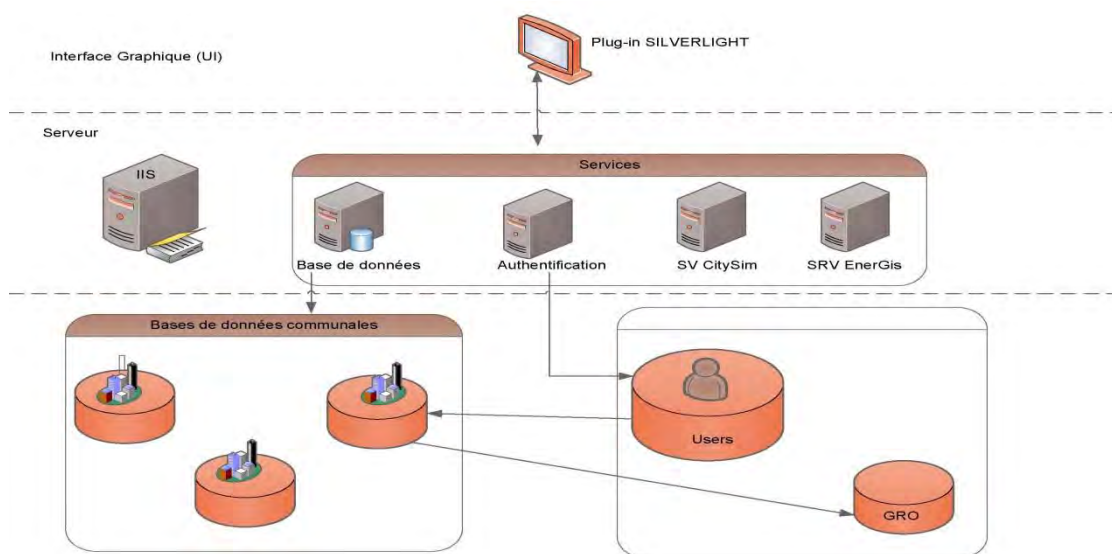
- EnerGis : développé par l'EPFL, LENI (Laboratoire d'Energétique Industrielle), il s'agit d'un outil qui permet de calculer le besoin énergétique d'une ville.

<sup>2</sup> Descriptif de projet fonds The Ark – 01.03.2012

- CitySim : développé par l'EPFL, LESO-PB (Laboratoire d'Energie Solaire et Physique du Bâtiment), il s'agit d'un outil pour la modélisation de bâtiments et de leurs consommations (eau, énergie, température, ...).
- GEMIS : développé par l'Université de Kassel, il s'agit d'un outil permettant d'analyser les impacts environnementaux et financiers et de fournir un bilan énergétique.

La plateforme accède à ces outils à l'aide de web services.

**Figure 2 : MEU - Architecture**



Source : Présentation de M. Alain Duc du 02.05.12

L'application a été développée avec Silverlight.

Silverlight est un plug-in de Microsoft développé en 2007. Il permet la création d'interfaces graphiques plus riches (RIA) et une plus grande interactivité de l'utilisateur avec les interfaces graphiques.

Cependant le futur de Silverlight est en danger. Des rumeurs envahissent le web : le plug-in Silverlight est voué à disparaître.

Le but de ce rapport est d'étudier l'évolution de l'IHM de cette application créée en Silverlight.

## 2 Compréhension du fonctionnement de Silverlight

---

L'utilisation de Silverlight dans le projet MEU augmente l'interactivité des utilisateurs avec les cartes proposées.

Silverlight est une technologie développée par Microsoft. Elle permet de créer des RIA (Rich Internet Application).

Avant de comprendre le fonctionnement de cette nouvelle technologie, il est important et nécessaire de comprendre ce que sont les RIA.

Les RIA sont des applications web évoluées offrant aux utilisateurs de nombreuses fonctionnalités. Il est difficile de donner une définition exacte à ce concept, car il est apparu petit à petit sur le web. Mais qu'en est-il exactement ?

### 2.1 RIA

Le premier site web a été créé par Tim Berners-Lee en 1992. Il ne contenait aucune image, son interface était très pauvre. La présence de liens permet aux utilisateurs de naviguer facilement vers d'autres pages. Il est étonnant de découvrir que ce site est encore disponible et fonctionnel.

Figure 3 : World Wide Web



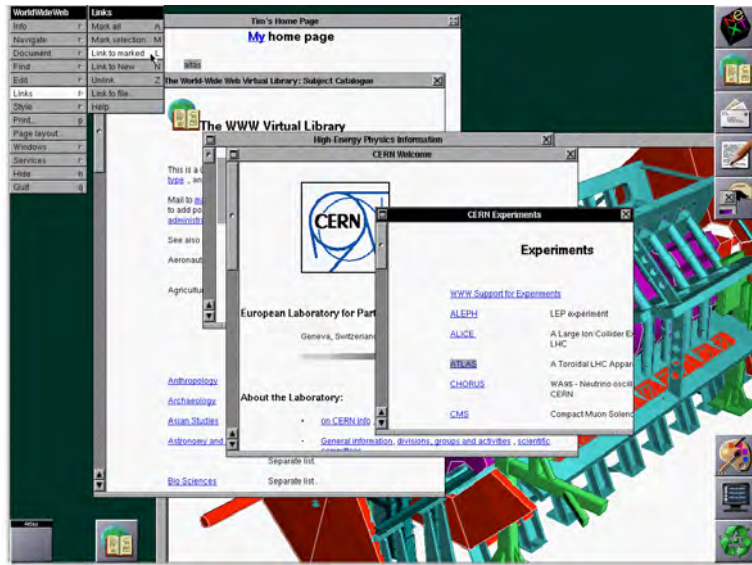
Source : <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html>

Mais déjà en 1993, les premières interfaces plus élaborées sont apparues.

L'inconvénient était qu'il n'était disponible que sur un système d'exploitation neXTStep du CERN. L'interface ressemble à ce que nous connaissons actuellement.

Pendant cette période, les développeurs portaient leur attention sur le fond et négligeaient la forme. Tous les sites se ressemblaient.

### Figure 4 : neXT - CERN



Source : <http://info.cern.ch/NextBrowser1.html>

L'évolution a suivi son cours. Le terme de RIA est apparu pour la première fois en 2002 dans un white paper de macromedia, créateur de Flash.

Il est intéressant de souligner que cette évolution a été rendue possible par l'analyse de l'existant. En effet, les développeurs analysent ce qui existe sur la toile et ne retiennent que les meilleurs concepts en les améliorant en suivant toujours le même fil d'Ariane, c'est-à-dire de rendre la navigation plus simple et plus intuitive.

### Figure 5 : RIA



Source : <http://blog.cozic.fr/Que-sont-les-applications-riches-RIA-Partie-1-Definition-et-usages-195>

RIA, **Rich Internet Application**, est l'aboutissement de ce développement. Son but est d'augmenter l'interactivité et la rapidité d'exécution avec le client grâce à l'interface utilisateur. En effet, la partie de l'application, qui élabore les données, est exécutée sur le poste Client, ce qui permet une réponse rapide, tandis que la plus grande partie des données et de l'application se trouvent sur le serveur.

Pour pouvoir créer des RIA, il existe différentes technologies à notre disposition.

Il est possible de citer:

- Flex
- Mozilla
- Microsoft
- Google
- et bien d'autres.

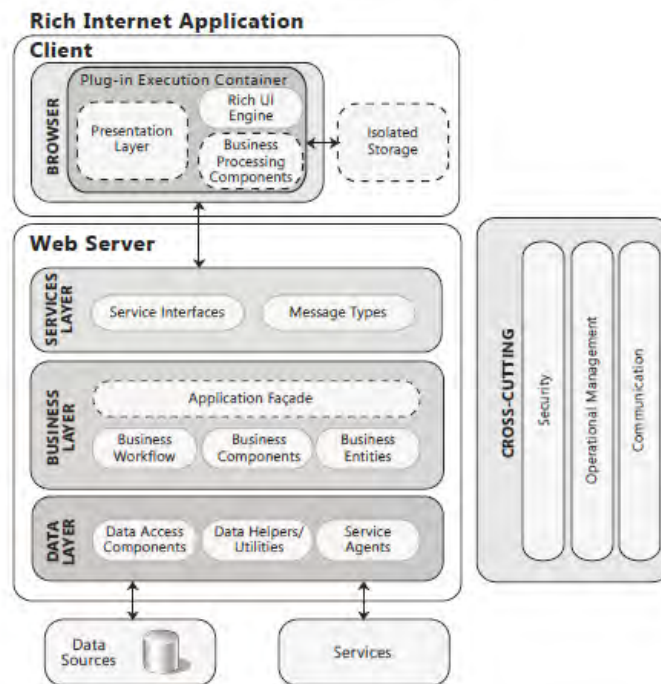
En ce qui concerne ce rapport, l'intérêt est porté exclusivement sur la technologie de Microsoft : Silverlight.

## 2.2 Architecture RIA - Silverlight

Une application RIA tourne dans un navigateur dans le bac à sable. Elle est structurée comme une application multicouche:

- Couche présentation
- Couche application
- Couche données

**Figure 6 : Architecture RIA**



Source : .NET Application Architecture 320 Guide, 2nd Edition



De plus, elle dépend d'un plug-in côté client, dans notre cas Silverlight. Celui-ci communique avec le Web Serveur qui génère le code et les données utilisées par le client.

## 2.3 Silverlight

En 2007, Microsoft, dans cette optique de développement, lance sur le marché Silverlight lors de la conférence « National Association of Broadcaster » à Las Vegas. Cette nouvelle technologie est multiplateformes et multi-navigateurs. Elle permet d'intégrer des éléments multimédia, des animations pour rendre les pages web beaucoup plus interactives.

Les applications Silverlight sont modulables. Une page HTML peut l'utiliser comme un simple contrôle ou il est possible de créer des applications entières en Silverlight.

Les principaux avantages sont, selon Olivier Dewit (2011, p. 14) :

- Réactivité et fluidité de l'interface utilisateur
- Déploiement rapide et facile

Quelle a été son évolution ?

Figure 7 : Historique

### 2.3.1 Silverlight 1

Cette première version intégrait :

- un moteur vectoriel
- un support de flux audio et vidéo

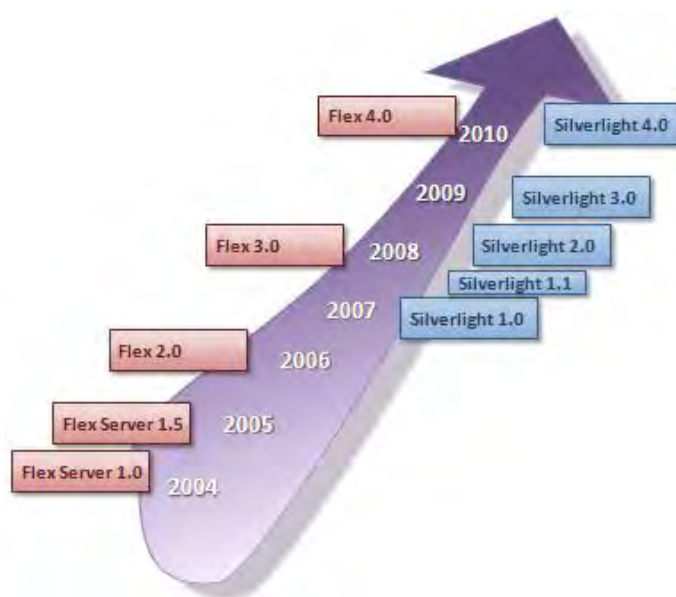
JavaScript était utilisé pour créer la partie logique de l'application.

### 2.3.2 Silverlight 2

La version précédente ne permettait pas de développer des applications de manière efficace.

Les nouveautés:

- Intégration d'une version de .NET Framework
  - Possibilité de coder en C#
- Le code-behind (logique de programmation) complétait les fichiers XAML de mise en page.
- De nombreuses fonctionnalités sont apparues grâce à l'existence de bibliothèques.



Source : <http://www.arcorama.fr/2010/02/flex-vs-silverlight-concepts-et.html>

- L'application tourne dans un bac à sable (sandbox), elle ne peut pas accéder aux fichiers de l'utilisateur. Elle est donc sécurisée.
- Un projet Silverlight, créé avec Visual Studio, peut être modifié par Expression Blend (logiciel graphique).
- Il inclut Deep Zoom qui permet de zoomer sur les images

**Figure 8 : Deep Zoom**



Source : <http://silverlike.net/magnifying-glass-effect>

### 2.3.3 Silverlight 3

La version 3 a été présentée le 18 mars 2009, lors de la conférence MIX09 à Las Vegas.

C'est une extension de la version 2 :

- Amélioration des capacités graphiques
- Support pour le graphisme 3D
- Animations
- Optimisation de moteurs de recherche
- Possibilité de lier 2 éléments (element-to-element binding / data binding)
- Les applications peuvent être utilisées en dehors du navigateur (out-of-browser-experiences).
- Les mises à jour se font automatiquement.

### 2.3.4 Silverlight 4

La version 4 hérite de la version 3 et l'enrichit. C'est pour cette raison qu'elle représente une plateforme idéale pour le développement et la distribution d'application moderne et interactive. Elle a été présentée en 2010.

Elle offre aux développeurs:

- Un ensemble complet de contrôle personnalisable
- Une amélioration du data binding
- Support complet de toutes les fonctionnalités par Visual Studio 2010
- Support du drag and drop
- Possibilité d'enregistrement en local d'audio ou vidéo
- Support pour le multitouch

### 2.3.5 Silverlight 5

A la grande surprise de tous les développeurs, Microsoft lance la version 5 en décembre 2011.

Les principales nouveautés sont :

- Amélioration du support multimédia
  - Enregistrement à des vitesses différentes
- Amélioration du formatage du texte
- Amélioration des présentations
  - Vitesse de chargement
- Amélioration du graphisme 3D, grâce à l'installation de XNA Studio
- Amélioration des tests unitaires

## 2.4 Application Silverlight

Microsoft met à disposition des utilisateurs des tutoriaux permettant de se familiariser avec le monde de Silverlight.

### 2.4.1 Les langages de programmation

Une application Silverlight est composée de deux parties :

- La programmation en langage supporté par .NET, dans notre application nous avons utilisé C#, appelé code-behind.
- La conception graphique basée sur le langage XAML, eXtensible Application Markup Language.

Le langage XAML décrit les objets .NET en XML. C'est pourquoi, tout ce qui est codé en XAML peut être codé en C#, mais l'inverse n'est pas possible.

Exemple de représentation de la création d'un effet de flou sur un élément :

En XAML

Figure 9 : BlurEffect - XAML

```
<Button x:Name="sauvegarder"
        Content="Sauvegarder"
        >Button.Effect<
        <BlurEffect/>
    </Button.Effect>
```



En C#

Figure 10 : BlurEffect - C#

```
BlurEffect effet = new BlurEffect();  
effet.Radius = 10;  
Application.Current.RootVisual.Effect = effet;
```

### 2.4.2 Les applications

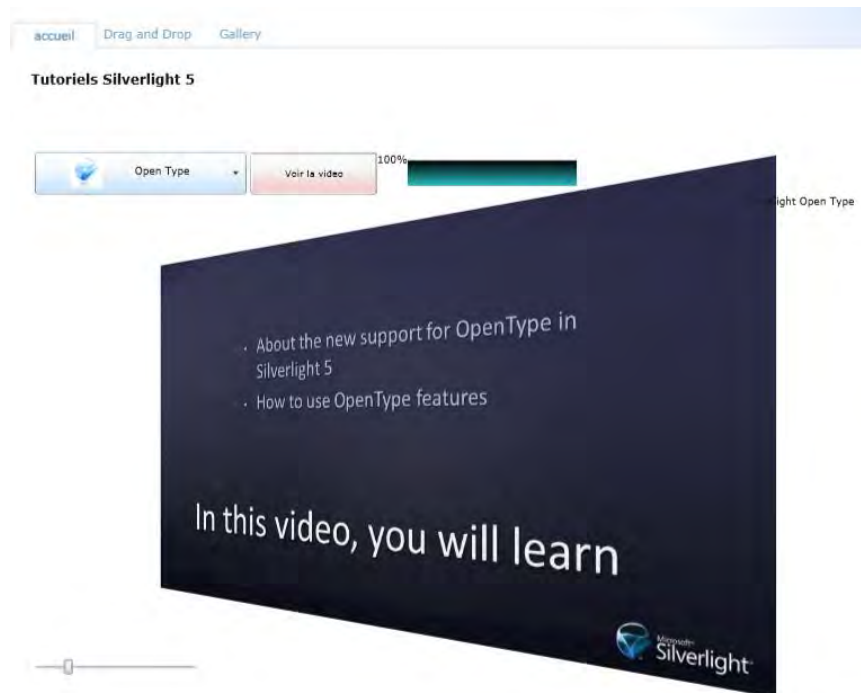
Pour tester, ces différents concepts, nous avons créé trois applications avec l'aide de tutoriaux :

- La gestion d'une bibliothèque de média
  - Lecture

Figure 11 : Lecture d'un média - Sélection

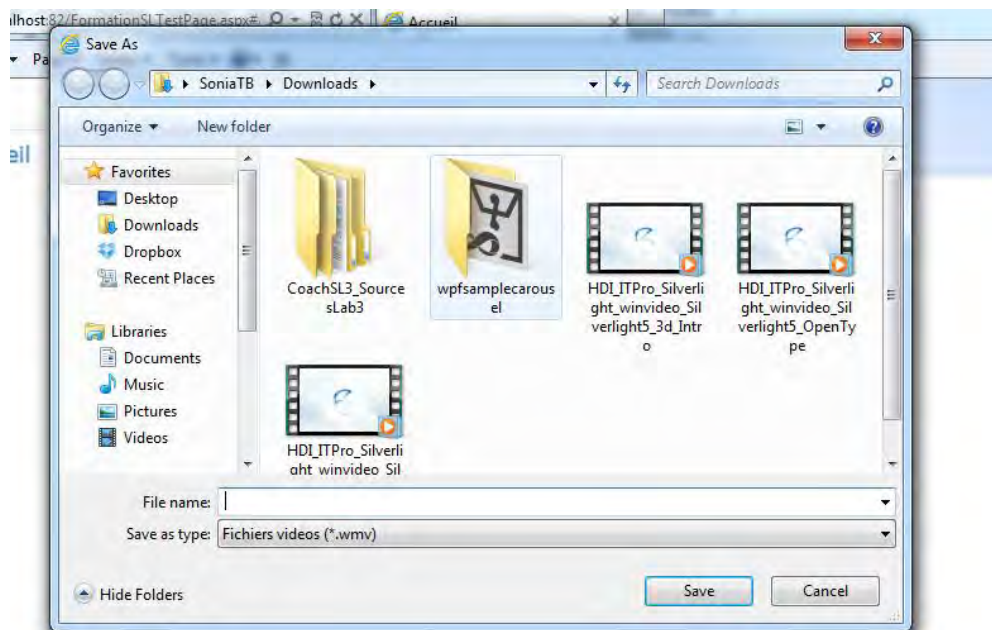


Figure 12 : Lecture d'un média - Sélection



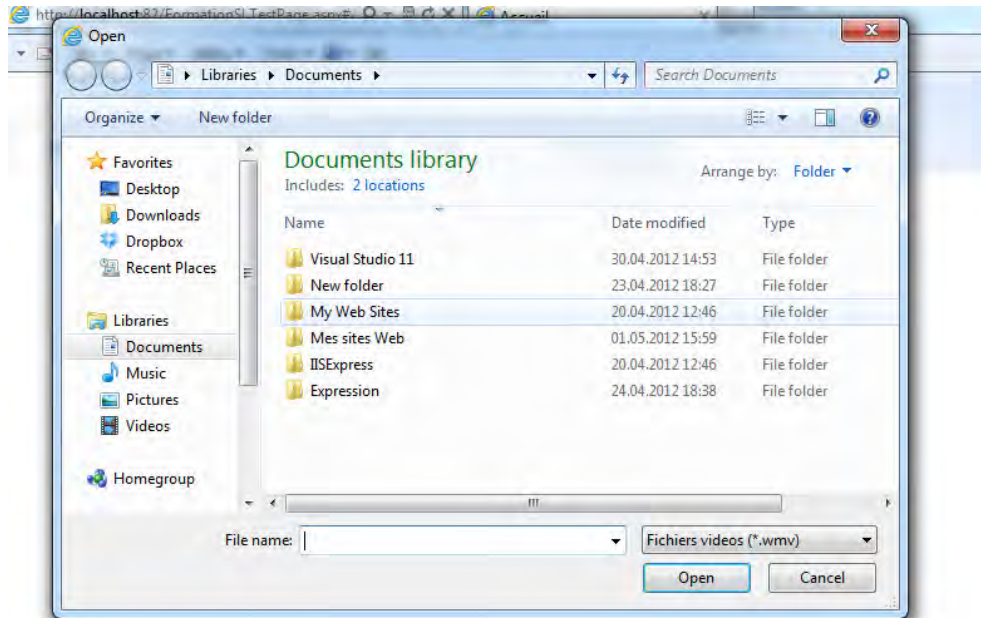
- Sauvegarder

Figure 13 : Sauvegarder un média



- Ouverture d'un média stocké en local

**Figure 14 : Ouverture d'un média**



- Le drag and drop de données.

**Figure 15 : Drag & Drop 1**

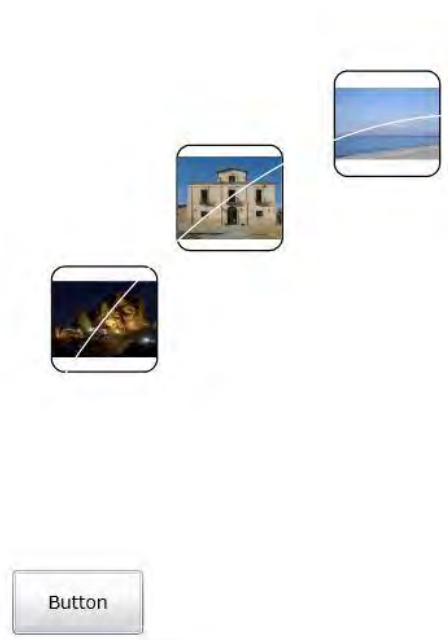


**Figure 16 : Drag & Drop 2**



- Une galerie d'images (créée avec Blend)

**Figure 17 : Galerie d'images**



NB : Les sources de ces applications sont jointes au présent rapport.

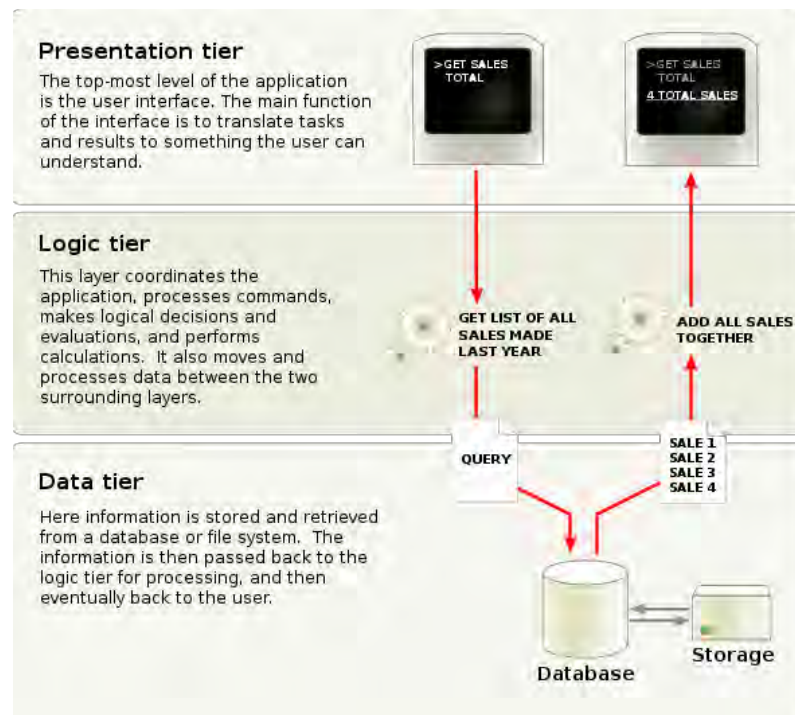
### 3 Compréhension de l'architecture de développement de l'application existante

---

#### 3.1 Architecture Silverlight

Comme nous l'avons vu, dans le paragraphe précédent, l'application Silverlight a une architecture multicouche.

Figure 18 : Architecture multicouche



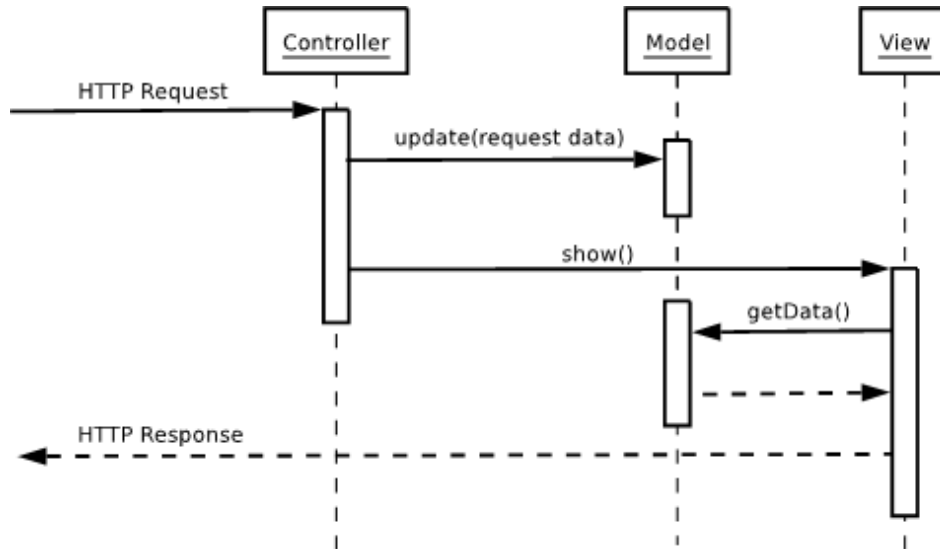
Source : [http://en.wikipedia.org/wiki/File:Overview\\_of\\_a\\_three-tier\\_application\\_vectorVersion.svg](http://en.wikipedia.org/wiki/File:Overview_of_a_three-tier_application_vectorVersion.svg)

Elle est donc divisée en trois couches :

- La couche présentation : elle représente l'interface utilisateur, la partie graphique.
- La couche application : elle représente l'ensemble des règles de gestion, la partie fonctionnelle de l'application.
- La couche des données : elle enregistre les informations dans une base de données et les remet à disposition des différentes couches selon les besoins.

Le modèle d'architecture le plus connu est certainement le MVC, Model – View – Controller.

Figure 19 : MVC – Diagramme de séquence



Source : <http://julien-pauli.developpez.com/tutoriels/php/mvc-controleur/>

- Model : représente la couche applicative. Elle contient toutes les méthodes qui permettent de modifier les données.
- Controller : représente la logique de l'application
- View : représente l'interface graphique de l'utilisateur

Dans une architecture MVC, le Controller est la partie la plus importante de l'architecture. En effet, il reçoit la requête et l'analyse. Ensuite, il accède au modèle en le mettant à jour et retourne à la vue les informations demandées.

L'application Silverlight utilise un tout autre modèle : le MVVM, Model – View – ViewModel.

Etudions son fonctionnement.

## 3.2 Architecture MVVM

Le modèle d'architecture MVVM a été présenté le 8 octobre 2005 par John Gossman sur son blog dans un article intitulé « Tales from the Smart Client »<sup>3</sup>.

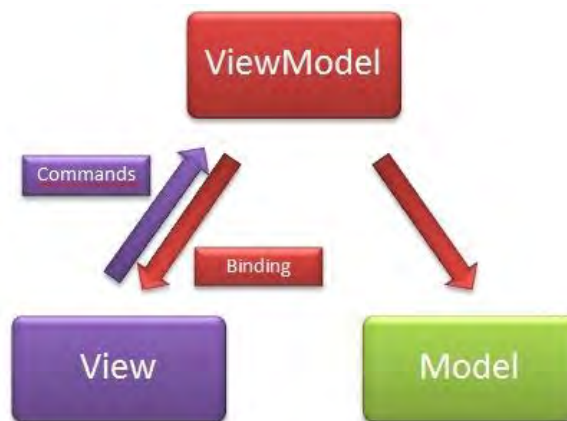
John Gossman est un architecte Microsoft de Silverlight.

Dans son article, il explique que le modèle MVVM est une variation du modèle MVC.

La conception de RIA (Rich Internet Applications) demande la séparation et la collaboration entre designer et développeur.

Ce nouveau modèle permet effectivement de séparer la conception de l'interface graphique (View) du code gérant l'application (ViewModel). Ces deux parties sont liées entre elles grâce au Data Binding et aux Commands que nous expliquerons dans ce chapitre.

Figure 20 : MVVM



Source : <http://julien-pauli.developpez.com/tutoriels/php/mvc-controleur/>

Cette séparation offre plusieurs avantages :

- Elle permet au développeur et au designer de travailler en même temps en développant chacun sa partie et pas forcément au même endroit.

Figure 21 : Séparation : Designer - Developer



Source : Alessandro Andreosè – Separazione dei ruoli tra Designer e Developer

<sup>3</sup> <http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx>

- Elle facilite

- Les tests

Chaque bloc pourra être testé de manière indépendante.

**View :**

Le designer peut contrôler visuellement que sa partie réponde bien au besoin du client. Il peut modifier l'interface sans créer de bogues majeurs dans l'application. S'il oublie de mettre un binding, l'application fonctionnera, mais l'élément n'aura pas le comportement souhaité. L'erreur est visible et facilement corrigeable.

**ViewModel et Model**

Grâce à la séparation des couches, la mise en place de tests unitaires est également facilitée.

Cette mise en place est possible, même si la vue n'existe pas encore.

Le développeur va invoquer les commandes et contrôler que le résultat obtenu corresponde bien au résultat recherché.

Les tests unitaires vont nous permettre de valider unitairement toutes les fonctionnalités de l'application.

En résumé, dans la couche ViewModel, les développeurs pourront tester les états des objets, les opérations et dans la couche Model, ils pourront tester les règles métiers.

Si nous prenons la peine de tester nos applications, nous pourrons livrer au client un produit de qualité.

- L'évolution de l'application

L'application pourra évoluer sans devoir partir de zéro. Il pourrait y avoir différentes interfaces utilisateurs utilisant les mêmes données.

- La maintenabilité

Chaque partie peut être modifiée ou remplacée sans causer de problème au reste de l'application.

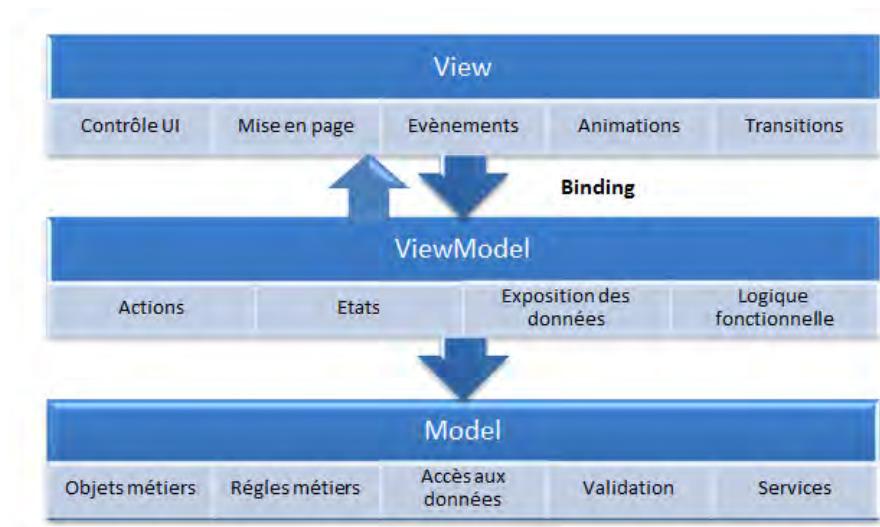
- La réutilisabilité du code

En séparant le code de l'interface, le développeur économisera beaucoup de temps, car à tout moment il peut récupérer les bouts de code fonctionnant et les intégrer à une autre application.



Le diagramme ci-dessous montre l'interaction entre ces parties.

**Figure 22 : MVVM - Description**



Source : TechDays 2012 – MVVM de A à Z

Analysons-les en détails :

### 3.2.1 View

La vue a pour responsabilité de définir la structure et l'apparence de l'interface graphique. Il s'agit de la page que l'utilisateur voit sur son écran. Elle est décrite en XAML, le code-behind ne contiendra que le constructeur.

Cependant, il est important de préciser que lorsque les animations sont complexes, il est parfois nécessaire d'utiliser le code-behind pour aider, voire compléter la partie graphique en XAML.

Le rôle de la vue est de générer l'interface utilisateur et le code relatif à cette interface.

**Figure 23 : MVVM - View**

Person ID :	<input type="text" value="2"/>	<input type="button" value="SEARCH"/>
First Name :	<input type="text" value="Antonio"/>	
Last Name :	<input type="text" value="Foti"/>	
Age :	<input type="text" value="46"/>	
Adress :	<input type="text" value="Melito Porto Salvo"/>	

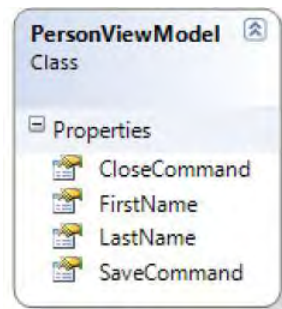
### 3.2.2 ViewModel

Le ViewModel encapsule la présentation logique et les données de la vue. La vue n'est pas directement référencée par le ViewModel. En effet, celui-ci implémente les propriétés et les commandes que la vue à besoin.

Il coordonne les interactions entre le modèle et la vue. Il manipule et convertit les données pour que la vue puisse les utiliser plus facilement et peut également implémenter de nouvelles propriétés que le modèle ne possède pas.

En résumé, nous pouvons dire que le ViewModel expose les propriétés, récupère les données et la vue va venir les consommer.

Figure 24 : MVVM - ViewModel



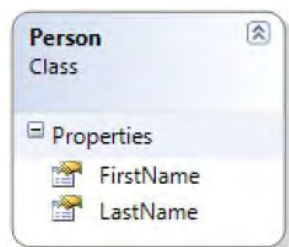
Source : Alessandro Andreosè – Separazione dei ruoli tra Designer e Developer

### 3.2.3 Model

Dans le Model, nous trouvons tous les objets et les règles métiers. Il s'occupe également de la validation des données.

Le modèle encapsule les données. Il dialogue avec le ViewModel, en lui fournissant un accès aux données et aux services.

Figure 25 : MVVM - Model



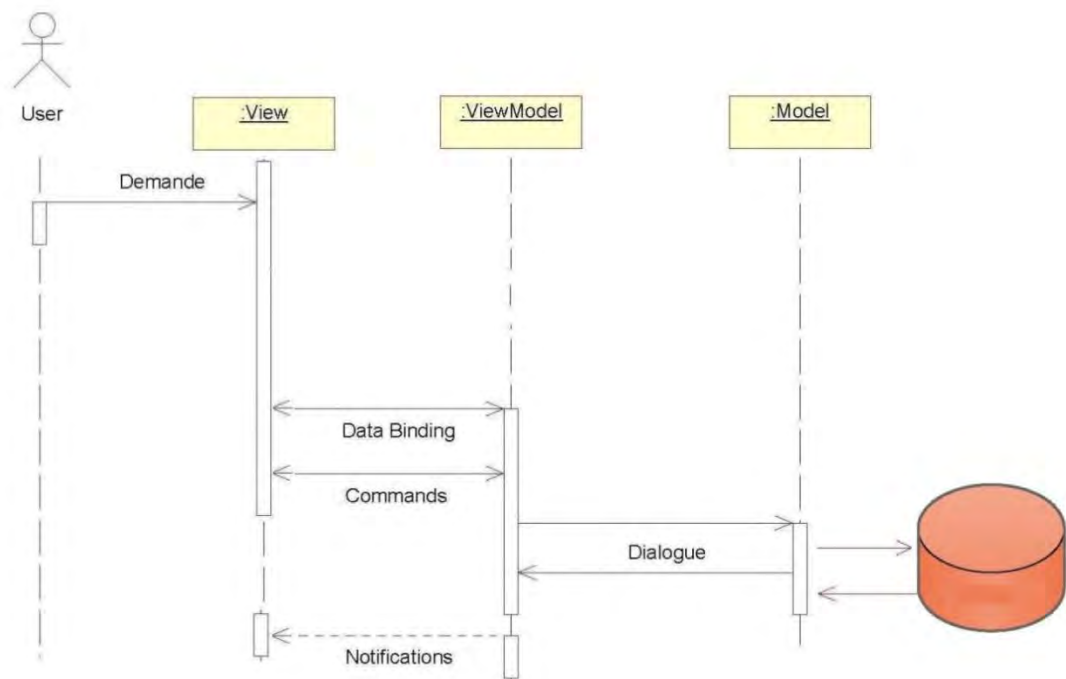
Source : Alessandro Andreosè – Separazione dei ruoli tra Designer e Developer

### 3.2.4 Fonctionnement

L'utilisateur devant son écran (View) va effectuer une demande. Les éléments de la vue sont liés au ViewModel grâce au Data Binding. Ce dernier va récupérer toutes les informations auprès du Model. Le Model va traiter la demande en accédant aux bases de données ou à des services. Ensuite, il va retourner le résultat de ses recherches au ViewModel, qui les transmettra à son tour à la View.

Le diagramme ci-dessous illustre le mécanisme de base.

Figure 26 : MVVM – Diagramme



### 3.3 Data Binding

Comme son nom l'indique, le Data Binding sert à relier des données à l'interface utilisateur.

Figure 27 : Data Binding

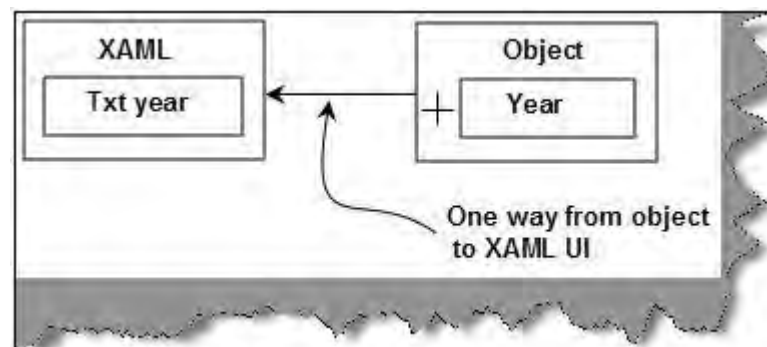
```
<sdk:Label Height="25" Grid.Row="3" Grid.Column="0"
    HorizontalAlignment="Right" Name="label2" VerticalAl
    Width="120" Content="First Name : "/>
<TextBox Height="25" Grid.Row="3" Grid.Column="1"
    HorizontalAlignment="Left" Name="txtFirstName"
    VerticalAlignment="Top" Width="200" Text="{Binding Path=PersonData.FirstName, Mode=TwoWay}'
```

```
public class PersonData
{
    public int PersonId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public int Age { get; set; }
    public string Address { get; set; }
}
```

Dans le MVVM, le Data Binding joue un rôle important. Il existe différentes manières pour pouvoir implémenter ces liaisons :

- OneWay : ce mode est utilisé lorsque l'utilisateur ne doit faire aucune modification de données. Il les affiche simplement. Les données vont uniquement de l'objet à l'interface graphique.

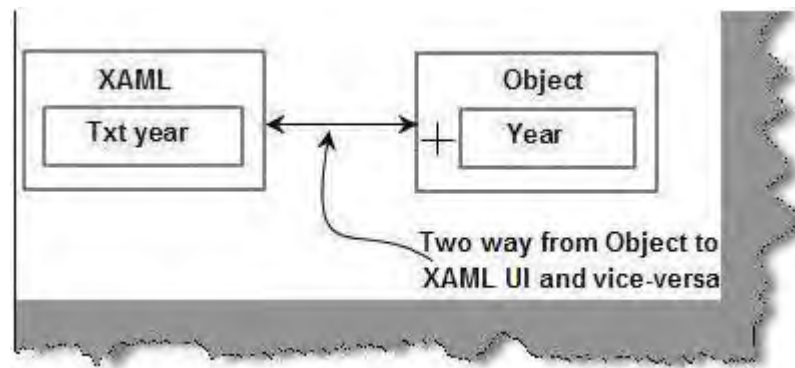
Figure 28 : Data Binding - OneWay



Source : <http://www.codeproject.com/Articles/37250/One-Way-Two-Way-and-One-Time-Bindings-using-Silver>

- **TwoWay :** ce mode permet à l'utilisateur de lire les données et de les modifier.  
Il assure la synchronisation entre les données et l'interface utilisateur. Ainsi, tous les changements de données sont directement visibles sur l'interface et tout changement de l'interface est notifié à l'objet.

**Figure 29 : Data Binding - TwoWay**



Source : <http://www.codeproject.com/Articles/37250/One-Way-Two-Way-and-One-Time-Bindings-using-Silver>

- **OneTime :** ce mode circule de l'objet à l'interface utilisateur. Il n'est fait qu'une seule fois.

**Figure 30 : Data Binding - Mode**

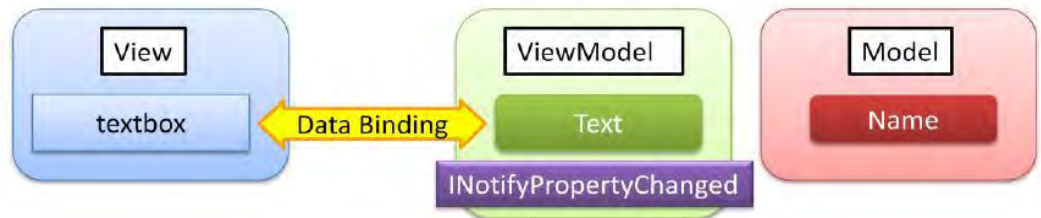
```
<sdk:Label Height="25" Grid.Row="3" Grid.Column="0"
    HorizontalAlignment="Right" Name="label2" VerticalAlignment="Center"
    Width="120" Content="First Name : "/>
<TextBox Height="25" Grid.Row="3" Grid.Column="1"
    HorizontalAlignment="Left" Name="txtFirstName"
    VerticalAlignment="Top" Width="200" Text="{Binding Path=PersonData.FirstName Mode=TwoWay}"
```

Pour que le binding soit possible, l'élément source doit pouvoir notifier qu'il y a eu un changement.

C'est pourquoi, il doit implémenter l'interface :

- INotifyPropertyChanged qui possède l'évènement PropertyChanged

**Figure 31 : Data Binding - INotifyPropertyChanged**



Source : Alessandro Andreosè – Separazione dei ruoli tra Designer e Developer

**Figure 32 : Data Binding - PropertyChanged**

```
public class ViewModelBase : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    public bool isDesignTime
    {
```

### 3.4 Commands

Commands sont des objets qui représentent une action de type Sauvegarder ou Supprimer.

Il implémente l'interface ICommand.

ICommand possède 3 méthodes que nous devons implémenter lors de son utilisation :

- Execute
- CanExecute
- CanExecuteChanged

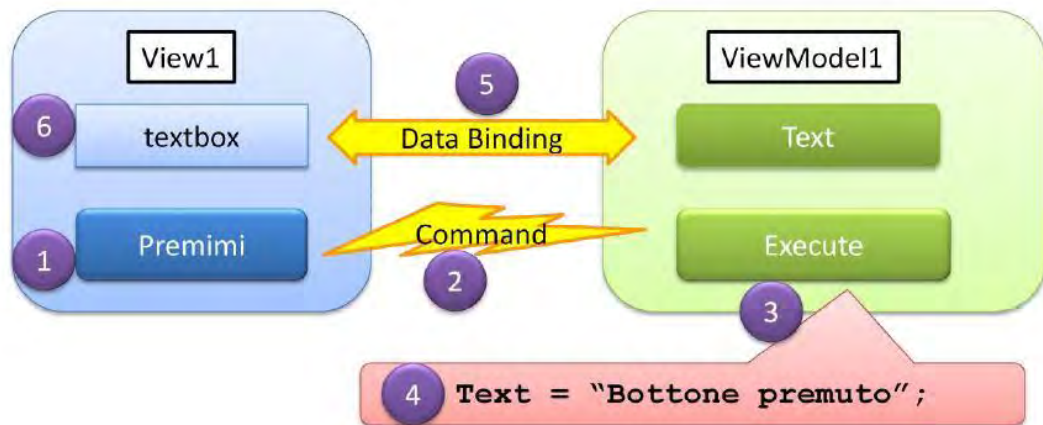
Ces méthodes sont déclenchées soient par code, soit par un élément comme un Button, une CheckBox ...

Prenons l'exemple d'une vue avec un champ de recherche et un bouton pour lancer la recherche. Lorsque l'utilisateur appuie sur le bouton, les champs se complètent.

**Figure 33 : Commands - Exemple**

Person ID :	<input type="text" value="2"/>	<input type="button" value="SEARCH"/>	Person ID :	<input type="text" value="2"/>	<input type="button" value="SEARCH"/>
First Name :	<input type="text"/>		First Name :	<input type="text" value="Antonio"/>	
Last Name :	<input type="text"/>		Last Name :	<input type="text" value="Foti"/>	
Age :	<input type="text"/>		Age :	<input type="text" value="46"/>	
Adress :	<input type="text"/>		Adress :	<input type="text" value="Melito Porto Salvo"/>	

**Figure 34 : Commands - Schéma**



Source : Alessandro Andreosè – Separazione dei ruoli tra Designer e Developer

Les étapes :

1. L'utilisateur remplit le champ (Person ID) et appuie sur le bouton (SEARCH)
2. Le bouton déclenche une commande
3. La commande fait appel à la méthode "FindPersonCommand"
4. Les variables "Text" sont implémentées
5. Grâce au binding, les textbox sont mises à jour
6. L'utilisateur voit les champs modifiés dans les textbox

## 3.5 MVVM Frameworks

Il existe sur le marché de nombreux frameworks<sup>4</sup> qui peuvent nous aider à implémenter le pattern MVVM, nous pouvons citer :

- PRISM
- MVVM Light Toolkit
- Simple MVVM Toolkit
- Caliburn

### 3.5.1 PRISM

PRISM est l'abréviation de Composite Application Guidance for WPF and Silverlight. Il s'agit d'un outil créé en 2008 par l'équipe Patterns & Practices de Microsoft pour aider les développeurs à mettre facilement en place une architecture MVVM.

Actuellement, la version 4.1 est disponible. Microsoft prépare des mises à jour du produit pour qu'il soit compatible avec Windows 8.

Figure 35 : Prism 4.1



Source : <http://msdn.microsoft.com/en-us/practices/bb232643>

A la fin de ce document, vous trouverez un exemple d'application créé avec l'aide de PRISM.

<sup>4</sup> <http://www.japf.fr/silverlight/mvvm/index.html>



### 3.5.2 MVVM Light Toolkit

Pour nous éviter de ne partir de rien, Laurent Bugnion, a mis en place un framework. Il nous aide à séparer correctement les différentes couches et à créer une application testable.

Il s'agit du MVVM Light Toolkit. Actuellement la version 4 Beta1 est disponible. Il peut être utilisé pour les applications WPF, Silverlight et Windows Phone 7.

Cet outil nous propose des bibliothèques contenant des classes qui peuvent nous aider à l'utilisation du pattern. Nous avons par exemple :

- ViewModelBase class : c'est une classe qui est implémentée comme base pour les ViewModels

Figure 36 : ViewModelBase

```
public class ViewModelBase : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    public bool isDesignTime
    {
        get
        {
            return DesignerProperties.IsInDesignTool;
        }
    }

    protected virtual void OnPropertyChanged(string propertyName)
    {
        var propertyChanged = PropertyChanged;

        if (propertyChanged != null)
        {
            propertyChanged(this, new PropertyChangedEventArgs(propertyName));
        }
    }
}
```

- Messenger class : elle permet la communication entre les classes de l'application

Figure 37 : MVVM Light Toolkit



Source : <http://www.galasoft.ch/mvvm/>

### 3.5.3 Simple MVVM Toolkit

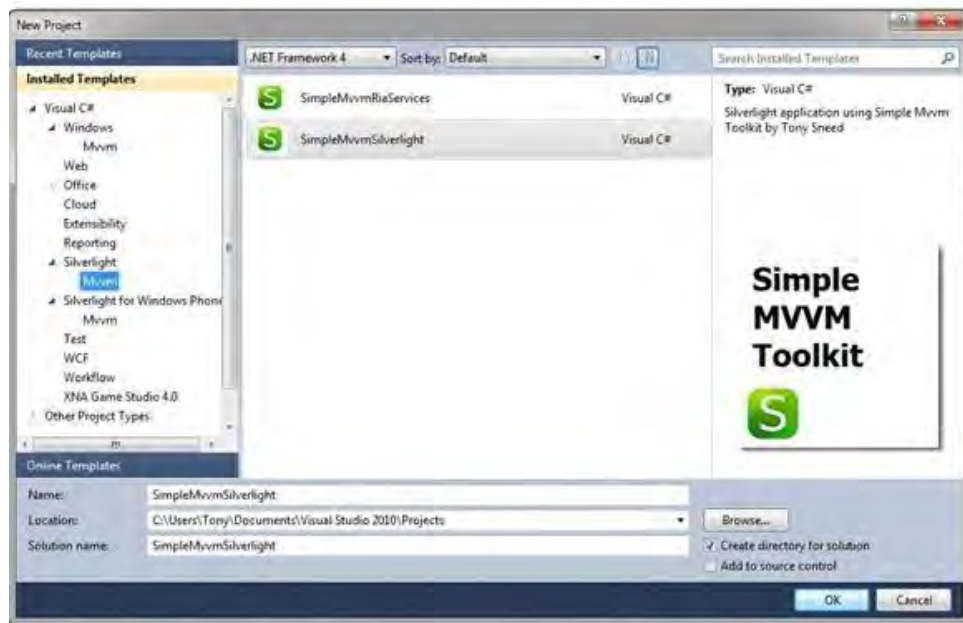
Il a été mis sur le marché par l'équipe de Tony Sneed en 2010. Il s'agit d'un outil Open Source.

Le but principal est d'aider le développeur à implémenter le pattern MVVM en toute simplicité en lui fournissant des classes d'aide ayant déjà des liens entre elles.

L'utilisation de ce produit permettra d'avoir une application extensible et maintenable.

En mars 2012, la version 3.0.0.0 est mise en ligne.

Figure 38 : Simple MVVM Light Toolkit



Source : <http://simplemvvmtoolkit.codeplex.com/wikipage?title=Getting%20Started>

### 3.5.4 Caliburn.Micro

La première version stable date du 21 février 2011. Aujourd'hui nous sommes à la version 1.3.1. Cet outil permet également la mise en place du pattern MVVM en mettant à disposition une série de classes pour la gestion de l'application.

Figure 39 : Caliburn.Micro



Source : <http://caliburnmicro.codeplex.com/documentation>

Selon leur site, Caliburn.Micro, Simple MVVM Toolkit et MVVM Light Toolkit supportent WinRT.

## 4 Positionnement de Microsoft face à Silverlight et HTML5

Avant de nous pencher sur le positionnement de Microsoft face à Silverlight et HTML5, il nous semble judicieux de se pencher sur l'histoire, le fonctionnement et l'architecture utilisée par HTML5.

### 4.1 L'histoire de HTML5

HTML (**H**yper **T**ext **M**arkup **L**anguage) a été le langage utilisé par Tim Berners-Lee, son créateur, pour la création du premier site web en 1991.

Déjà à cette époque, Microsoft avait un concurrent : Netscape. Les deux sociétés avaient créé des extensions de HTML propriétaires. Les développeurs étaient obligés de choisir soit l'un, soit l'autre. Ces solutions devenaient rapidement très coûteuses.

Tim Berners-Lee avait très vite compris la potentialité du développement du web à travers le monde. Pour cela, il était nécessaire d'introduire des standards pour que la planète entière puisse communiquer avec le même langage et de manière totalement gratuite.

C'est alors qu'il a décidé, en 1994, avec l'aide d'autres personnes, de créer le W3C : **W**orld **W**ide **W**eb **C**onsortium.

Figure 40 : Tim Berners-Lee



Source : <http://www.redicecreations.com/article.php?id=4813>

Cette organisation avait pour but de proposer des standards, d'introduire des normes pour le web.

Les principaux standards sont :

- L'interopérabilité : capacité à fonctionner avec d'autres produits
- La portabilité : application qui peut fonctionner sur n'importe quel environnement
- La pérennité des documents : s'assurer que le document reste accessible longtemps
- L'accessibilité : accès facile, rapide et partout
- La réduction des coûts : diminuer les coûts avec des produits open-source.

Quelques années plus tard, Tim Berners-Lee créait la World Wide Web Foundation. Cette fondation s'assure que les normes de W3C soient respectées et que les découvertes, qui en découlent, restent ouvertes et accessibles à tous.

W3C donnera en 1994 les premières règles pour le fonctionnement de HTML3.0. En 1998, pour la première fois, il sera possible d'utiliser une feuille de style le CSS (Cascading Style Sheets) pour améliorer la présentation des sites web.

Le 4 juin 2004, après une séance de travail avec W3C, Apple, Mozilla Foundation et Opera Software ont eu peur des décisions que prenait W3C pour le développement de XHTML2.

Ils ont alors décidé de fonder le WHATWG (Web Hypertext Application Technology Working Group) pour continuer le développement du HTML standard.

L'association permet à tous les développeurs du monde de prendre part au développement de HTML5. Il suffit d'envoyer un mail à [whatwg@whatwg.org](mailto:whatwg@whatwg.org) et les propositions seront prises en considération.

En 2007, W3C accepte la vision de WHATWG et l'intègre à son groupe pour continuer ensemble le développement de HTML5.

Les principaux concepts<sup>5</sup> sont clairs :

- Simplifier l'usage de HTML
- Officialiser les pratiques courantes
- Assurer un maximum de rétrocompatibilité

Figure 41 : Logo HTML5



Source : [http:// http://www.w3schools.com/](http://http://www.w3schools.com/)

---

<sup>5</sup> [www.alsacecreations.com/article/lire/750-HTML5-nouveautes.html](http://www.alsacecreations.com/article/lire/750-HTML5-nouveautes.html)

## 4.2 La composition de HTML5

HTML5 se compose de la manière suivante :

dans la première couche nous trouvons les organisations de standardisation :

- W3C
- ECMA : European association for standardizing information and communication systems  
Fondée en 1992, elle s'occupe de la standardisation de langage de programmation comme JavaScript, de la sécurité.

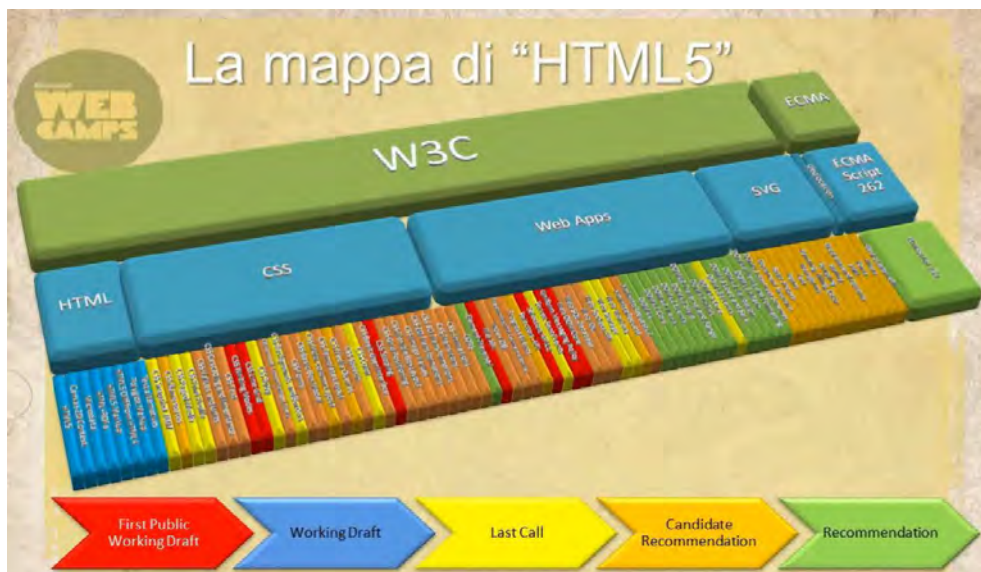
Sous chaque organisation, nous pouvons observer les éléments dont chacune d'elle s'occupe :

- W3C
  - HTML
  - CSS
  - Web Apps : Stockage et évolution des bases de données,
  - SVG : Scalable Vector Graphic (éléments graphiques)
- ECMA
  - ECMA Script 2002 : JavaScript

et sous chaque élément, nous trouvons les spécifications en cours d'évolution. Chacune d'elle a un cycle d'approbation et une maturation différente.

Les flèches colorées nous indiquent à quel niveau d'évolution elles se trouvent.

Figure 42: Composition de HTML5



Source : <http://www.microsoft.com/italy/webcamps/online/default.aspx>

### 4.3 Les nouvelles fonctionnalités

HTML5 offre à ses utilisateurs tous les moyens nécessaires pour créer des sites riches, variés, rapides et surtout mobiles.

Les principales nouvelles fonctionnalités sont :

- Simplification du code  
Avant :

**Figure 43 : Simplification de code - Avant**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <link rel="stylesheet" type="text/css" href="design.css" />
  <script type="text/javascript" src="script.js"></script>
</head>
<body>
</body>
</html>
```

Source : <http://www.html5-css3.fr/>

Après :

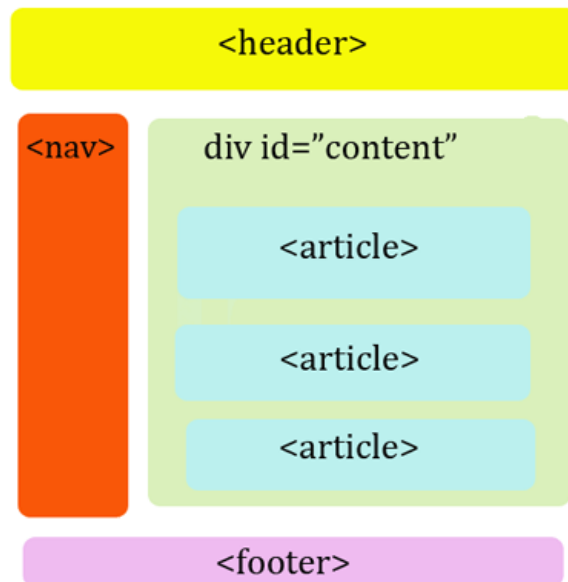
**Figure 44 : Simplification de code - Après**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8" />
  <link href="design.css" rel="stylesheet" />
  <script src="script.js"></script>
</head>
```

Source : <http://www.html5-css3.fr/>

- de nouvelles balises

Figure 45 : Nouvelles balises



Source : <http://www.html5-css3.fr/>

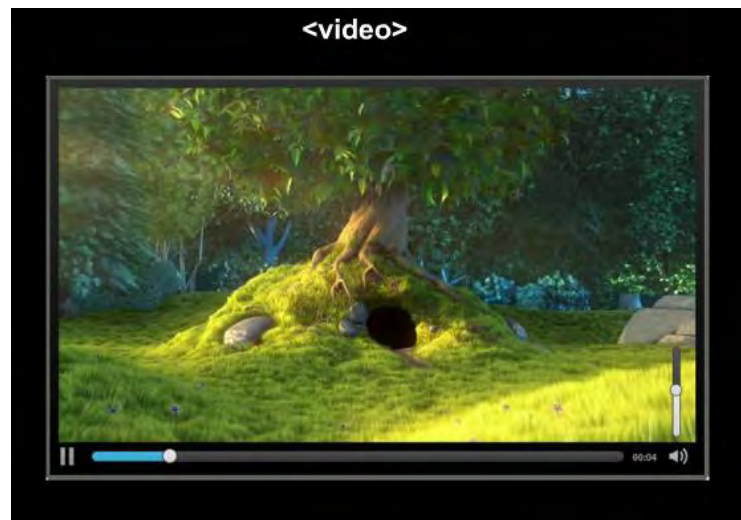
Figure 46 : Nouvelle balise - `<audio>`



Source : <http://www.html5-css3.fr/>



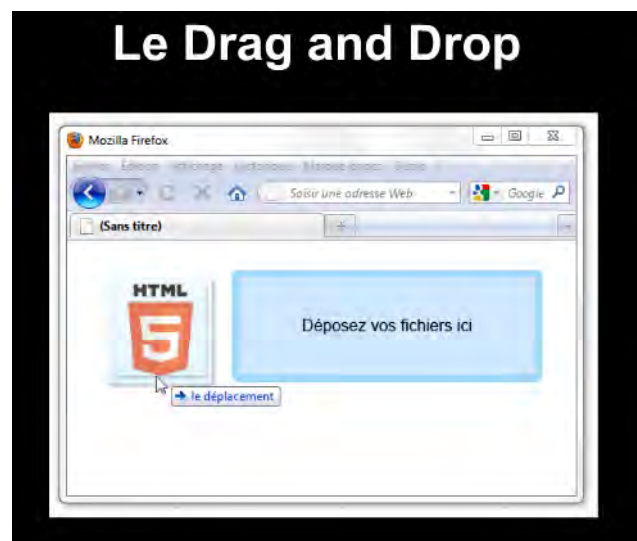
**Figure 47 : Nouvelle balise - <video>**



Source : <http://www.html5-css3.fr/>

- Drag and Drop

**Figure 48 : Drag and Drop**

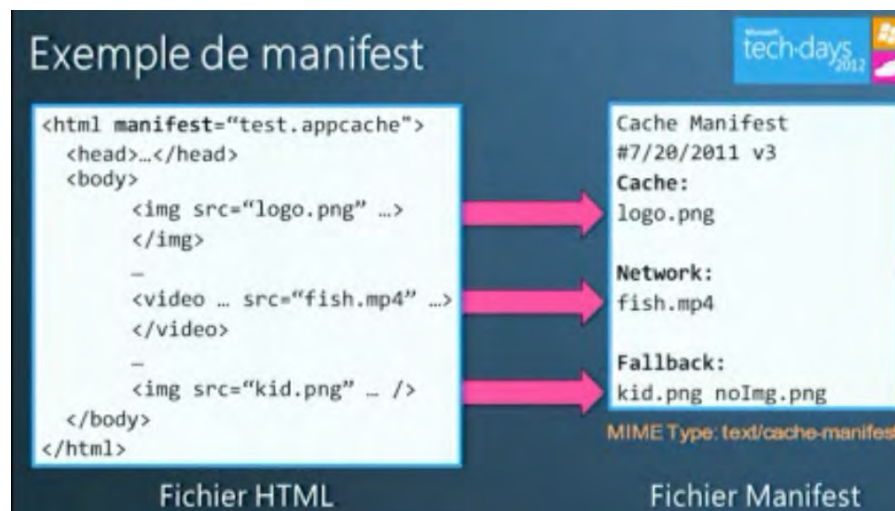


Source : <http://www.html5-css3.fr/>



- les applications offline : application utilisée en dehors de toute connexion.  
Le fonctionnement est le suivant :  
Il s'agit de mettre en cache un ensemble de ressources.  
Un fichier **manifest** est disponible et il faut le compléter avec toutes les ressources nécessaires.  
Lors de la 1<sup>ère</sup> connexion, le navigateur va télécharger ces ressources et si la connexion venait à manquer, l'application continuerait à fonctionner.

Figure 49 : Applications offline



Source : TechDays 2012 - Les nouveautés HTML5 et CSS3 dans Internet Explorer 10 (RIA201)

- lecture audio et vidéo sans plug-in additionnel
- amélioration des formulaires. Les validations sont faites par le navigateur.
- le stockage :  
Existence d'une base de données, IndexedDB, côté client.  
Ce n'est pas une base de données relationnelle. Toutes les requêtes se font en JavaScript. Cette base de données est créée avec le système « Clé / Valeur »  
Il existe 2 types de stockage :
  - Le LocalStorage  
En dehors de la session
  - Le SessionStorage  
Il est utilisé durant toute la durée de la session.

- la géolocalisation

**Figure 50 : Géolocalisation**

```
navigator.geolocation.getCurrentPosition(function(pos){
    alert(position.coords.latitude + " " + pos.coords.longitude);
});
```

Source : <http://www.html5-css3.fr/>

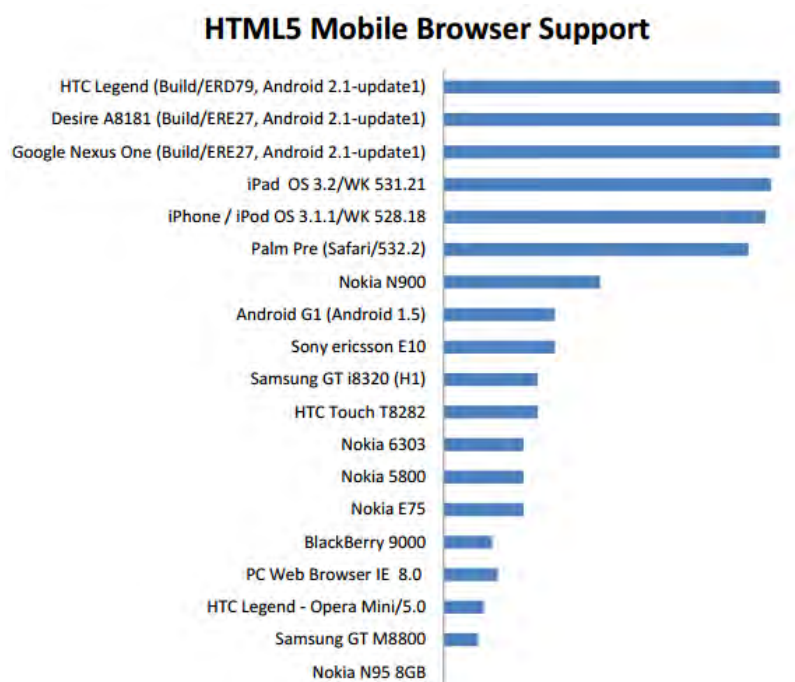
- le développement 2D et 3D

#### 4.4 Le développement d'applications mobiles

Les ventes de Smartphones et de tablettes ont explosé ces dernières années. Actuellement, il existe un vrai potentiel de développement d'applications mobiles pour le particulier ou le professionnel. Le développement se trouve à un tournant de l'histoire du web et les nouvelles fonctionnalités de HTML5 sont là pour aider les développeurs.

Le tableau, ci-dessous, nous montre la prise en charge de HTML5 par les Smartphones que l'on trouve sur le marché.

**Figure 51 : Smartphone et HTML5**



Source : <http://www.servicesmobiles.fr/files/momac-html5-mobile-whitepaper.pdf>

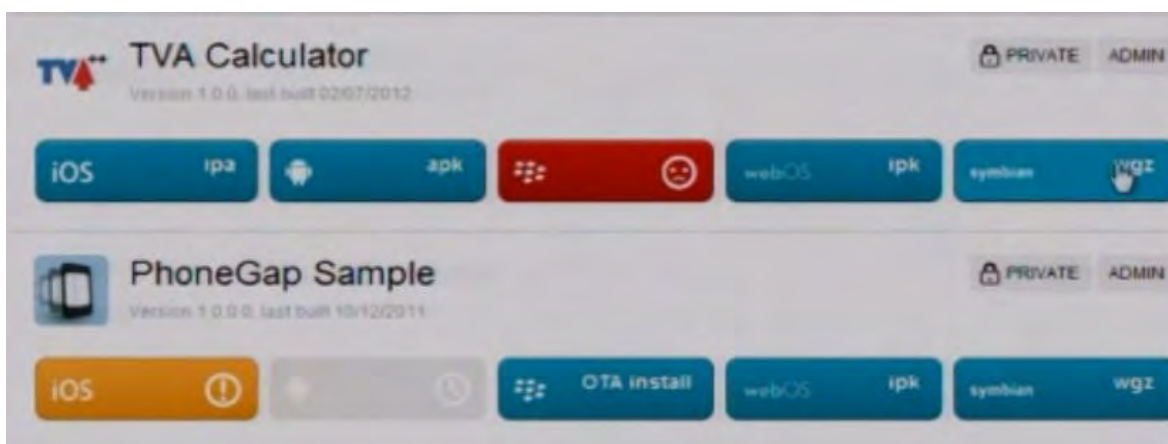
Lorsque nous développons avec HTML5, nous devons être attentifs, car toutes les nouvelles spécifications doivent encore évoluer et il arrive souvent que les implémentations évoluent également et ces spécifications ne fonctionneront jamais.

Pour cela, il existe des sites comme « HTML5 Please » qui aide à savoir si les spécifications fonctionnent et sur quelle version de navigateur.

Nos recherches nous ont fait découvrir un Framework open-source qui nous aide à développer des applications mobiles multiplateformes. Il va créer des packages de l'application et les tester.

Ensuite, il renvoie le résultat comme le montre la figure ci-dessous. Il s'agit de PhoneGap<sup>6</sup>.

**Figure 52 : PhoneGap - Exemple**



Source : Tech Days 2012 : <http://www.microsoft.com/france/mstechdays/programmes/parcours.aspx>

## 4.5 L'architecture de HTML

HTML est, par définition, un langage à balise utilisé pour la création de site web.

Seul, il n'a pas d'architecture à proprement parler, car il ne permet de faire que du contenu statique.

Cependant, nous le constatons dans notre apprentissage qu'il n'est jamais utilisé seul. Nous lui associons toujours une page CSS pour donner des styles à nos pages, du code JavaScript pour la manipulation de données et la gestion des événements.

Dès cet instant, nous pouvons parler d'architecture. Il est possible d'utiliser l'une ou l'autre des architectures que nous avons abordé dans les chapitres précédents, c'est-à-dire MVC ou MVVM.

Pour cela, il est recommandé de créer une application en utilisant la programmation orientée objet.

Le choix, dès lors, dépendra du programmeur. Il la choisira selon son affinité ou selon les spécificités de l'application.

---

<sup>6</sup> <http://phonegap.com/>

Nous trouvons sur le marché, différents frameworks aidant à créer des applications en respectant les modèles d'architecture. Parmi cela, nous pouvons citer Knockout.js pour le MVVM et Backbone.js pour le MVC.

Les figures, ci-dessous, nous montrent comment sont liées les deux vues entre elles, grâce au binding :

## La View

Figure 53 : HTML – MVVM – La View

```
<!-- This is a *view* - HTML markup that defines the appearance of your UI -->
<p>First name: <strong data-bind="text: firstName"></strong></p>
<p>Last name: <strong data-bind="text: lastName"></strong></p>
```

View (HTML)

Source : <http://learn.knockoutjs.com/#/?tutorial=intro>

## Le ViewModel

Figure 54 : HTML – MVVM – Le ViewModel

```
// This is a simple *viewmodel* - JavaScript that defines the data and behavior
function AppViewModel() {
  this.firstName = "Bert";
  this.lastName = "Bertington";
}

// Activates knockout.js
ko.applyBindings(new AppViewModel());
```

View Mo  
(JavaScri

Source : <http://learn.knockoutjs.com/#/?tutorial=intro>

## Le résultat

Figure 55 : HTML – MVVM – Résultat

### Output

First name: **Bert**

Last name: **Bertington**

Source : <http://learn.knockoutjs.com/#/?tutorial=intro>

## 4.6 Positionnement de Microsoft

Depuis plusieurs mois, les adeptes des outils de Microsoft ne savaient plus quoi penser. Des rumeurs courraient à travers la toile et les incertitudes augmentaient. Est-ce que Silverlight va mourir ? Va-t-il être remplacé ? A quoi devons-nous nous attendre ? Et des milliers d'autres questions bousculaient les esprits des programmeurs.

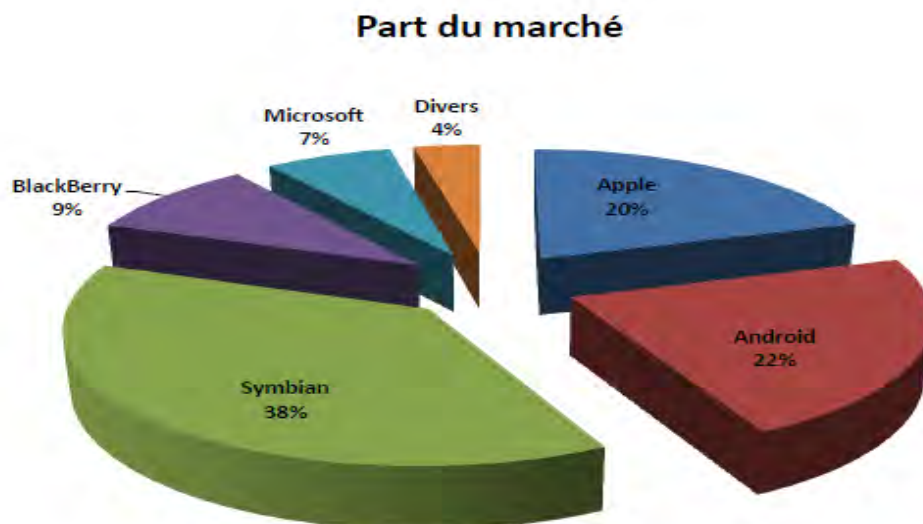
Les TechDays 2012 ont dévoilé leurs plus grands secrets.

Microsoft a décidé d'attaquer le marché des tablettes et des Smartphones.

En 2011, sa part de marché, pour la vente de Smartphones, n'était que de 7%, mais Microsoft pense pouvoir modifier cette réalité en annonçant que HTML5 serait dorénavant un langage utilisé par ses développeurs.

Elle souhaite récupérer quelques parts de ce marché en pleine essor. Et pour cela, elle inclut dans ses langages de programmation le HTML5.

Figure 56 : Part du marché en Europe (2011)<sup>7</sup>



Source : <http://www.journaldunet.com/developpeur/expert/51048/le-developpement-multi-plateforme---enjeux--promesses-et-realite.shtml>

Mais pourquoi décide-t-elle de passer à HTML5 ?

Les réponses sont multiples.

Microsoft souhaite développer des applications mobiles multiplateformes, grâce à HTML5 combiné au CSS3 et à JavaScript, et vaincre son rival de toujours Apple. Mais il est difficile de trouver de bons développeurs C#, alors que les adeptes au HTML5 sont nombreux, car l'apprentissage de ce langage est plus facile.

---

<sup>7</sup>

[http://www.comscore.com/fre/Press\\_Events/Press\\_Releases/2011/9/Android\\_Captures\\_number\\_2\\_Ranking\\_Among\\_Smartphone\\_Platforms\\_in\\_EU5](http://www.comscore.com/fre/Press_Events/Press_Releases/2011/9/Android_Captures_number_2_Ranking_Among_Smartphone_Platforms_in_EU5)

Microsoft espère qu'ils vont s'empresse d'adopter Windows et ses outils pour développer des applications mobiles et approvisionner le Windows Store. Windows Store est la plateforme de distribution des applications Metro.

Quels sont les avantages et les désavantages d'utiliser HTML5 ?

Avantages	Désavantages
Facilité de développement grâce aux standards	Technologie jeune (arrivera à maturité en 2014)
Couvre un large éventail de plateforme	Supports différents selon les plateformes
Déploiement simplifié	Spécifications pas valables pour toutes les plateformes
Disponibilité immédiate	
Interface graphique riche	
Gestion des ressources en mode déconnecté	
Base de données en local	
Utilisation de médias sans plug-in	

#### 4.6.1 Plateforme Windows 8 et les outils

La plateforme Windows 8, dont le nom de code était Jupiter, nous montre le nouveau visage de développement de Microsoft.

Elle est séparée en deux parties :

- Les applications Metro Style
- Les applications Desktop.

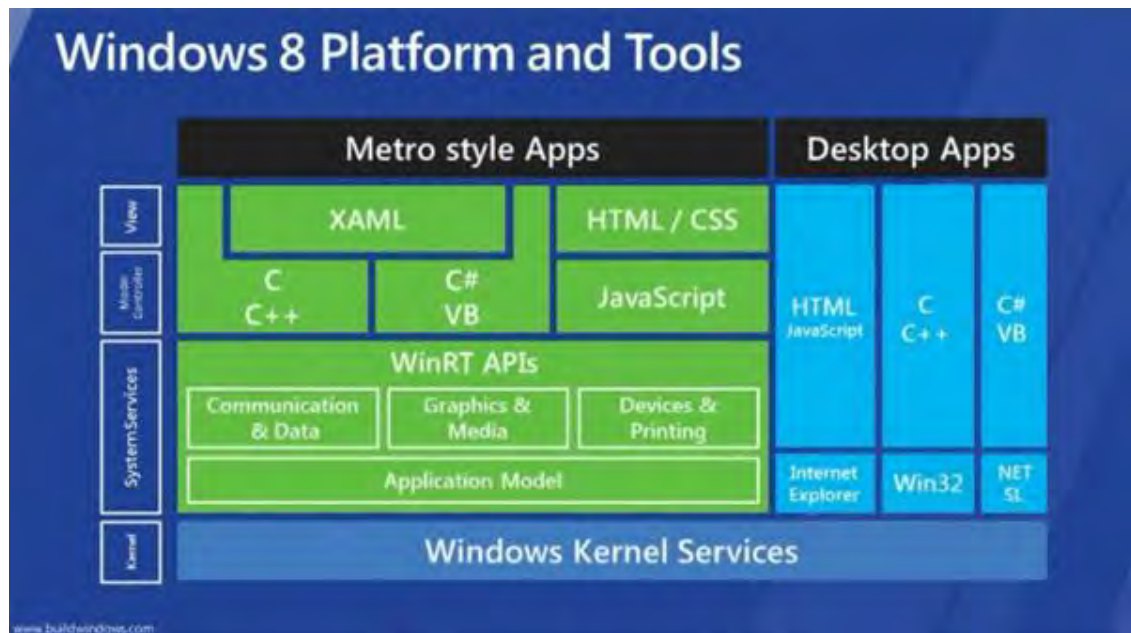
Desktop est le monde que nous connaissons aujourd'hui et qui existera demain. En fait, il s'agit de Windows 7 corrigé et amélioré. Les plug-ins, comme Silverlight, seront toujours présents.

Metro Style est adapté aux développements d'applications pour tablettes et Smartphones. Il va aider les programmeurs à implémenter des applications mobiles. Dans cette partie, Microsoft arrête le support de plug-in.

Avec Metro Style, Microsoft souhaite attirer les développeurs de tous les langages. Chacun y trouvera son compte. Les anciens pourront continuer à développer en utilisant XAML et .NET. Par la même occasion, elle ouvre son environnement aux programmeurs HTML/CSS et JavaScript.

Les outils de développement n'ont pas changé. En effet, Blend et Visual Studio ont été améliorés.

**Figure 57 : Windows 8 - Plateforme**



Source : <http://www.developpez.com/actu/36957/>

IE10, sur Windows 8, possède ces deux modes de fonctionnement.

#### 4.6.2 Et Silverlight ?

Silverlight est toujours là pour les applications Desktop. La version 5 vient de sortir.

Microsoft distingue bien deux parties du développement :

- Le développement web avec l'utilisation de Silverlight
- Le développement des applications mobiles avec HTML5

L'un ne remplace pas l'autre, mais ils se complètent en développant chacun une partie bien définie.



## 5 Metro Style Apps

---

La grande nouveauté, lors des TechDays 2012, a été la présentation de Windows 8. L'interface de Windows 8 a été totalement réinventée. Elle consolide, d'un côté le desktop, et de l'autre, elle introduit un nouveau style pour le développement d'application mobile pour tablettes ou Smartphones, le Metro Style. Ce style est déjà utilisé pour le développement d'applications de Windows Phone. Dans ce chapitre, nous allons étudier plus en détail ce qui a poussé Microsoft à converger sa vision vers Metro.

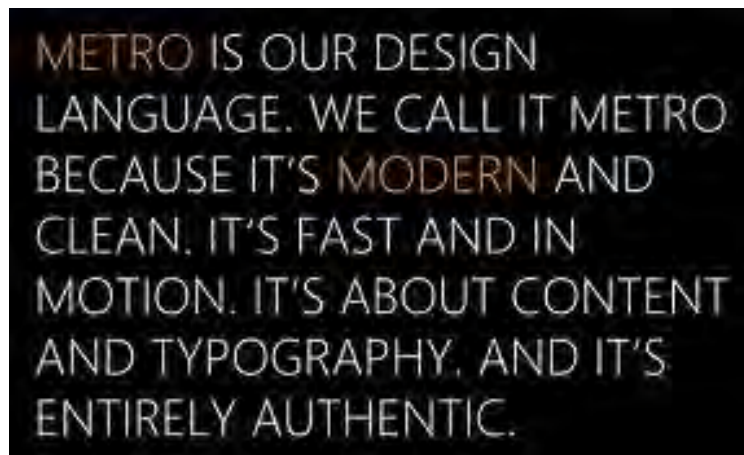
### 5.1 L'histoire de Metro Style

En 1998, Microsoft mit en place un nouveau langage de design : Metro Style. Il avait été conçu spécialement pour les applications de Windows Phone.

Metro est basé sur trois concepts :

- Simplicité
- Propreté
- Modernité

**Figure 58 : Metro Design Language**



Source : [http://it.wikipedia.org/wiki/Metro\\_\(interfaccia\\_grafica\)](http://it.wikipedia.org/wiki/Metro_(interfaccia_grafica))

En 2011, à la D9 Conference de Californie, Microsoft confirma que Metro serait le nouveau modèle d'application de Windows 8.

Durant ces vingt dernières années, Microsoft a mis sur le marché différentes versions. L'interface graphique à "fenêtre" a eu au cours des années une certaine évolution pour devenir de plus en plus sophistiquée pour attirer les utilisateurs.



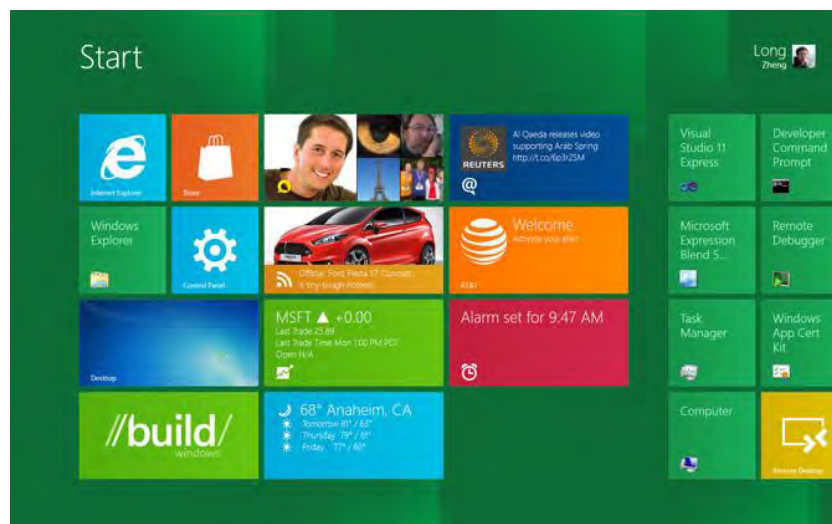
Les améliorations étaient surtout orientées vers :

- la simplification d'écriture d'applications
- la simplification des services offerts aux utilisateurs
  - connexions internet ou intranet
- l'augmentation de la vitesse des PC.

Pendant de nombreuses années, l'ordinateur est resté un outil de travail utilisant un langage technique complexe, construit toujours autour d'un écran, d'une souris et d'un clavier.

De nos jours, il est devenu un outil que tout le monde utilise quotidiennement même si ce n'est que pour regarder la météo. De plus, nous constatons que la nouvelle tendance est l'utilisation des écrans tactiles.

**Figure 59 : Windows 8 – Start page**



Source <http://www.istartedsomething.com/20110914/an-analysis-of-metro-on-windows-8-developer-preview/>

En effet, l'interface graphique reprend clairement le principe de l'écran tactile de Windows Phone.

Microsoft veut offrir aux utilisateurs une certaine homogénéité entre les systèmes. Elle leur propose donc la même interface que ce soit sur mobile, tablette, ordinateur ou Xbox. L'utilisateur pourra ainsi passer d'un outil à un autre sans avoir à comprendre, réétudier son fonctionnement et cela de manière tout à fait naturelle.

Pour les inconditionnels, il sera toujours possible de basculer en mode Windows 7 et d'utiliser le clavier et la souris.

## 5.2 Metro Style Apps

Metro Style Apps est un nouveau type d'application fonctionnant sur Windows 8. Elle permet de créer la même application pour différentes tailles d'écran. Ces applications sont conçues de manière à recouvrir tout l'écran, pour éviter que l'utilisateur ne soit distrait par d'autres fenêtres.

Figure 60 : Windows 8 - Screen



Source : [http://msdn.microsoft.com/en-us/library/windows/apps/hh974576\(d=printer\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/hh974576(d=printer).aspx)

En effet, lorsque nous créons un nouveau projet Windows Metro Style, il est possible de configurer les orientations que l'application doit supporter.

Figure 61 : Windows Metro Style – Orientation preferences



Microsoft vante les qualités de Metro Style Apps :

- Fonctionne aussi bien avec
  - Tactile
  - Stylo
  - Clavier
- La barre de navigation est cachée pour offrir plus de place. Elle réapparaît lorsque l'on clique dans un angle de l'écran.
- Les applications non utilisées se ferment toutes seules pour diminuer la consommation de mémoire.

- Les développeurs peuvent choisir le langage de programmation qui leur convient le mieux.
- Ils ont la possibilité de vendre leurs créations dans le Windows Store.

Ces applications seront signées et auront passé les contrôles de vérification. Microsoft met à disposition "Windows App Certification Kit". L'utilisateur peut vérifier si son application est conforme aux règles de Microsoft.

L'application sera contrôlée à nouveau avant sa publication dans le Windows Store.

Après la vente, Microsoft garde le 30% de commission. Si l'application a un chiffre d'affaires de plus de \$ 25'000.00, elle ne garde que le 20%.

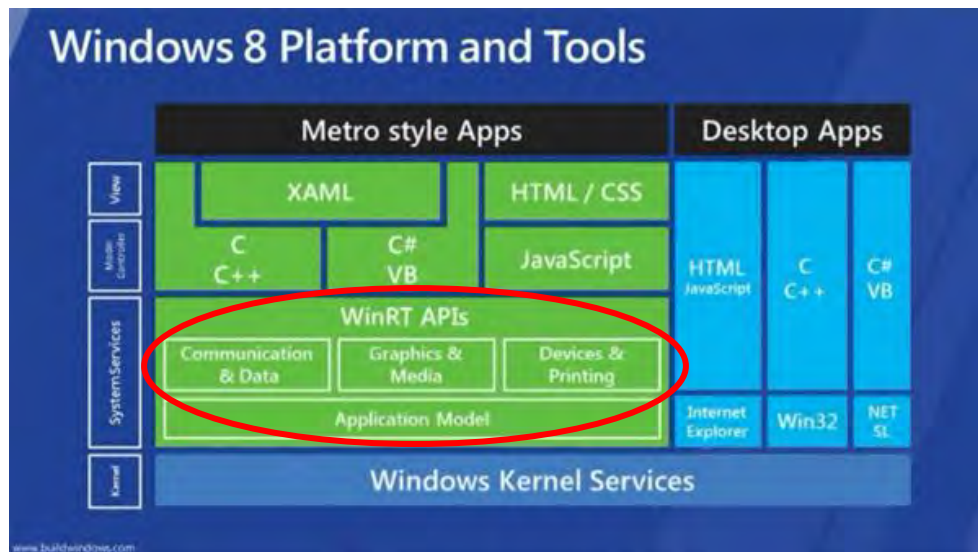
**Figure 62 : Windows Store**



Source : [http://msdn.microsoft.com/en-us/library/windows/apps/hh974576\(d=printer\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/hh974576(d=printer).aspx)

Lors de la présentation de Windows 8, nous constatons que Metro Style Apps se compose, en dessous des langages de programmation, d'une couche "System Services" : le WinRT APIs,

Figure 63 : Windows 8 - Plateforme



Source : <http://www.developpez.com/actu/36957/>

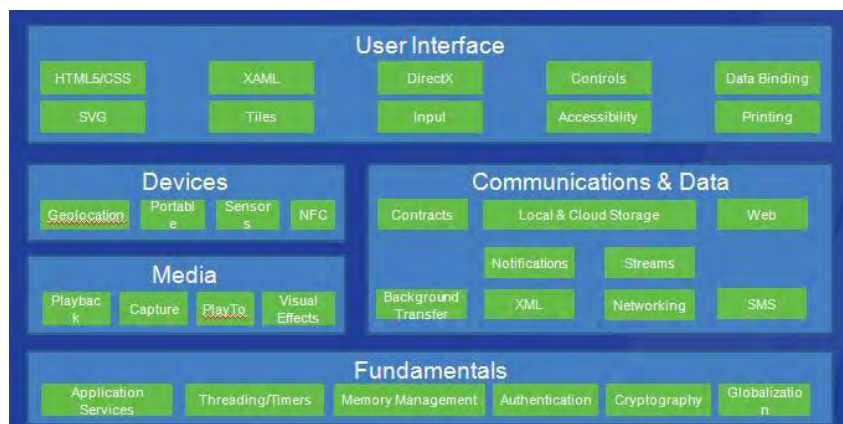
De quoi s'agit-il?

### 5.2.1 WinRT APIs

Le Windows Runtime, WinRT, est le nouveau modèle de programmation de Microsoft. Il est la colonne vertébrale de Metro Style Apps dans Windows 8. Selon le schéma ci-dessus, WinRT est une couche abstraite. Il est composé d'un modèle d'application et de trois APIs :

- Communication & Data
- Graphics & Media
- Device & Printing

Figure 64 : WinRT



Source : <http://channel9.msdn.com/Events/Lang-NEXT/Lang-NEXT-2012/The-Windows-Runtime>

Il représente un nouvel environnement de développement différent de Win32. C'est également un ensemble de bibliothèques qui offre des fonctionnalités de Windows accessibles par les différents langages.

A ce stade, nous devons préciser que WinRT ne remplace pas Win32, il en augmente la capacité.

Cependant WinRT résout différents problèmes qu'avait Win32. Nous pouvons citer :

- La sécurité
  - Les applications tournent dans un bac à sable, elles sont ainsi exécutées dans un environnement isolé.
  - Demande d'autorisation à l'utilisateur pour les accès au système de fichiers, à la camera ou au réseau.
  - Création d'une base de données en local pour la sauvegarde des données.
- Installation rapide
- Meilleure performance
- Utilisation de la programmation objet

### 5.2.2 Applications

La même application est créée en Silverlight et HTML5 pour comprendre et analyser les différences entre ces deux langages.

Un article<sup>8</sup> sur internet prétend qu'il suffit de mettre les nouveaux namespaces dans une application Silverlight et d'adapter le code pour la porter dans Metro. Cependant Metro Style Apps, selon nos recherches, ne supporte pas le plug-in Silverlight.

Alors où se trouve la vérité?

---

<sup>8</sup> <http://www.codeproject.com/Articles/328551/Part-3-Introduction-to-WinRT-the-new-Windows-Runti>  
(page 5)

## 6 Applications

Dans ce paragraphe, une application Silverlight est mise en comparaison avec une application créée avec Metro Style.

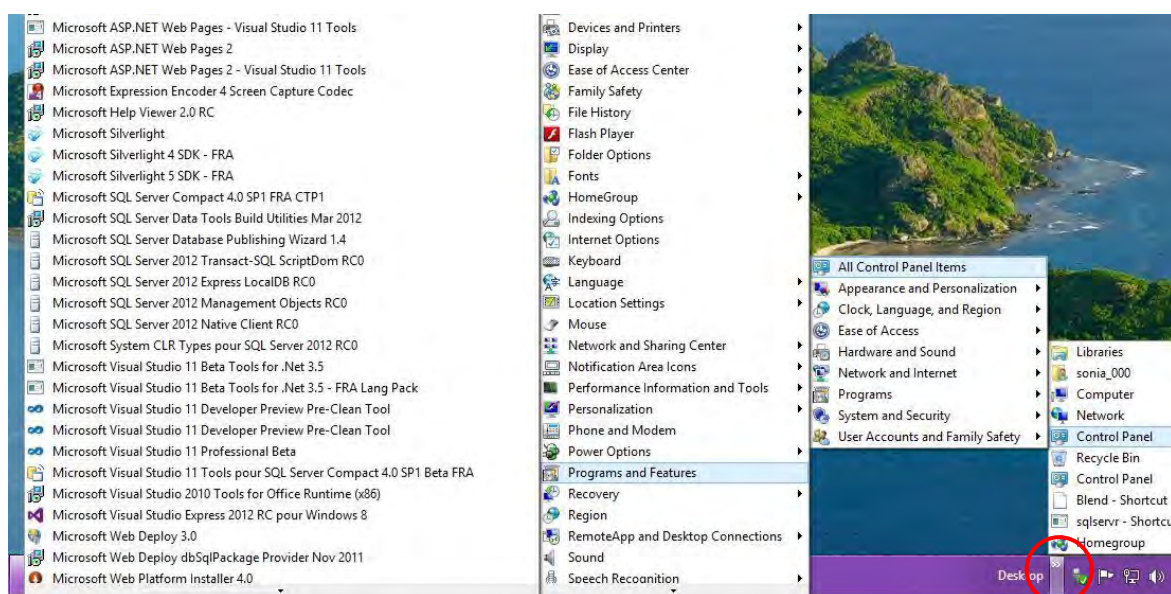
### 6.1 Environnement de travail

#### 6.1.1 Windows 8

La première chose, à faire, a été la création de l'environnement de travail. Pour cela, une machine virtuelle a été créée avec Windows 8. L'interface utilisateur nous offre une vision colorée et nous pouvons constater qu'elle est conçue pour le Touch screen.

Au début, il est difficile de maîtriser ces changements. Mais, petit à petit, nous nous rendons compte que Microsoft laisse aux utilisateurs inconditionnels de la souris la possibilité de naviguer entre les applications facilement, même si parfois cette possibilité est bien cachée.

Figure 65 : Navigation – Windows 8



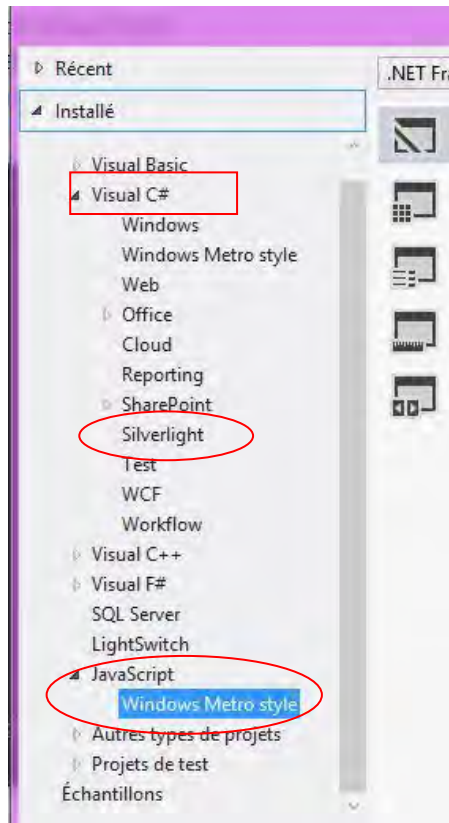


### 6.1.2 Visual Studio 2011

L'installation de Visual Studio 2011 s'est passée sans trop de problème.

Visual Studio 2011 nous offre le menu complet de tous les langages de programmation supportés.

Figure 66 : Visual Studio 2011 - Menu



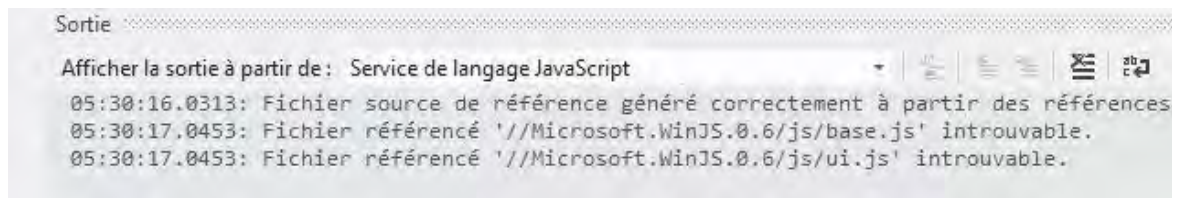
La création d'une application Silverlight se trouve sous Visual C# et nous remarquons qu'il n'y a pas la possibilité de l'utiliser dans Windows Metro style.

En ce qui concerne JavaScript, nous pouvons créer une application Windows Metro style.

Cependant, nous avons une surprise lorsque nous créons un nouveau projet en JavaScript. Un message d'erreur nous avertit qu'il y a un problème. (Figure 67).

Au début, nous avons pensé qu'une erreur de manipulation, lors de l'installation, s'était produite. C'est pourquoi, nous avons désinstallé et réinstallé le tout maintes fois. Mais le problème subsistait toujours.

Figure 67 : Visual Studio 2011 – Message erreur



### 6.1.3 Visual Studio Express 2012 RC

A ce stade, nous avons décidé d'installer Visual Studio Express 2012 RC for Windows 8 et obtenu une licence pour développeur.

Le menu de ce programme n'offre que les langages supportés par Windows Metro style.

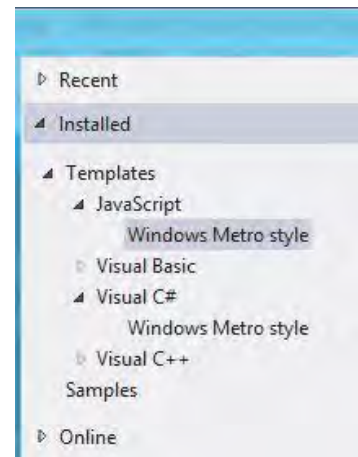
Nous pouvons créer et lancer sans problème une application JavaScript.

Pour comprendre d'où vient l'erreur de la figure 67, nous décidons de comparer les pages default.js.

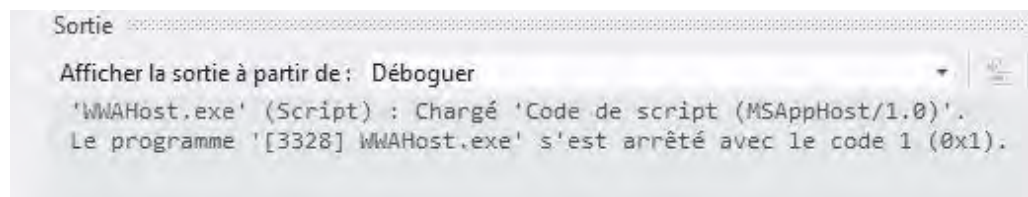
Nous créons le même projet :

- dans Visual Studio Express RC :  
“Templates/JavaScript/ Windows Metro Style” (figure 68).

**Figure 68 : Visual Studio Express - Menu**



**Figure 69 : Visual Studio Express - Message**



- dans Visual Studio 2011 :  
“Installé/ JavaScript/ Windows Metro Style” (figure 66)

L'exécution de l'application est impossible et nous obtenons le message d'erreur de la figure 67.



Comparons les deux pages default.js, pour déterminer les différences :

Figure 70 : Visual Studio 2011 – Page default.js

```
(function () {
    "use strict";

    var app = WinJS.Application;

    app.onactivated = function (eventObject) {
        if (eventObject.detail.kind === Windows.ApplicationModel.Activation.ActivationKind.launch)
            if (eventObject.detail.previousExecutionState !==
                Windows.ApplicationModel.Activation.ApplicationExecutionState.terminated)
                // TODO: This application has been newly launched. Initialize
                // your application here.
            } else {
                // TODO: This application has been reactivated from suspension.
                // Restore application state here.
            }
        WinJS.UI.processAll();
    }
};
```

Figure 71 : Visual Studio Express – Page default.js

```
// http://go.microsoft.com/fwlink/?LinkId=232509
(function () {
    "use strict";

    var app = WinJS.Application;
    var activation = Windows.ApplicationModel.Activation;
    WinJS.strictProcessing();

    app.onactivated = function (args) {
        if (args.detail.kind === activation.ActivationKind.launch) {
            if (args.detail.previousExecutionState !== activation.ApplicationExecutionState.terminated)
                // TODO: cette application vient d'être lancée. Initialisez
                // votre application ici.
            } else {
                // TODO: cette application a été réactivée après avoir été suspendue.
                // Restaurez l'état de l'application ici.
            }
            args.setPromise(WinJS.UI.processAll());
        }
    }
};
```

La figure 71 nous montre deux différences.

Le reste du code est identique.

**Figure 72 : Visual Studio 2011 – Page default.js - 2**

```
app.oncheckpoint = function (eventObject) {  
    // TODO: This application is about to be suspended. Save any state  
    // that needs to persist across suspensions here. You might use the  
    // WinJS.Application.sessionState object, which is automatically  
    // saved and restored across suspension. If you need to complete an  
    // asynchronous operation before your application is suspended, call  
    // eventObject.setPromise().  
};  
  
app.start();  
})();
```

**Figure 73 : Visual Studio Express – Page default.js - 2**

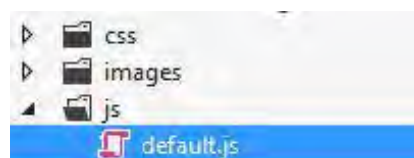
```
app.oncheckpoint = function (args) {  
    // TODO: cette application est sur le point d'être suspendue. Enregistrez tout état  
    // devant être conservé lors des suspensions ici. Vous pouvez utiliser l'objet  
    // WinJS.Application.sessionState, qui est automatiquement  
    // enregistré et restauré en cas de suspension. Si vous devez effectuer une  
    // opération asynchrone avant la suspension de votre application, appelez  
    // args.setPromise().  
};  
  
app.start();  
})();
```

La version de Visual Studio 2011 Beta ne rajoute pas automatiquement les références :

- base.js
- ui.js

L'erreur pourrait provenir de là.

**Figure 74 : Fichier js**



#### 6.1.4 Visual Studio Professional 2012 RC

Suite au mail de M. Jean-Pierre Rey du 22 juin 2012, nous avons cherché des informations sur Visual Studio LightSwitch 11.

Pour pouvoir l'utiliser, nous avons téléchargé Visual Studio Professional 2012 RC.

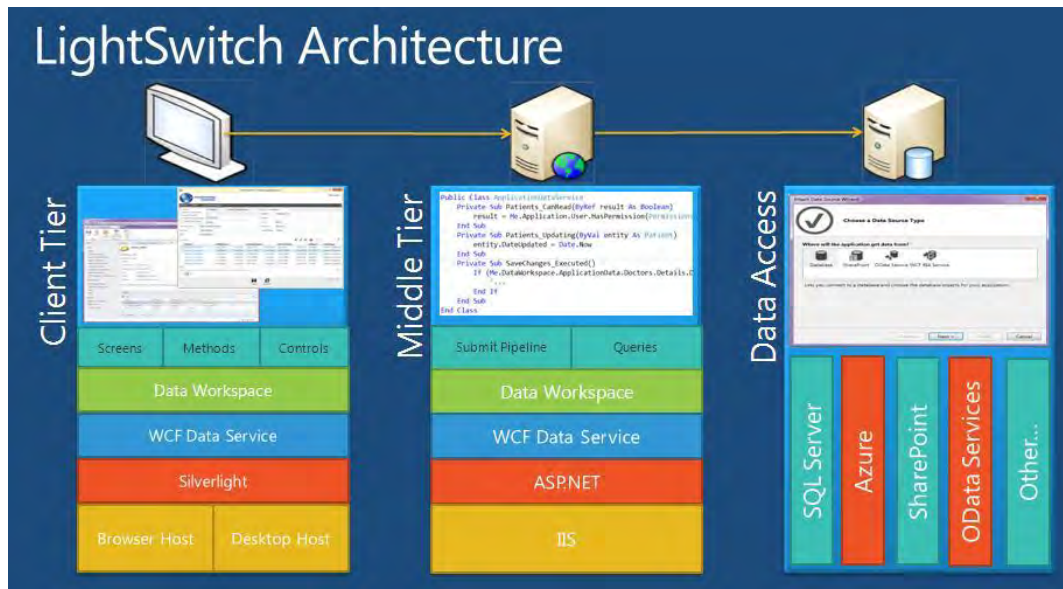
Cet outil nous offre le même menu que Visual Studio 2011 (figure 66).

LightSwitch est un logiciel de développement pour la création d'application à partir d'une base de données et il utilise Silverlight.

Ce genre d'applications peut être considéré comme une application Desktop, car il n'y a pas de possibilité de l'utiliser dans Windows Metro.

Ce logiciel utilise une architecture multicouche.

Figure 75 : LightSwitch – Architecture



Source :

<http://channel9.msdn.com/Events/TechEd/NorthAmerica/2012/DEV348>

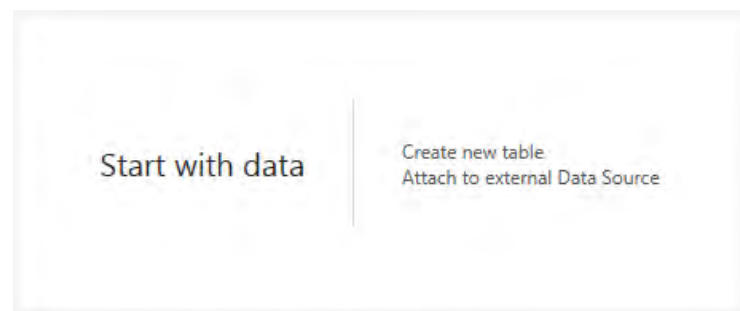
Dans la partie client, nous trouvons les écrans, les méthodes et les contrôles.

Dans la partie application, se trouve les requêtes pour questionner les bases de données

Dans la partie base de données, se trouve les données et les accès aux données.

Lors de la création d'un projet LightSwitch, le développeur a la possibilité d'accéder à une source de données ou de créer une nouvelle base de données (figure 76).

Figure 76 : LightSwitch – Start screen



L'accès à une source externe peut être créé de différentes manières.

Figure 77 : LightSwitch – Attach Data Source

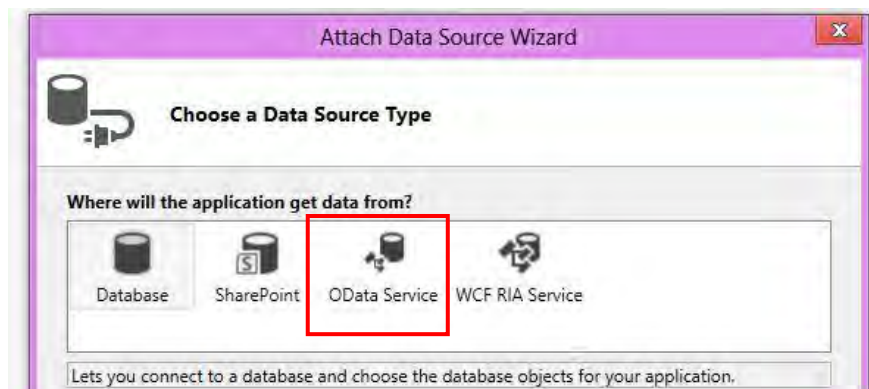
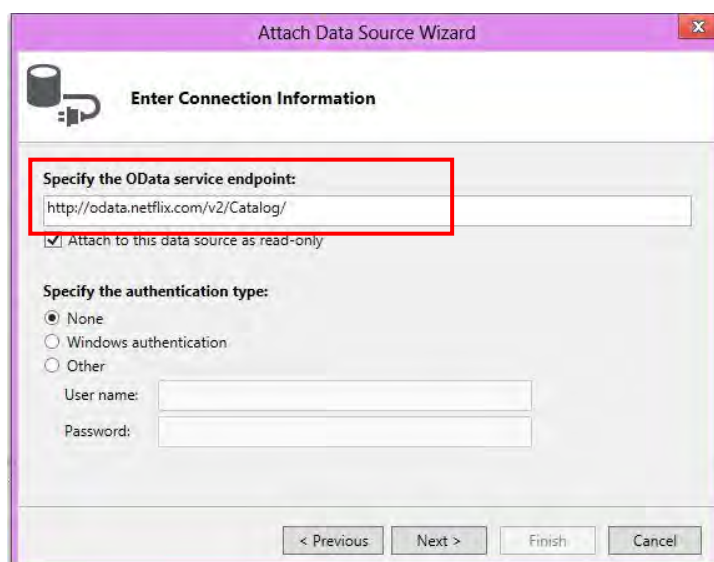


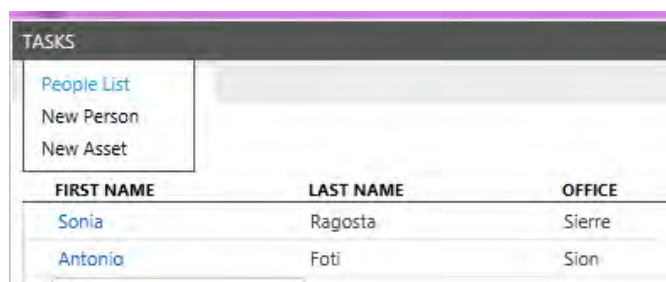
Figure 78 : LightSwitch – OData\_Connection



Si nous choisissons OData, ce protocole nous permet d'accéder facilement à une base de données externe. L'adresse de sa localisation suffit. Ce protocole sera étudié à la fin du chapitre.

Une application LightSwitch basique peut être créée sans toucher le code. L'utilisateur est dirigé pas à pas lors de la création. Les écrans de base sont préconstruits et il obtient rapidement une application fonctionnelle.

**Figure 79 : LightSwitch – UI**



Par contre, des connaissances plus approfondies en .NET sont nécessaires si le code proposé doit être modifié.

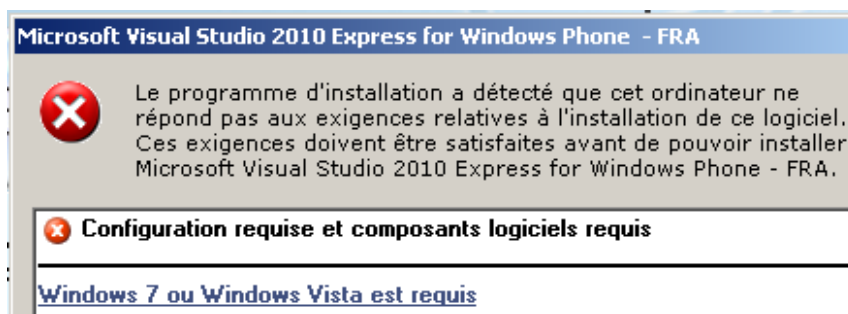
Il en sort que Microsoft a conçu un excellent logiciel pour un développement rapide et simple.

En observant attentivement les deux menus, nous constatons, cependant, que la création d'applications pour Windows Phone n'est pas disponible.

### 6.1.5 Windows Phone

Les recherches effectuées indiquent qu'il existe Visual Studio 2010 Express for Windows Phone, mais cette version n'est pas compatible avec Windows 8.

**Figure 80 : Visual Studio Express for Windows Phone**



Différents forums, dont developpez.com, avertissent que la version 2012 Express ne devrait plus tarder, car Microsoft a présenté le 20 juin 2012 le nouveau Windows Phone 8.

Il est cependant possible de contourner cet obstacle.

Il suffit d'installer :

- Installer Visual Studio 2010
- Installer Visual Studio 2010 SP1
- Installer Games for Windows Marketplace Client (<http://www.xbox.com/en-US/LIVE/PC/DownloadClient>)  
(<http://blogs.msdn.com/b/astebner/archive/2012/02/29/10274694.aspx>)
- Installer Windows Phone SDK 7.1
- Installer Windows Phone SDK 7.1.1 Update

Pour éviter de devoir ouvrir différents outils de programmation, nous utiliserons Visual Studio Professional 2012 RC pour les applications Silverlight et Windows Metro Style, et Visual Studio 2010 pour les applications Windows Phone.

## 6.2 Silverlight / JavaScript : les différences

Pour se rendre compte des différences entre une application Silverlight et une application JavaScript, il a été décidé de prendre un exemple très simple.

### 6.2.1 Interface graphique

A la création de l'interface graphique, nous constatons que les tags sont différents d'un langage à un autre.

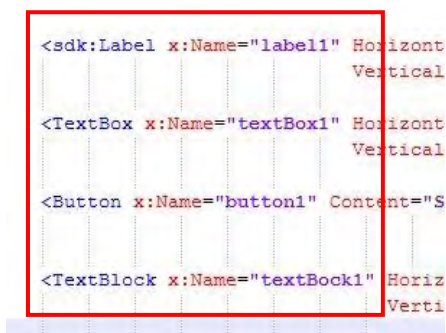
Par exemple pour le bouton, nous utilisons :

- **<Button>** pour Silverlight
- **<button>** pour HTML5

En ce qui concerne la liaison des données, il existe également une différence:

- **x:Name="button1"** pour Silverlight
- **id="helloButton"** pour HTML5

Figure 81 : Silverlight – Code XAML



```
<sdk:Label x:Name="label1" Horizontal
Vertical
<TextBox x:Name="textBox1" Horizontal
Vertical
<Button x:Name="button1" Content="S
<TextBlock x:Name="textBlock1" Horiz
Verti
```



Figure 82 : JavaScript – HTML5

```
<!-- HelloWorld_JS references -->
<link href="/css/default.css" rel="
<script src="/js/default.js"></scri
</head>
<body>
  <h1>Hello World !</h1>
  <p>What's your name ?</p>
  <input id="nameInput" type="text" /
  <button id="helloButton">Say Hello
  <div id="greetingOutput"></div>
</body>
</html>
```

### 6.2.2 Gestion de l'évènement

La gestion de l'évènement est gérée par des méthodes.

Nous constatons que JavaScript récupère les valeurs dans des variables et ensuite il renvoie le résultat en sortie.

Le code-behind de Silverlight est plus simple il utilise directement les valeurs grâce au binding.

Figure 83 : Silverlight – Gestion de l'évènement

```
namespace HelloWorld_SL
{
    public partial class MainPage : UserControl
    {
        public MainPage()
        {
            InitializeComponent();
        }

        private void button1_Click_1(object sender, RoutedEventArgs e)
        {
            textBock1.Text = "Hello " + textBox1.Text + "!";
        }
    }
}
```

Figure 84 : JavaScript – Gestion de l'évènement

```
//Gestion de l'évènement bouton

function buttonClickHandler(eventInfo) {
    var userName = document.getElementById("nameInput").value;
    var greetinString = "Hello, " + userName + "!";

    document.getElementById("greetingOutput").innerText = greetinString;
}
```

### 6.2.3 Affichage

JavaScript propose une interface de base de couleur noire. Nous pouvons la modifier en nous rendant dans la page default.css.

Tandis que Silverlight nous offre un affichage plus sobre de couleur blanc. Il peut être modifié par la feuille de style : Styles.xaml.

Figure 85 : Silverlight - Screen

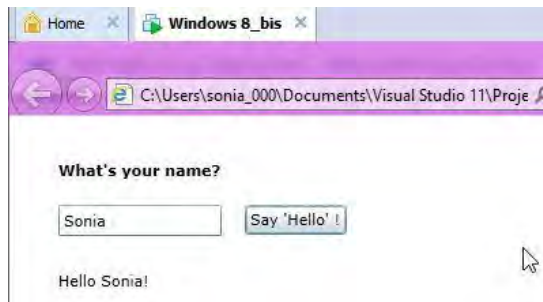


Figure 86 : JavaScript - Screen



## 6.3 Peut-on passer facilement les applications d'un langage à l'autre?

Les exemples sont simples mais les différences sont importantes. Les tags des interfaces graphiques sont différents et même la déclaration des méthodes.

Microsoft n'offre aucune solution miracle pour passer les applications existantes en applications Windows Metro apps.

Pour les applications en Silverlight de Windows Phone 7, il nous encourage à reprendre le code existant et de modifier les tags et les API (figure 87).



**Figure 87 : Modification des APIs**

API type	Silverlight API namespaces	Windows Runtime API namespaces (C#/Visual Basic)
Devices	Microsoft.Devices, Microsoft.Devices.Radio, Microsoft.Devices.Sensors	Windows.Devices.Enumeration, Windows.Devices.Enumeration.Pnp, Windows.Devices.Input, Windows.Devices.Sensors
Windows Phone app model and environment	Microsoft.Phone, Microsoft.Phone.Info, Microsoft.Phone.Notification, Microsoft.Phone.Reactive, Microsoft.Phone.Shell, Microsoft.Phone.Tasks	Windows.Networking.PushNotifications, Windows.Devices.Sms, Windows.ApplicationModel.Background, Windows.ApplicationModel.Contacts, Windows.ApplicationModel.Contacts.Provider, Windows.ApplicationModel.Core
Maps	Microsoft.Phone.Controls.Maps, Microsoft.Phone.Controls.Maps.AutomationPeers, Microsoft.Phone.Controls.Maps.Core, Microsoft.Phone.Controls.Maps.Design, Microsoft.Phone.Controls.Maps.Overlays, Microsoft.Phone.Controls.Maps.Platform	Windows.Devices.Geolocation

Source : <http://msdn.microsoft.com/en-us/library/windows/apps/hh465128.aspx>

Pour un développeur XAML/C#, il est possible de reprendre une application existante et la passer sur Metro Style Apps. L'environnement de travail est le même, mais il devra de toute façon reprendre le code existant et le modifier.

Actuellement, le problème ne se pose pas pour les applications existantes. Elles resteront des applications desktop.

Durant les TechDays 2012, Microsoft nous propose sa nouvelle vision de l'architecture des applications.

Avant de débiter la programmation de l'application, il est important d'étudier le besoin métier de chaque entreprise et ses contraintes.

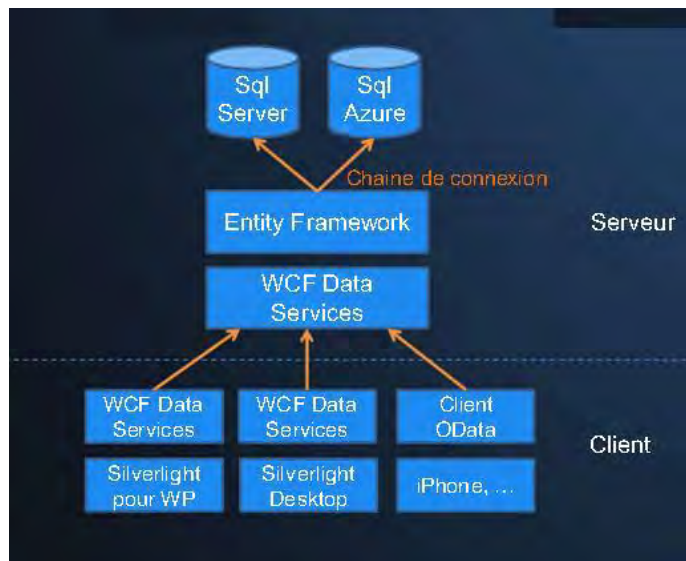
De nos jours, une application devrait être multiplateforme et adaptée à chaque dispositif, que ce soit le téléphone, la tablette ou l'écran d'un ordinateur.

Une solution serait de développer des applications en langage natif pour chaque plateforme. Ils pourraient ainsi utiliser Silverlight pour les applications de Windows Phone ou Objectif C pour iPhone.

Les applications seront ainsi différentes selon la plateforme, mais les données resteront identiques.

La figure 88 montre différents clients qui, grâce aux web services ou OData, accèdent aux mêmes données.

Figure 88 : Architectures

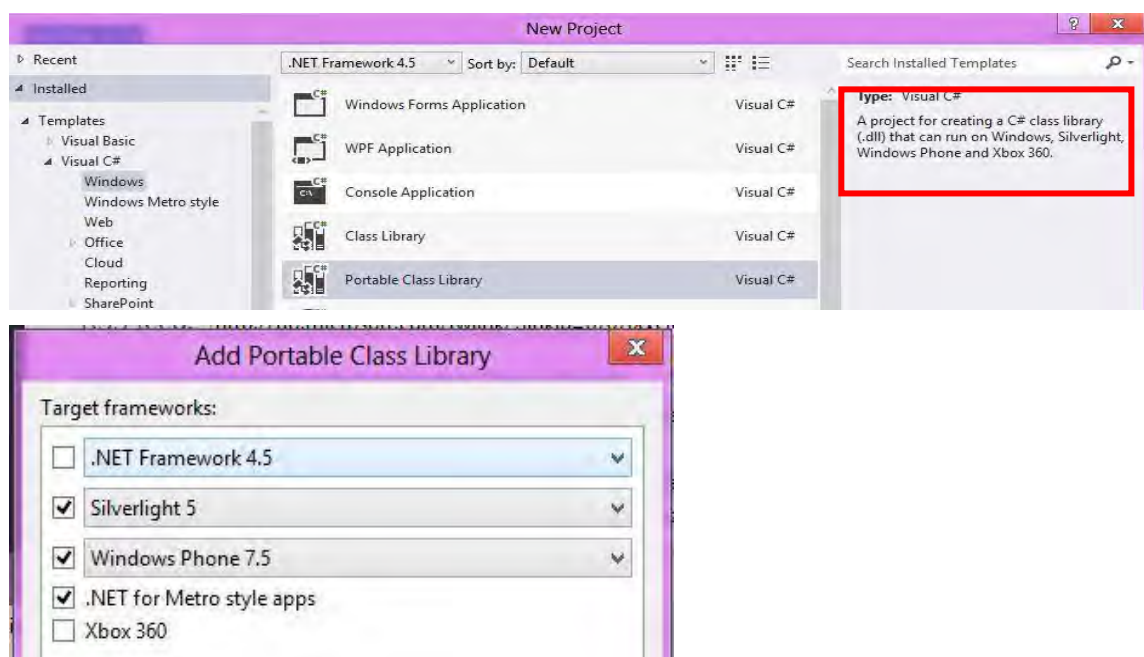


Source : <http://www.microsoft.com/france/mstechdays/programmes/parcours.aspx#DomID=a1b85459-4eaa-471a-8927-1be7c76fdbb2&SessionID=66174549-97b8-404e-89e8-7f9bfe6fc74e>

Microsoft attaque aujourd'hui le marché des applications mobiles. Il met à disposition des développeurs tous les outils et les langages nécessaires pour recréer et non modifier l'existant. Chaque dispositif aura ainsi son application propre.

Visual Studio permet la création d'un projet "Portable Class Library". Il aide le partage de code entre les différentes plateformes Silverlight, Windows Metro Style, Windows Phone ou Xbox.

Figure 89 : Multiplateformes



L'utilisation de l'architecture MVVM et la création d'une application Windows « Portable Class Library » rend ce partage simple et rapide.

L'application « Portable Class Library » contiendra le code commun à toutes les autres applications et elle sera ajoutée comme référence.

**Figure 90 : Portable Class Library – Code commun**

Portable Class Library	DAL		ViewModel		
	IWine	ICaveAVinsDataProvider	ViewModelBase	WineVM	CaveAVinsVM
Win8		CaveAVinsDataProvider		Win8WineVM	
WP		CaveAVinsDataProvider	idem Portable Class Library		
SL		CaveAVinsDataProvider		SLWineVM	

L'accès aux données (DAL) est réalisé grâce aux web services ou OData.

Donnons quelques informations sur OData.

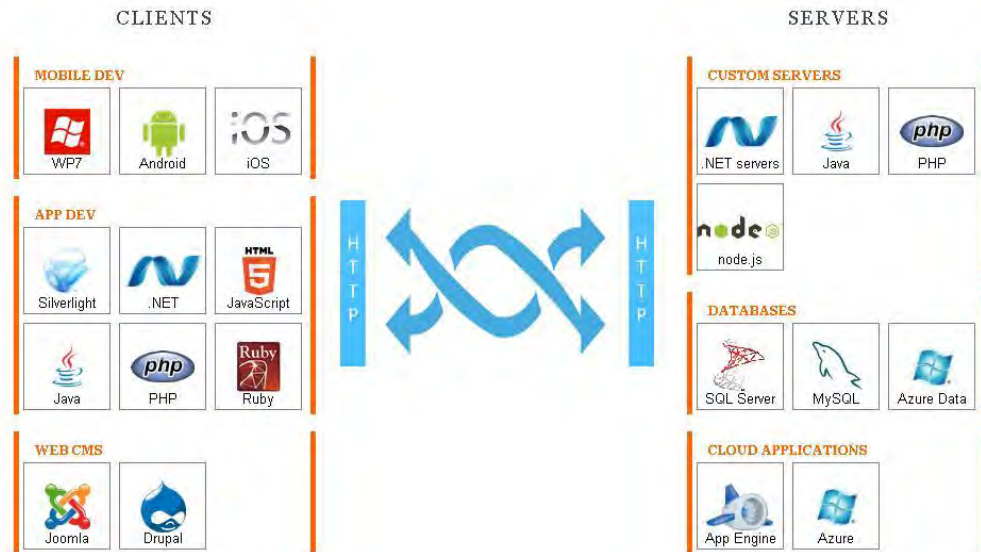
## 6.4 OData

OData, **O**pen **D**ata **P**rotocol, est un protocole web ouvert de Microsoft. Son but est d'exposer et de partager des données à travers internet. Ce protocole n'est pas spécifique à Microsoft, mais il peut être utilisé par tous les clients supportant HTTP. Il utilise une architecture REST (**R**epresentational **S**tate **T**ransfer) et ses principes :

- URI : pour l'identification de l'objet
- HTTP : pour le traitement des données (PUT – GET – POST – DELETE)
- HTML / XML / JSON : pour la navigation et la liaison vers d'autres ressources

Il propose pour chaque plateforme un SDK (**S**oftware **D**evelopment **K**it) spécifique. Il s'agit de bibliothèque pour les langages comme PHP, JavaScript ou Silverlight.

**Figure 91 : OData - SDK**



Source : <http://www.odata.org/libraries>

Si le SDK n'existe pas, il est facile de construire une demande grâce à l'URI. L'URI nous permet également de modifier, supprimer, trier ou filtrer les données.

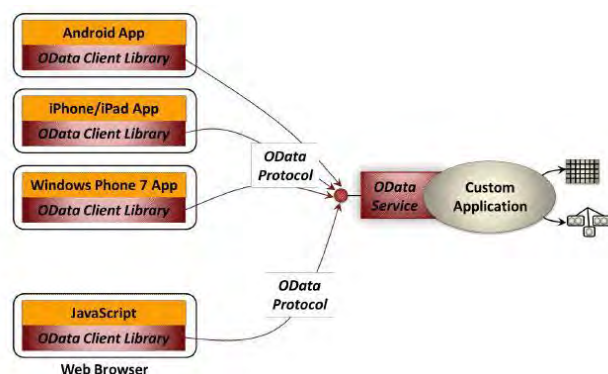
Par exemple :

`http://localhost:59404/WcfDataService1.svc/Employees?$filter=Firstname eq 'Bill'`

Cette requête permet de filtrer toutes les personnes qui se prénomme Bill.

OData est un protocole facile à implémenter. Cependant il est important de préciser que tous les échanges de données se font au niveau du http, ce qui ne garantit pas leur sécurité.

**Figure 92 : OData - Multiplateforme**

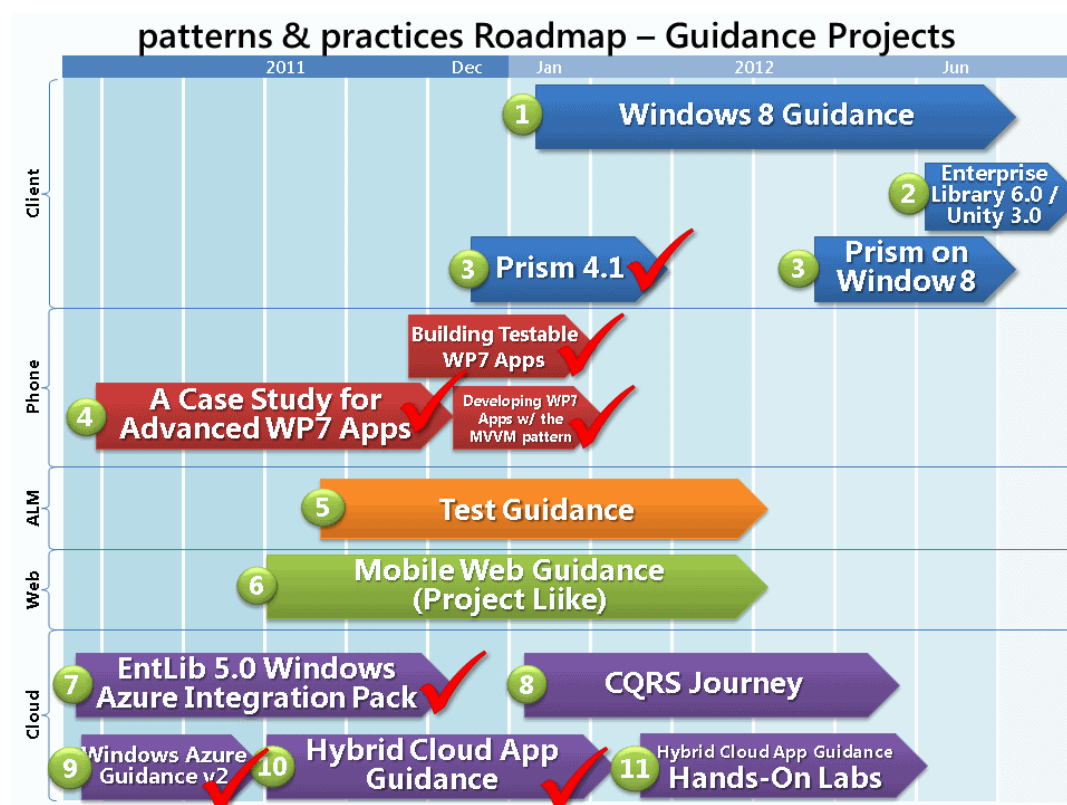


Source : David Chappell – Introducing OData.pdf

## 7 Planification de Microsoft

Microsoft donne sur son site une feuille de route des tâches planifiées pour le développement de ses produits.

Figure 93 : Roadmap



Source : [http://msdn.microsoft.com/en-us/practices/bb190332.PnP-FY12-Roadmap-201203-v4\(l=en-us\).png](http://msdn.microsoft.com/en-us/practices/bb190332.PnP-FY12-Roadmap-201203-v4(l=en-us).png)

### 7.1 Windows 8 Guidance

Bien que Steven Sinofski ait annoncé officiellement<sup>9</sup> que Windows 8 sera en vente dès le 26 octobre 2012, ce logiciel reste une partie de développement planifiée à long terme. Immersive Windows 8, le nom utilisé par Microsoft pour définir les applications Metro Style, continuera à être amélioré au retour des différentes remarques reçues par les utilisateurs.

Windows 8 offre à ses utilisateurs une nouvelle interface orientée tactile. Elle sera identique sur tous les dispositifs qu'elle présentera au grand public. L'utilisateur ne se sentira pas dépaycé. Il passera d'un dispositif à un autre d'une manière tout à fait naturelle.

Pendant cette phase, les programmeurs de Microsoft vont créer des applications et tester le binding, la sécurité, le stockage des données en local par exemple. Ils utiliseront le pattern MVVM.

<sup>9</sup> <http://blogs.msdn.com/b/b8/archive/2012/08/01/releasing-windows-8-august-1-2012.aspx>

Une enquête du centre de recherche Gartner<sup>10</sup> d'avril 2012, indique que la vente des tablettes a augmenté de 98% par rapport à 2011.

Bien qu'Apple soit toujours en tête, le tableau, ci-dessous, nous montre que Microsoft a décidé de s'attaquer à ce nouveau marché et de récupérer son retard. L'apparition de Microsoft sur le marché coïncide avec la présentation de Windows 8. Cette augmentation serait due, selon Gartner, au fait que l'utilisateur choisira ce système car l'interface sera la même que celle de son poste fixe.

**Figure 94 : Vente Tablettes**

**Table 1**  
**Worldwide Sales of Media Tablets to End Users by OS (Thousands of Units)**

OS	2011	2012	2013	2016
iOS	39,998	72,988	99,553	169,652
Android	17,292	37,878	61,684	137,657
Microsoft	0	4,863	14,547	43,648
QNX	807	2,643	6,036	17,836
Other Operating Systems	1,919	510	637	464
<b>Total Market</b>	<b>60,017</b>	<b>118,883</b>	<b>182,457</b>	<b>369,258</b>

Source : <http://www.gartner.com/it/page.jsp?id=1980115>

Gartner prévoit que, de 2012 à 2013, les ventes devraient tripler et que cette tendance devrait continuer.

## 7.2 Mobile web Guidance

Un autre élément planifié à long terme est le développement d'application mobile.

A ce sujet, Microsoft distingue différentes catégories :

- Les applications natives

Les applications natives sont des applications mobiles écrites dans le langage spécifique de la plateforme. Elles utilisent les ressources de la machine.

Ainsi pour le Windows Phone, elles seront écrites en XAML et C# et pour iPhone elles utiliseront Objective-C.

Les avantages sont :

- Accès aux fonctionnalités du système d'exploitation
- Grandes richesses des interfaces utilisateurs
- Les applications sont présentes dans des magasins spécifiques. Elles sont ainsi faciles à trouver et à installer.
- Les mises à jour sont faites directement par la machine.

---

<sup>10</sup> <http://www.gartner.com/it/page.jsp?id=1980115>



- Les applications web

Les applications web utilisent les standards. Elles sont écrites en CSS, HTML5 et JavaScript. Elles sont ensuite envoyées d'un serveur web vers internet.

Les avantages sont :

- Elles partagent le navigateur comme environnement d'exécution
- Aucune nécessité d'installer le navigateur, il est préinstallé
- Langage connu par de nombreux développeurs
- Possibilité de réutiliser des parties de code
- Aucune installation n'est nécessaire, ni de gérer les mises à jour

- Les applications hybrides

Les applications hybrides utilisent les technologies natives et celles du web.

Egalement pour cette phase, Microsoft propose une ligne directrice des tâches à exécuter.

Les développeurs créeront des applications web interactives orientées cloud. Pour cela, ils comptent utiliser HTML5.

Les sous-tâches sont également nombreuses. Ils devront tester la sécurité, le pattern MVC, le tactile et l'accès aux données.

Actuellement la vente des Windows Phone est en chute. En effet, le diagramme ci-dessous montre une baisse de 0.7% entre le dernier trimestre 2011 et le premier trimestre 2012.

**Figure 95 : Vente smartphone**

**Table 2**  
**Worldwide Smartphone Sales to End Users by Operating System in 1Q12**  
**(Thousands of Units)**

Operating System	1Q12		1Q11	
	Units	Market Share (%)	Units	Market Share (%)
Android	81,067.4	56.1	36,350.1	36.4
iOS	33,120.5	22.9	16,883.2	16.9
Symbian	12,466.9	8.6	27,598.5	27.7
Research In Motion	9,939.3	6.9	13,004.0	13.0
Bada	3,842.2	2.7	1,862.2	1.9
Microsoft	2,712.5	1.9	2,582.1	2.6
Others	1,242.9	0.9	1,495.0	1.5
<b>Total</b>	<b>144,391.7</b>	<b>100.099</b>	<b>775.0</b>	<b>100.0</b>

Source: Gartner (May 2012)

Source : <http://www.gartner.com/it/page.jsp?id=2017015>

## 8 Conclusion

Cette année 2012 offre au monde des utilisateurs Microsoft de nombreuses nouveautés. Le site de Microsoft présente chaque jour des nouvelles, des mises à jour ou des coups d'éclats.

Tout au long de ce rapport, Microsoft nous a donné sa nouvelle vision. Cette entreprise d'envergure mondiale a décidé de diversifier ses stratégies. Elle souhaite se lancer dans le développement d'application mobile pour rattraper le retard tout en continuant à développer le Desktop et Windows Phone.

Dans le communiqué<sup>11</sup> du 8 juin 2012, M. Somasegar, vice-président de la division de développement de Microsoft, précise :

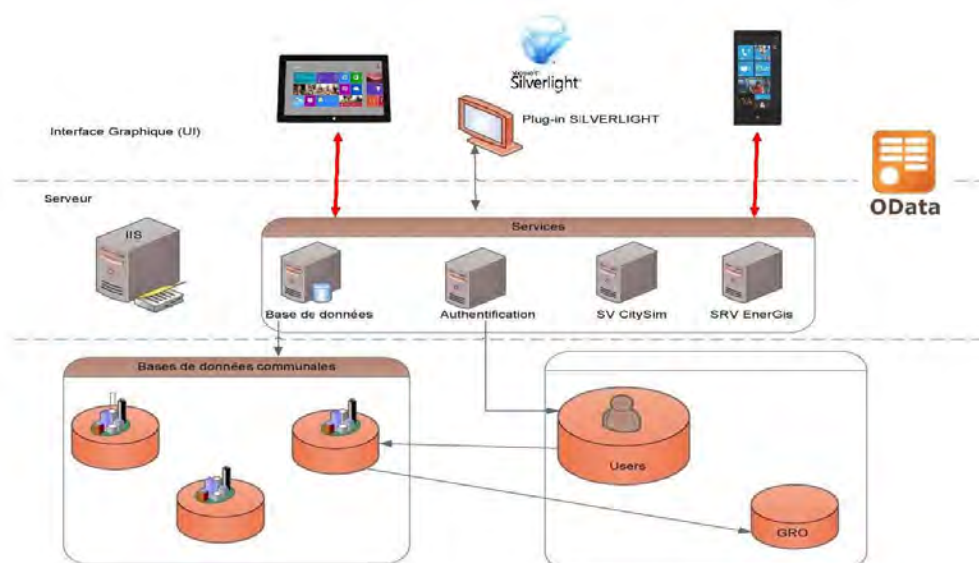
"Desktop development has always been a core part of Windows. With Visual Studio 2012, we continue to extend those desktop development capabilities and provide a great development experience for developers building desktop applications."

L'évolution de l'Interface Homme – Machine de la plateforme du projet MEU suivra son cours naturel. Les développeurs n'ont pas la nécessité d'envisager de l'abandonner et de passer à un autre langage, car le plug-in Silverlight 5 n'est pas prêt à mourir.

En effet, ils annoncent des améliorations de ce plug-in lors de la mise en service de Windows 8 Ultimate.

Par contre, le projet MEU pourra être amélioré en appliquant les nouveautés que propose Silverlight 5 et, peut-être, il serait intéressant d'envisager de porter cette application sur tablette.

Figure 96 : Futur du projet MEU



<sup>11</sup> <http://blogs.msdn.com/b/visualstudio/archive/2012/06/08/visual-studio-express-2012-for-windows-desktop.aspx>



## 9 Attestation

---

Je déclare par ce document, que j'ai effectué le travail de bachelor ci-annexé seule, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après.

Sonia Ragosta Foti

## 10 Références

---

### Chapitre 1 : Introduction

Analyze and model your world. What is GIS?

Consulté le 19 juillet 2012, disponible sur :

<http://www.esri.com/what-is-gis/index.html>

Management Energétique Urbain – MEU. *Outils innovants pour la planification et le management de systèmes énergétiques en zones urbaines. Rapport final.*

Consulté le 23 mars 2012, reçu par mail de M. Rey

MEU 2012 (GC). The Ark Energy. *Descriptif de projet fonds The Ark.*

Consulté le 23 mars 2012, reçu par mail de M. Rey

### Chapitre 2 : Compréhension du fonctionnement de Silverlight

Arcorama (2010). *Flex vs. Silverlight*

- Concepts et historiques.

Consulté le 22 avril 2012, disponible sur :

<http://www.arcorama.fr/2010/02/flex-vs-silverlight-concepts-et.html>

- Technologies et exemples concrets

Consulté le 26 avril 2012, disponible sur :

<http://blog.cozic.fr/que-sont-les-applications-riches-ria-partie-2-technologies-et-exemples-concrets-199>

- Quel futur pour l'Internet riche ?

Consulté le 26 avril 2012, disponible sur :

<http://blog.cozic.fr/que-sont-les-applications-riches-ria-partie-3-quel-futur-pour-linternet-riche-209>

Centre de développement Silverlight DC. (2012). *Dissection des nouveautés de Silverlight 5 beta 1.*

Consulté le 10 avril 2012, disponible sur :

<http://msdn.microsoft.com/fr-fr/silverlight/hh184212>

Centre de développement Silverlight. *Coach Silverlight 3. Atelier 1 à 11.*

Consulté le 3 avril 2012

<http://msdn.microsoft.com/fr-fr/silverlight/msdn.coachsilverlight>

Centre de développement Silverlight. *Webcasts Silverlight.*

Consulté le 3 avril 2012, disponible sur :

<http://msdn.microsoft.com/fr-fr/silverlight/ee851164>

Dewit O (2011), *Silverlight, Développez des applications riches universelles.* St-Herblain : Editions ENI

macromedia white paper JA. (March 2002). *Macromedia Flash MX-A next generation rich client*.

Consulté le 26 avril 2012, disponible sur

<http://download.macromedia.com/pub/flash/whitepapers/richclient.pdf>

Microsoft Application Architecture Guide, S. Somasegar. (2009). *Pattern & Practices*, 2<sup>nd</sup> Edition. (Version PDF)

mrwebmaster VG (2010). *Introduzione a Silverlight*

Consulté le 20 avril 2012, disponible sur :

[http://www.mrwebmaster.it/aspnet/articoli/introduzione-silverlight\\_1216.html](http://www.mrwebmaster.it/aspnet/articoli/introduzione-silverlight_1216.html)

VB & .Net MDG. (2012). *La nuova versione di Silverlight 5 ancora piu luminosa*.

Consulté le 27 avril 2012, disponible sur :

<http://deghetto.wordpress.com/2012/03/20/la-nuova-versione-di-silverlight-5-ancora-pi-luminosa-2/>

### **Chapitre 3 : Compréhension de l'architecture de développement de l'application existante**

CodePlex TS (2010). *Simple MVVM Toolkit for Silverlight, WPF and Windows Phone*.

Consulté le 6 juillet 2012, disponible sur :

<http://simplemvvmtoolkit.codeplex.com/wikipage?title=Introduction>

CodePlex RE (2011). *Caliburn.Micro*.

Consulté le 6 juillet 2012, disponible sur :

<http://caliburnmicro.codeplex.com/wikipage?title=Introduction&referringTitle=Documentation>

Développez.com NC (2010). *Tutoriel : Utilisez le pattern Model-View-ViewModel dans vos applications Silverlight grâce à MVVM Light toolkit*.

Consulté le 4 mai 2012, disponible sur :

<http://nico-pyright.developpez.com/tutoriel/vs2010/csharp/silverlight/introduction-mvvm-toolkit-silverlight/>

e-n@axos OD (2010). *M-V-VM et Silverlight – De la théorie à la pratique*.

Consulté le 9 mai 2012, disponible sur :

<http://www.e-naxos.com/dnlManager.aspx>

GalaSoft LB (2007-2011). *MVVM Light Toolkit*.

Consulté le 14 mai 2012, disponible sur :

<http://www.galasoft.ch/mvvm/>

MVVM Frameworks Explorer by www.japf.fr

Consulté le 15 mai 2012, disponible sur :

<http://www.japf.fr/silverlight/mvvm/index.html>

Tales from Rich Client JG (2005). *Introduction to Model/View/ViewModel pattern for building WPF apps.*

Consulté le 11 mai 2012, disponible sur :

<http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx>

TechDays 2012 AA & TL. *MVVM de A à Z.*

Consulté le 11 mai 2012, disponible sur :

[http://www.youtube.com/watch?v=PN\\_PYTnGIHA](http://www.youtube.com/watch?v=PN_PYTnGIHA)

The code project SK (2010). *One Way, Two Way and One Time Bindings using Silverlight.*

Consulté le 14 mai 2012, disponible sur :

<http://www.codeproject.com/Articles/37250/One-Way-Two-Way-and-One-Time-Bindings-using-Silver>

Università degli studi di Trieste AA (2009). *Separazione dei ruoli tra Designer e Developer nello sviluppo delle applicazione Desktop : uso di WPF e del design pattern MVVM.*

Consulté le 4 juin 2012, disponible sur :

<http://www.slideshare.net/alessandroandreose/tesi-laurea-specialistica-ingegneria-informatica-alessandro-andreos>

#### **Chapitre 4: Positionnement de Microsoft face à Silverlight et HTML5**

alsacrations, communauté d'apprentissage pour les standards du web J (2012). *HTML5 se dévoile.*

Consulté le 25 mai 2012, disponible sur :

[www.alsacrations.com/article/lire/750-HTML5-nouveautes.html](http://www.alsacrations.com/article/lire/750-HTML5-nouveautes.html)

Coding Horror JA (2008). *Understanding Model-View-Controller.*

Consulté le 28 mai 2012, disponible sur :

<http://www.codinghorror.com/blog/2008/05/understanding-model-view-controller.html>

Developpez.com TT (2007). *Programmation orientée objet avec le langage JavaScript (1<sup>ère</sup> partie).*

Consulté, le 28 mai 2012, disponible sur :

<http://t-templier.developpez.com/tutoriel/javascript/javascript-pool1/>

HTML5-CSS3.fr V (2011). *Tutoriels et démonstrations HTML5 et CSS3.*

Consulté le 25 mai 2012, disponible sur :

<http://www.html5-css3.fr/mot-cle/slides>

HTML5 RR (2012). *Une référence pour le développeur web.* Editions Eyrolles. Paris.

learn.knockoutjs.com. *Tutorial MVVM with HTML5.*

Consulté le 27 mai 2012, disponible sur :

<http://learn.knockoutjs.com/#/?tutorial=intro>

Le Journal du Net OP (2012). *Le développement multi-plateforme : enjeux, promesses et réalité.*

Consulté le 29 mai 2012, disponible sur :

<http://www.journaldunet.com/developpeur/expert/51048/le-developpement-multi-plateforme---enjeux--promesses-et-realite.shtml>

Le Site du zéro MN (2011). *Apprenez à créer votre site web avec HTML5 et CSS3.*

Consulté le 24 mai 2012, disponible sur :

<http://www.siteduzero.com/tutoriel-3-13666-apprenez-a-creer-votre-site-web-avec-html5-et-css3.html>

Microsoft Web Camps GG (2011). *Discutiamo insieme di Open Web Standards, HTML e futuro del web.*

Consulté le 27 mai 2012, disponible sur :

<http://www.microsoft.com/italy/webcamps/online/default.aspx>

Microsoft Tech Days 2012. *Diverses vidéos.*

Consulté le 27 avril 2012, disponible sur :

<http://www.microsoft.com/france/mstechdays/#&fbid=yz-RKzGi-wI>

## **Chapitre 5: Metro Style Apps**

.toolbox courses | tutorials (2012). *Metro Design Language of Windows Phone 7.*

Consulté le 5 juin 2012, disponible sur :

[www.microsoft.com/design/toolbox/tutorials/windows-phone-7/metro/](http://www.microsoft.com/design/toolbox/tutorials/windows-phone-7/metro/)

Channel 9 ML (2012). *Martin Lowell : The Windows Runtime. (Vidéo)*

Consulté le 11 juin 2012, disponible sur :

<http://channel9.msdn.com/Events/Lang-NEXT/Lang-NEXT-2012/The-Windows-Runtime>

developpez.com HR (2012). *Windows 8 : Microsoft veut supporter plus de langage dans WinRT.*

Consulté le 6 juin 2012, disponible sur :

<http://www.developpez.net/forums/d1205707/dotnet/general-dotnet/windows-8-microsoft-veut-supporter-plus-langages-winrt/>

Kishore – Software Architect, Washington DC KG (2011). *Part 2 : Getting started with using new Windows 8 features.*

Consulté le 6 juin 2012, disponible sur :

<http://kishore1021.wordpress.com/2011/12/28/part-2-getting-started-with-using-new-windows-8-features/>

techopedia. *Windows Runtime Library (WinRT)*

Consulté le 6 juin 2012, disponible sur :

<http://www.techopedia.com/definition/27608/windows-runtime-library-winrt-windows-8>

The code project KG (2012). *Part 3 : Introduction to WinRT, the new "Windows Runtime" in Windows 8.*

Consulté le 6 juin 2012, disponible sur :

<http://www.codeproject.com/Articles/328551/Part-3-Introduction-to-WinRT-the-new-Windows-Runti>

Wikipedia L'enciclopedia libera (2012). *Metro (Interfaccia grafica).*

Consulté le 5 juin 2012, disponible sur :

[http://it.wikipedia.org/wiki/Metro\\_\(interfaccia\\_grafica\)](http://it.wikipedia.org/wiki/Metro_(interfaccia_grafica))

WinRTItalia.com AL (2012). *Analisi dell'evoluzione di Metro : da Windows Phone à Windows 8.*

Consulté le 2 juin 2012, disponible sur :

[www.winrtitalia.com/articoli/windows8/evoluzione-metro-windows-phone-windows8.aspx](http://www.winrtitalia.com/articoli/windows8/evoluzione-metro-windows-phone-windows8.aspx)

WinRTItalia.com CC (2012). *Sviluppare Metro style app pre Windows 8 in WinRT con Visual Studio 11.*

Consulté le 5 juin 2012, disponible sur :

[www.winrtitalia.com/articoli/windows8/sviluppare-metro-style-app-winrt-visual-studio.aspx](http://www.winrtitalia.com/articoli/windows8/sviluppare-metro-style-app-winrt-visual-studio.aspx)

Windows. Dev Center – Metro style apps (2012). *What's Metro Style app?*

Consulté le 6 juin 2012, disponible sur :

[http://msdn.microsoft.com/en-us/library/windows/apps/hh974576\(d=printer\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/hh974576(d=printer).aspx)

ZDNet.fr (2012). *Le Windows Store ouvre ses portes dans 38 pays.*

Consulté le 26 juillet 2012, disponible sur :

<http://www.zdnet.fr/actualites/le-windows-store-ouvre-ses-portes-virtuelles-dans-38-pays-39770915.htm>

## **Chapitre 6: Applications**

Aaron Stebner's WebLog AS (2012). *How to work around a possible XNA Game Studio or Windows Phone SDK install failure on Windows 8.*

Consulté le 23 juillet 2012, disponible sur :

<http://blogs.msdn.com/b/astebner/archive/2012/02/29/10274694.aspx>

Introducing OData DC (2011). *Data access for the web, the cloud, modele devices, and more.*

Consulté le 30 juillet 2012, disponible sur :

<http://pdfflare.com/index.php?s=odata&page=1>

MSDN-Blogs SH (2012). *Partager du code entre plateforme grâce à la Portable Class Library.*

Consulté le 3 juillet 2012, disponible sur :

<http://blogs.msdn.com/b/stephe/archive/2012/05/07/partager-du-code-entre-plateformes-gr-226-ce-224-la-portable-class-library-1-4.aspx>

Tech Days SH (2012). *De A à Z. Choisir une architecture pour sa solution applicative*. Consulté le 11 mai 2012, disponible sur :  
<http://www.microsoft.com/france/mstechdays/programmes/parcours.aspx#DomID=a1b85459-4eaa-471a-8927-1be7c76fdbb2&SessionID=66174549-97b8-404e-89e8-7f9bfe6fc74e>

Tech Ed JS (2012). *Building Business Applications in LightSwitch*. Consulté le 30 juin 2012, disponible sur :  
<http://channel9.msdn.com/Events/TechEd/NorthAmerica/2012/DEV348>

Windows. Dev Center – Metro style apps (2012). *Migrating a web app (Metro style apps using JavaScript and HTML)*. Consulté le 18 juin 2012, disponible sur :  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh465143.aspx>

Windows. Dev Center – Metro style apps (2012). *Migrating a web site (Metro style apps using JavaScript and HTML)*. Consulté le 18 juin 2012, disponible sur :  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh465155.aspx>

Windows. Dev Center – Metro style apps (2012). *Migrating a Windows Phone 7 app (Metro style apps)*. Consulté le 18 juin 2012, disponible sur :  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh452743.aspx>

Windows. Dev Center – Metro style apps (2012). *Migrating a Windows Phone 7 app to JavaScript*. Consulté le 19 juin 2012, disponible sur :  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh465128.aspx>

## **Chapitre 7 : Planification de Microsoft**

Gartner CP (2012). *Gartner says Worldwide Media Tablets Sales to Reach 119 Million Units in 2012*. Consulté le 26 juillet 2012, disponible sur :  
<http://www.gartner.com/it/page.jsp?id=1980115>

Microsoft, patterns & devices (2012). *Choosing between a web and native experience*. Consulté le 20 juillet 2012, disponible sur :  
<http://msdn.microsoft.com/en-us/library/jj149679.aspx>

Numerama GC (2012). *Windows 8 sera lancé le 26 octobre*. Consulté le 26 juillet 2012, disponible sur :  
<http://www.numerama.com/magazine/23215-windows-8-sera-lance-le-26-octobre.html>

patterns & practices (2012). *Upcoming Release Roadmap*. Consulté le 26 juillet 2012, disponible sur :  
<http://msdn.microsoft.com/en-us/practices/bb232643>



## **Chapitre 8 : Conclusion**

The Visual Studio Blog SS (2012). Visual Studio Express 2012 for Windows Desktop. Consulté le 27 juin 2012, disponible sur :

<http://blogs.msdn.com/b/visualstudio/archive/2012/06/08/visual-studio-express-2012-for-windows-desktop.aspx>

## 11 Annexes

### 11.1 Support de Microsoft

<http://support.microsoft.com/lifecycle/?c2=12905>

#### Aide et Support Microsoft

[Accueil](#) [Support produits](#) [Rechercher](#) [Télécharger](#) [Contacts](#)

[Partager](#)

#### Politique de Support Microsoft

##### Politique de Support Microsoft

##### Informations sur les dates de début et de fin de support par familles de produits

Le support est assuré par Microsoft pour le service pack en cours, et dans certains cas le service pack précédent, pour les produits listés ci-dessous. Voir le [forum aux questions sur la politique de support](#) pour plus d'informations sur la politique de support des Service Packs. Pour toute question concernant la disponibilité du support d'un produit en particulier, veuillez contacter votre représentant Microsoft. Si vous avez besoin d'un support technique, veuillez visiter la page [contacter Microsoft](#).

Les informations contenues dans cette page sont susceptibles d'évaluer et d'être modifiées. Veuillez consulter ce site de façon régulière pour prendre connaissance des évolutions et mises à jour.

Produit commercialisé	Date initiale de politique de support *	Date de fin de la phase principale de support	Date de fin de la phase d'extension de support	Date de retrait du Service Pack	Remarques
Silverlight 2	10.10.2008	12.10.2010	Non applicable		
Silverlight 3	09.07.2009	12.04.2011	Non applicable		
Silverlight 4	15.04.2010	08.01.2013	Non applicable		
Silverlight 5	09.12.2011	12.10.2021	Non applicable		

[Contactez-nous](#)

[Conditions d'utilisation \(anglais\)](#)

[Marques](#)

[Confidentialité et cookies](#)

[Contrat de prestations](#)

**Microsoft**

©2012 Copyright

## 11.2 Applications

### 11.2.1 Pivot Viewer - Silverlight

#### Pivot Viewer Sample – Silverlight 5

---

Création d'un projet Silverlight : PivotViewer\_SL

Dans MainPage.xaml.cs, création de l'objet. L'objet est défini par un nom, une description et une date de création.

```
public class Item
{
    public string Name { get; set; }
    public string Description { get; set; }
    public DateTime CreationDate { get; set; }
}
```

Création d'une collection d'objet qui va nous aider à tester cette fonctionnalité de Silverlight 5.

Les objets sont créés dans un ordre quelconque.

```
InitializeComponent();

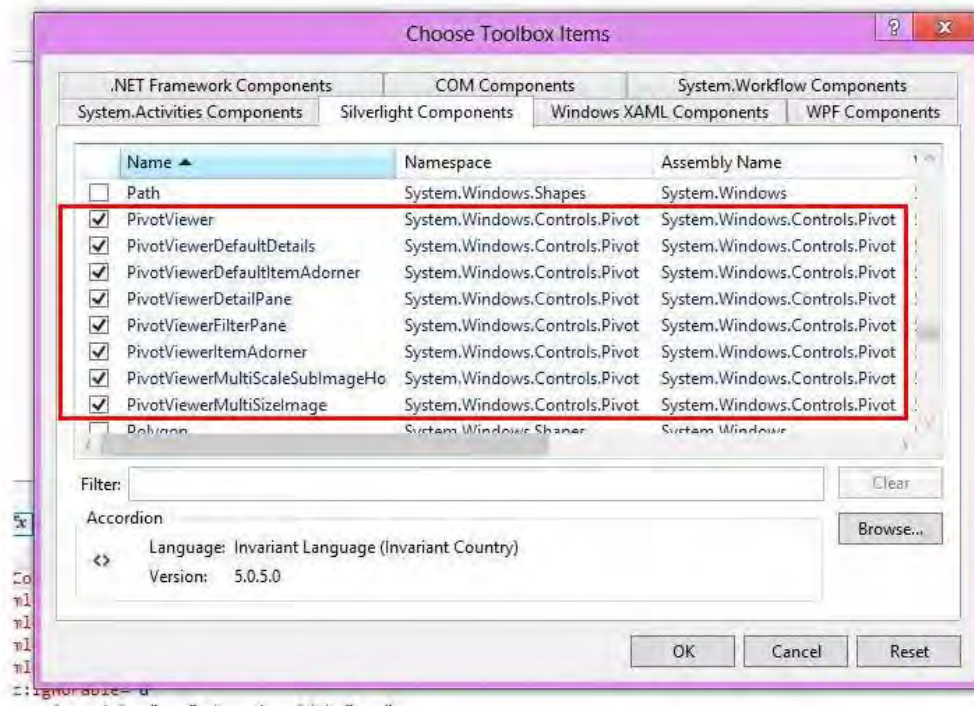
//Création d'une collection d'objets

ObservableCollection<Item> items = new ObservableCollection<Item>();

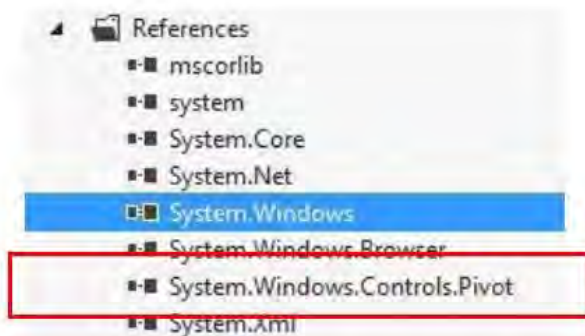
items.Add(new Item { Name = "Ballon", Description = "Ballon de football", CreationDate = DateTime.Now.AddDays(12)});
items.Add(new Item { Name = "Chat", Description = "Race Siamois", CreationDate = DateTime.Now.AddDays(19) });
items.Add(new Item { Name = "Ballon", Description = "Ballon de volley", CreationDate = DateTime.Now.AddDays(1) });
items.Add(new Item { Name = "Ballon", Description = "Ballon de basketBall", CreationDate = DateTime.Now.AddDays(13) });
items.Add(new Item { Name = "Robe", Description = "Robe de mariage", CreationDate = DateTime.Now.AddDays(20) });
items.Add(new Item { Name = "Ballon vert", Description = "Ballon de handBall", CreationDate = DateTime.Now.AddDays(52) });
items.Add(new Item { Name = "Ballon", Description = "Ballon de rugby", CreationDate = DateTime.Now.AddDays(22) });
items.Add(new Item { Name = "Chien", Description = "Race Berger", CreationDate = DateTime.Now.AddDays(31) });
items.Add(new Item { Name = "Chien", Description = "Race Dalmatien", CreationDate = DateTime.Now.AddDays(7) });
```

Le pivot viewer n'apparaît pas dans la liste Toolbox. Pour cela nous devons faire:

Click droit / add new Items/ Silverlight Components et sélectionnez tous les composants "PivotViewer ..."/ok



Apparition dans les références de "System.Windows.Controls.Pivot".



Dans la "MainPage.xaml", nous devons déclarer la nouvelle référence:

```

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:pivot="clr-namespace:System.Windows.Controls.Pivot;assembly=System.Windows.Controls.Pivot"

```

Nous pouvons maintenant utiliser le PivotViewer. L'intelligence nous propose toutes les différentes options présentes pour cette fonctionnalité.

```

<pivot:PivotViewer.PivotProperties>
  <pivot:PivotViewerStringProperty Id="Name"
    Options="CanFilter"
    DisplayName="Name"
    Binding="{Binding Name}"/>
  <pivot:PivotViewerStringProperty Id="Description"
    Options="CanSearchText"
    DisplayName="Description"
    Binding="{Binding Description}"/>
  <pivot:PivotViewerDateTimeProperty Id="CreationDate"
    Options="CanFilter"
    DisplayName="Creation date"
    Binding="{Binding CreationDate}"/>
</pivot:PivotViewer.PivotProperties>

<pivot:PivotViewer.ItemTemplates>
  <pivot:PivotViewerItemTemplate>
    <Border Width="200" Height="200">
      <StackPanel>
        <TextBlock Text="{Binding Name}" FontWeight="bold"/>
        <TextBlock Text="{Binding CreationDate}"/>
        <TextBlock Text="{Binding Description}" Margin="0,10,0,0"/>
      </StackPanel>
    </Border>
  </pivot:PivotViewerItemTemplate>
</pivot:PivotViewer.ItemTemplates>

```

À la fin de la collection d'objet, dans MainPage.xaml.cs, il ne faut pas oublier de lier ces objets avec le PivotViewer de MainPage.xaml :

```

<Grid x:Name="LayoutRoot" Background="White">
  <pivot:PivotViewer x:Name="pvt">
    |

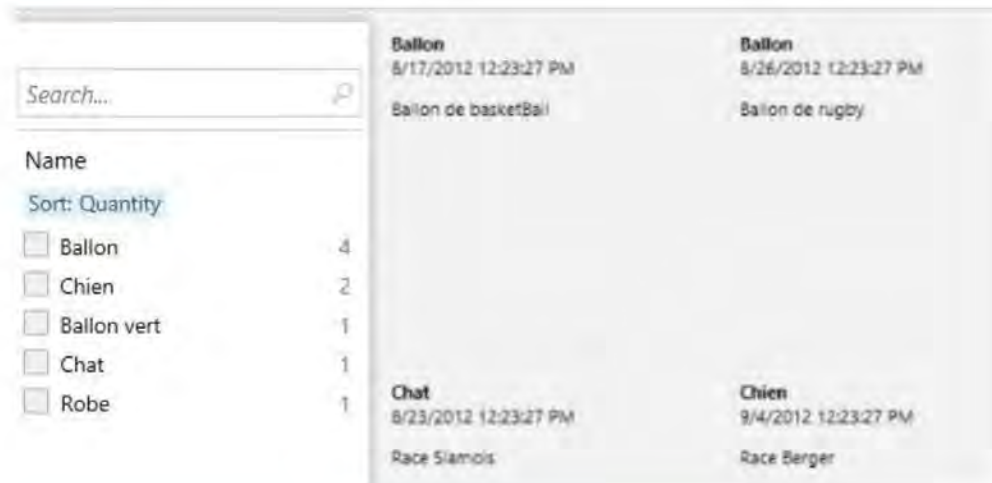
    items.Add(new Item { Name = "Chien",
    pvt.ItemsSource = items;
  }

```



pvt est le nom du PivotViewer et items le nom de la collection.

## Résultat

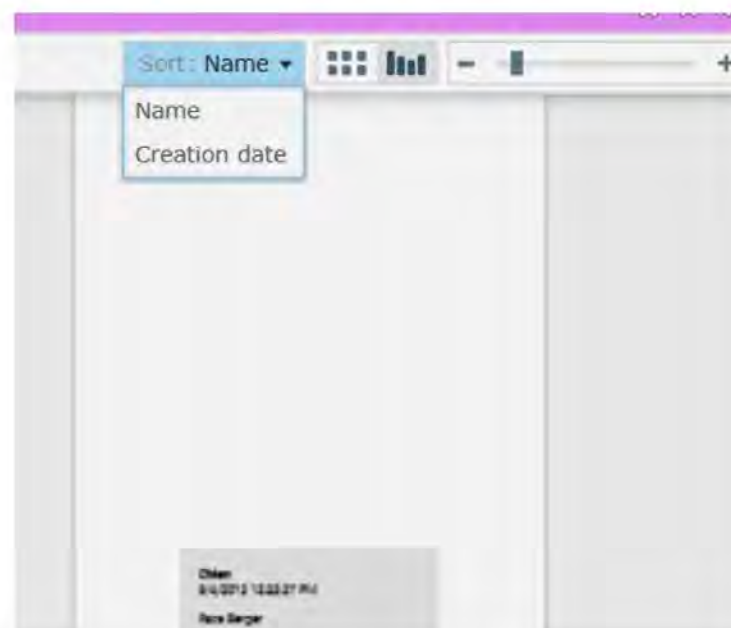


The screenshot shows a PivotViewer interface. On the left, there is a search bar labeled "Search..." and a list of items with checkboxes and quantities. The items are: Ballon (4), Chien (2), Ballon vert (1), Chat (1), and Robe (1). The "Sort: Quantity" option is selected. On the right, there are two pivot views. The top pivot view shows "Ballon" items: "Ballon de basketBall" (created 8/17/2012 12:23:27 PM) and "Ballon de rugby" (created 8/26/2012 12:23:27 PM). The bottom pivot view shows "Chat" items: "Race Siamois" (created 8/23/2012 12:23:27 PM) and "Race Berger" (created 9/4/2012 12:23:27 PM).

Name	Quantity
<input type="checkbox"/> Ballon	4
<input type="checkbox"/> Chien	2
<input type="checkbox"/> Ballon vert	1
<input type="checkbox"/> Chat	1
<input type="checkbox"/> Robe	1

Item	Creation date
Ballon de basketBall	8/17/2012 12:23:27 PM
Ballon de rugby	8/26/2012 12:23:27 PM
Race Siamois	8/23/2012 12:23:27 PM
Race Berger	9/4/2012 12:23:27 PM

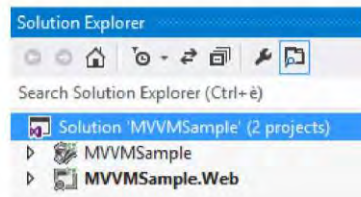


<http://www.youtube.com/watch?v=KwmWbik-kdU>

Auteur : Stylx59

## 11.2.2 MVVM

Création d'un projet Silverlight : MVVMSample



### MODEL

Dans MVVMSample.Web, création de 2 classes

PersonData.cs

```
namespace MVVMSample.Web
{
    //Exemple tiré de : http://learningcsharp.blogspot.com

    public class PersonData
    {
        public int PersonId { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public int Age { get; set; }
        public string Address { get; set; }
    }
}
```

PersonRepository.cs

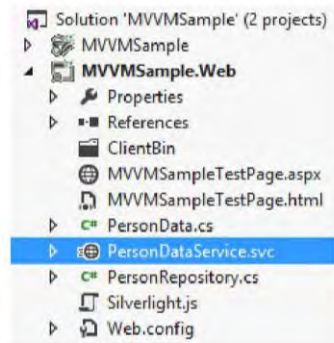
```
namespace MVVMSample.Web
{
    public class PersonRepository
    {
        public IList<PersonData> Person { get; set; }

        public PersonRepository()
        {
            GeneratePersonList();
        }

        public void GeneratePersonList()
        {
            Person = new List<PersonData>()
            {
                new PersonData() { PersonId = 1, FirstName = "Sonia", LastName = "Ragosta",
                                   Age=46, Address="Sierre"},
                new PersonData() { PersonId = 2, FirstName = "Antonio", LastName = "Foti",
                                   Age=46, Address="Melito Porto Salvo"},
                new PersonData() { PersonId = 3, FirstName = "Tina", LastName = "Romeo",
                                   Age=21, Address="Sion"},
                new PersonData() { PersonId = 4, FirstName = "Francesca", LastName = "Crettaz",
                                   Age=18, Address="Brig"},
            };
        }
    }
}
```



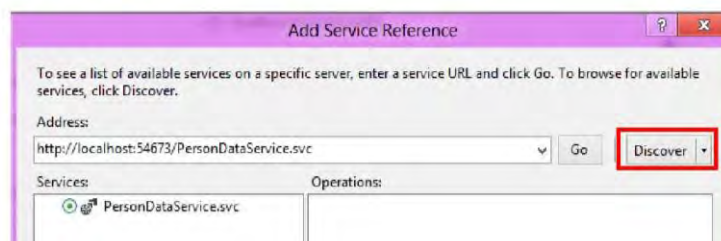
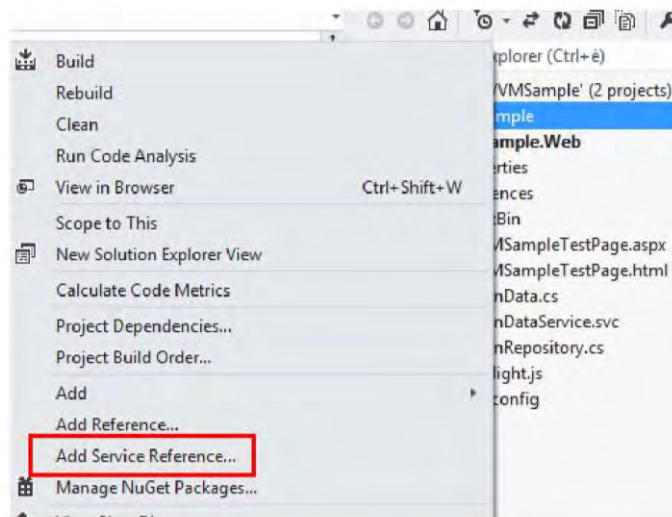
## Ajout d'un WCF Service Silverlight



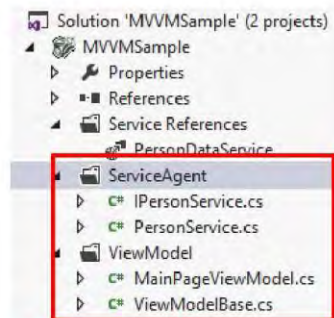
```
namespace MVVMSample.Web
{
    [ServiceContract(Namespace = "")]
    [AspNetCompatibilityRequirements(RequirementsMode =
        AspNetCompatibilityRequirementsMode.Allowed)]
    public class PersonDataService
    {
        [OperationContract]
        public PersonData FindPerson(int personID)
        {
            // Add your operation implementation here
            return new PersonRepository().Person.Where(x =>
                x.PersonId == personID).SingleOrDefault(); ;
        }
    }
}
```

Build la solution

## Ajout du nouveau Service au projet



## VIEWMODEL



```
using System.Windows.Snapes;
using MVVMSample.PersonDataService;

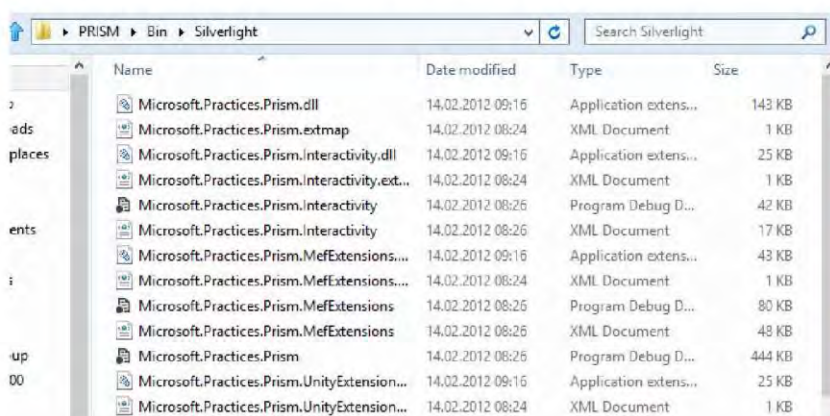
namespace MVVMSample.ServiceAgent
{
    public interface IPersonService
    {
        void FindPerson(int personID, EventHandler<FindPersonCompletedEventArgs> callback);
    }
}

using MVVMSample.PersonDataService;

namespace MVVMSample.ServiceAgent
{
    public class PersonService : IPersonService
    {
        private PersonDataServiceClient client = new PersonDataServiceClient();

        public void FindPerson(int personID, EventHandler<FindPersonCompletedEventArgs> callBack)
        {
            client.FindPersonCompleted += callBack;
            client.FindPersonAsync(personID);
        }
    }
}
```

Télécharger PRISM et ajouter Microsoft.Practices.Prism.dll comme référence



## Création de MainPageViewModel

### Les références

```
using MVVMSample.PersonDataService;  
using MVVMSample.ServiceAgent;  
using Microsoft.Practices.Prism;  
using Microsoft.Practices.Prism.Commands;
```

### attributs de la classe

```
//attributs  
  
private PersonService _personService;  
private PersonData _personData;  
private int _personID;  
  
public ICommand FindPersonCommand { get; set; }
```

### Le constructeur

```
public MainPageViewModel()  
{  
    if (isDesignTime != true)  
    {  
        this._personService = new PersonService();  
        this.FindPersonCommand = new DelegateCommand<object>(this.FindPersonByID);  
    }  
}
```

DelegateCommand provient de Microsoft.Practices.Prism.Commands

### Getter et Setter

```
public PersonData PersonData  
{  
    get  
    {  
        return _personData;  
    }  
    set  
    {  
        if (_personData != value)  
        {  
            _personData = value;  
            OnPropertyChanged("PersonData");  
        }  
    }  
}
```

```
public int PersonID  
{  
    get  
    {  
        return _personID;  
    }  
    set  
    {  
        if (_personID != value)  
        {  
            _personID = value;  
            OnPropertyChanged("PersonID");  
        }  
    }  
}
```

La méthode « FindPersonByID »

```
private void FindPersonByID(object obj)  
{  
    if (PersonID != 0)  
    {  
        _personService.FindPerson(PersonID, (s, e) => PersonData = e.Result);  
    }  
}
```

## VIEW

Nous allons utiliser le Binding pour lier la View à la ViewModel

The screenshot shows a WPF user interface with a light blue border. It contains five text boxes with labels to their left: 'Person ID : 0', 'First Name :', 'Last Name :', 'Age :', and 'Adress :'. To the right of the 'Person ID' text box is a button labeled 'SEARCH'. The text boxes are arranged vertically, and the labels are aligned to the left of each text box. The 'Person ID' text box has a value of '0'.

```

<UserControl.DataContext>
    <viewModel:MainPageViewModel/>
</UserControl.DataContext>

<sdk:Label Height="25" Grid.Row="1" Grid.Column="0"
    HorizontalAlignment="Right" Name="label1" VerticalAlignment="Center"
    Width="120" Content="Person ID : "/>

<TextBox Height="25" Grid.Row="1" Grid.Column="1"
    HorizontalAlignment="Left" Name="txtPersonID" VerticalAlignment="Top"
    Width="150" Text="{Binding PersonID, Mode=TwoWay}"/>

<Button Grid.Row="1" Grid.Column="2" Content="SEARCH"
    Height="25" HorizontalAlignment="Left" Name="btnSearch"
    VerticalAlignment="Top" Width="75" Command="{Binding FindPersonCommand}"/>

//attributs

private PersonService _personService;
private PersonData _personData;
private int _personID;

public ICommand FindPersonCommand { get; set; }

public MainPageViewModel()
{
    if (isDesignTime != true)
    {
        this._personService = new PersonService();
        this.FindPersonCommand = new DelegateCommand<object>(this.FindPersonByID);
    }
}

private void FindPersonByID(object obj)
{
    if (PersonID != 0)
    {
        _personService.FindPerson(PersonID, (s, e) => PersonData = e.Result);
    }
}

```



```

<sdk:Label Height="25" Grid.Row="3" Grid.Column="0"
    HorizontalAlignment="Right" Name="label12" VerticalAlignment="Center"
    Width="120" Content="First Name : "/>
<TextBox Height="25" Grid.Row="3" Grid.Column="1"
    HorizontalAlignment="Left" Name="txtFirstName"
    VerticalAlignment="Top" Width="200" Text="{Binding Path=PersonData.FirstName, Mode=TwoWay}"/>

```

```

public class PersonData
{
    public int PersonId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public int Age { get; set; }
    public string Address { get; set; }
}

```

```

<sdk:Label Height="25" Grid.Row="5" Grid.Column="0"
    HorizontalAlignment="Right" Name="label13" VerticalAlignment="Center"
    Width="120" Content="Last Name : "/>
<TextBox Height="25" Grid.Row="5" Grid.Column="1"
    HorizontalAlignment="Left" Name="txtLastName"
    VerticalAlignment="Top" Width="200" Text="{Binding Path=PersonData.LastName, Mode=TwoWay}"/>

<sdk:Label Height="25" Grid.Row="7" Grid.Column="0"
    HorizontalAlignment="Right" Name="label14" VerticalAlignment="Center"
    Width="120" Content="Age : "/>
<TextBox Height="25" Grid.Row="7" Grid.Column="1"
    HorizontalAlignment="Left" Name="txtAge"
    VerticalAlignment="Top" Width="200" Text="{Binding Path=PersonData.Age, Mode=TwoWay}"/>

<sdk:Label Height="25" Grid.Row="9" Grid.Column="0"
    HorizontalAlignment="Right" Name="label15" VerticalAlignment="Center"
    Width="120" Content="Address : "/>
<TextBox Height="25" Grid.Row="9" Grid.Column="1"
    HorizontalAlignment="Left" Name="txtAddress"
    VerticalAlignment="Top" Width="200" Text="{Binding Path=PersonData.Address, Mode=TwoWay}"/>

```

## RESULTAT

Person ID :

First Name :

Last Name :

Age :

Address :

Person ID :

First Name :

Last Name :

Age :

Address :

<http://learningcsharp.blogspot.ch/2011/10/mvvm-with-silverlight-start-to-end.html>

Auteur : Erandika Sandaruwan

### 11.2.3 LightSwitch

#### Application LightSwitch

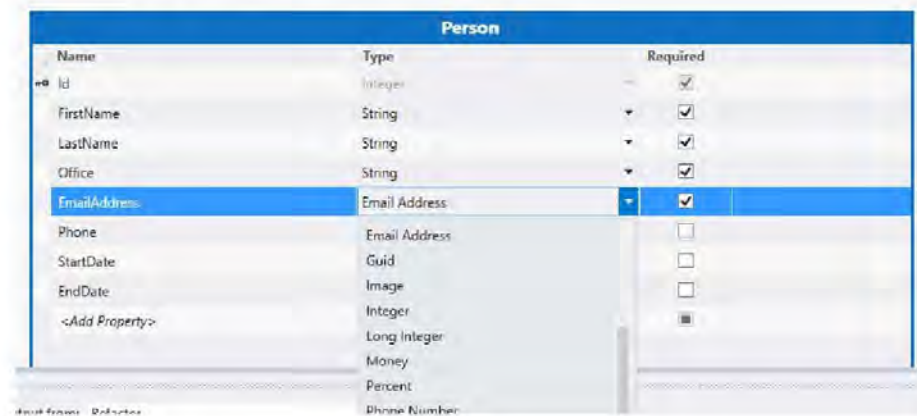
Création ou choisir une base de données



#### Create new table



Le type des champs est prédéfini, il ne reste que choisir



On peut créer une liste de type

The screenshot shows a 'Properties' dialog box with a 'General' tab. Under the 'Type' section, there are three checkboxes: 'Include in Unique Index' (unchecked), 'Is Computed' (unchecked), and 'Is Searchable' (checked). Below these is a 'Name:' label and a 'Type' dropdown menu. The 'Choice List...' option is highlighted with a red rectangle.

The 'Choice List' dialog box is shown, titled 'Define the choices for the property:'. It contains a table with two columns: 'Value' and 'Display Name'. The table lists five entries: 'C' for 'Computer', 'M' for 'Monitor', 'p' for 'Printer', 'S' for 'Screen', and 'O' for 'Other'. At the bottom of the table are two links: 'Add Value' and 'Add Display Name'. The dialog has 'OK' and 'Cancel' buttons at the bottom right.

Value	Display Name
C	Computer
M	Monitor
p	Printer
S	Screen
O	Other
<i>Add Value</i>	<i>Add Display Name</i>



## Relationship

**Add New Relationship** ✕

Select two tables and properties to form a relationship between the two:

	From	To
Name:	Asset	
Multiplicity:	Many	Asset
On Delete Behavior:		Person
Navigation Property:		

```
graph LR; Asset[Asset] ---|∞| Person[Person];
```

! Something needs to be selected for both sides of a relationship.

OK Cancel

**Add New Relationship** ✕

Select two tables and properties to form a relationship between the two:

	From	To
Name:	Asset	Person
Multiplicity:	Many	One
On Delete Behavior:		Restricted
Navigation Property:	Person	Assets

```
graph LR; Asset[Asset] ---|∞| Person[Person];
```

A 'Asset' **must have** a 'Person'.

A 'Person' **can have** many 'Asset' instances.

'Person' **cannot be deleted**, if there are related 'Asset' instances.

OK Cancel

Asset			
Person			
go			
1			
Person			
Name	Type		Required
LastName	String	▼	<input checked="" type="checkbox"/>
Office	String	▼	<input checked="" type="checkbox"/>
EmailAddress	Email Address	▼	<input checked="" type="checkbox"/>
Phone	Phone Number	▼	<input type="checkbox"/>
StartDate	Date	▼	<input type="checkbox"/>
EndDate	Date	▼	<input type="checkbox"/>
Assets	Asset Collection	▼	<input checked="" type="checkbox"/>
<Add Property>		▼	<input checked="" type="checkbox"/>

## Création des écrans

**Add New Screen**

Select a screen template:

- Details Screen**
- Editable Grid Screen
- List and Details Screen
- New Data Screen
- Search Data Screen

**Provide screen information:**

Screen Name: Detail

Screen Data: (None)

Use as Default Details Screen

Additional Data to Include:

**Details Screen**  
A screen used to display a single entity and its children, if any. Details screens do not appear in the application's navigation menu; they are launched from other screens.

Plusieurs possibilités

NEWPERSON

**Add New Screen**

Select a screen template:

- Details Screen
- Editable Grid Screen
- List and Details Screen
- New Data Screen**
- Search Data Screen

**Provide screen information:**

Screen Name: NewPerson

Screen Data: Person

Additional Data to Include:

- ☒ Person Details
- ☒ Person Assets

**New Data Screen**  
A screen used to create a new data record. The screen can also be used to create child records.

## Résultat

TASKS

New Asset ★ Antonio - Person Sonia - Person New Person ★

First Name:

Phone:

Last Name:

Start Date:

Office:

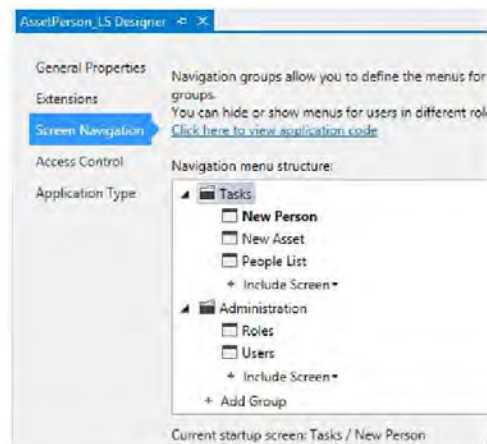
End Date:

Email Address:

Assets

NAME	TYPE	DATE ACQUIRED	COST

## Navigation



TASKS

People List  
New Person  
New Asset

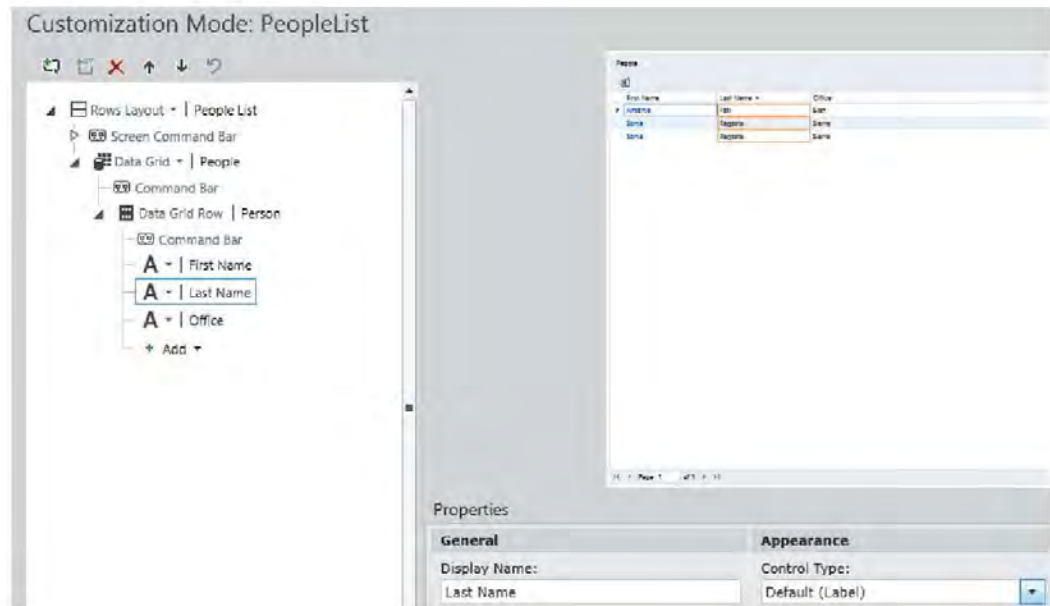
FIRST NAME	LAST NAME	OFFICE
Sonia	Ragosta	Sierre
Antonio	Foti	Sion
▶ Sonia	Ragosta	Sierre

People list se trouve en première position

## Type d'application (Properties)



## Modification du Design (Design Screen)



<http://msdn.microsoft.com/fr-fr/vstudio/msdn.lightswitch.aspx>

Auteur : Eric Vernié

## 11.3 Cahier des charges



### EVOLUTION DE L'IHM D'UNE PLATEFORME REALISEE EN SILVERLIGHT



Sierre, le 23 avril 2012

## Table des matières

---

1	INTRODUCTION .....	1
2	COMPRÉHENSION DE L'ARCHITECTURE DE DÉVELOPPEMENT DE L'APPLICATION EXISTANTE. 2	
3	POSITIONNEMENT DE MICROSOFT FACE AU FUTUR DU DÉVELOPPEMENT WEB.....	3
4	ETUDE D'AUTRES TECHNOLOGIES.....	4
5	DÉTERMINATION / CHOIX .....	4
6	OPTIONNEL : DÉVELOPPEMENT 3D WEB.....	5
7	RAPPORTS ET PLANNING.....	6
8	CONCLUSION .....	6
9	TABLE DES ILLUSTRATIONS.....	7
10	SOURCES .....	8
	10.1 LE COACH SILVERLIGHT 3 .....	8
	10.2 SILVERLIGHT 4 – TRAINING.....	8
	10.3 SILVERLIGHT 5 – LES NOUVEAUTÉS .....	8
	10.4 ARCHITECTURE MVVM.....	8
	10.5 DÉVELOPPEMENT 3D .....	8
I	ANNEXE : PLANNING PRÉVISIONNEL.....	9



## 1 Introduction

---

Dans le cadre du projet de recherche MEU (Management Energétique Urbain) en collaboration avec le CREM, l'EPFL et de l'IHES-SO, une plate-forme à l'usage des villes a été réalisée en Silverlight et C#. Cette plate-forme représente un outil d'aide à la décision. Elle va aider les utilisateurs à élaborer des stratégies énergétiques des zones urbaines.

Le but du travail de bachelor « Evolution de l'IHM d'une plateforme réalisée en Silverlight » est d'analyser les éléments suivants :

- Compréhension du fonctionnement de Silverlight
- Compréhension de l'architecture de développement de l'application existante
  - Modèle MVVM (Model – View – ViewModel)
- Positionnement de Microsoft face à Silverlight par rapport au futur du développement web
- Positionnement de Microsoft face à HTML 5
- Recommandations de la technologie idéale de développement pour la suite du projet
- Petite partie codage de proto dans les différentes phases (testing)
- Si temps, Framework pour un développement 3D web pouvant s'intégrer dans ce projet ou un autre similaire.

Figure 1 : Silverlight



Source :

<http://labs.infinitiesquare.com/fr/Projects/BI/SLFor3D/>



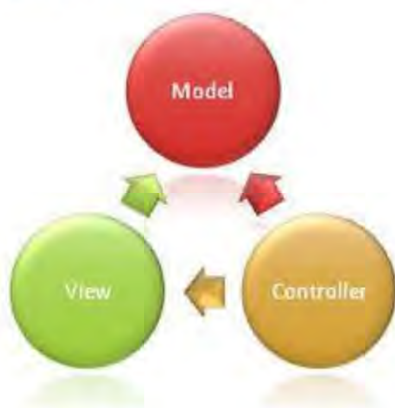
## 2 Compréhension de l'architecture de développement de l'application existante

---

Silverlight utilise une architecture MVVM (Model – View – ViewModel). Cette architecture a été développée par Microsoft. Elle dérive de l'architecture MVC (Model – View – Controller). Elle est également définie comme une architecture MVC passive qui a en plus du binding.

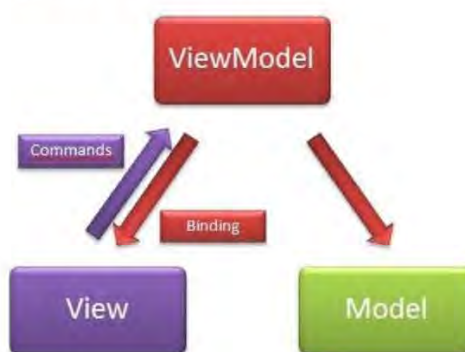
Pendant cette phase, nous analyserons l'architecture existante et son fonctionnement.

Figure 2 : Architecture MVC



Source : <http://nexeo.fr/nexdotnet/>

Figure 3 : Architecture MVVM



Source :  
<http://www.silverlightblog.nettagmvvm>

### 3 Positionnement de Microsoft face au futur du développement web

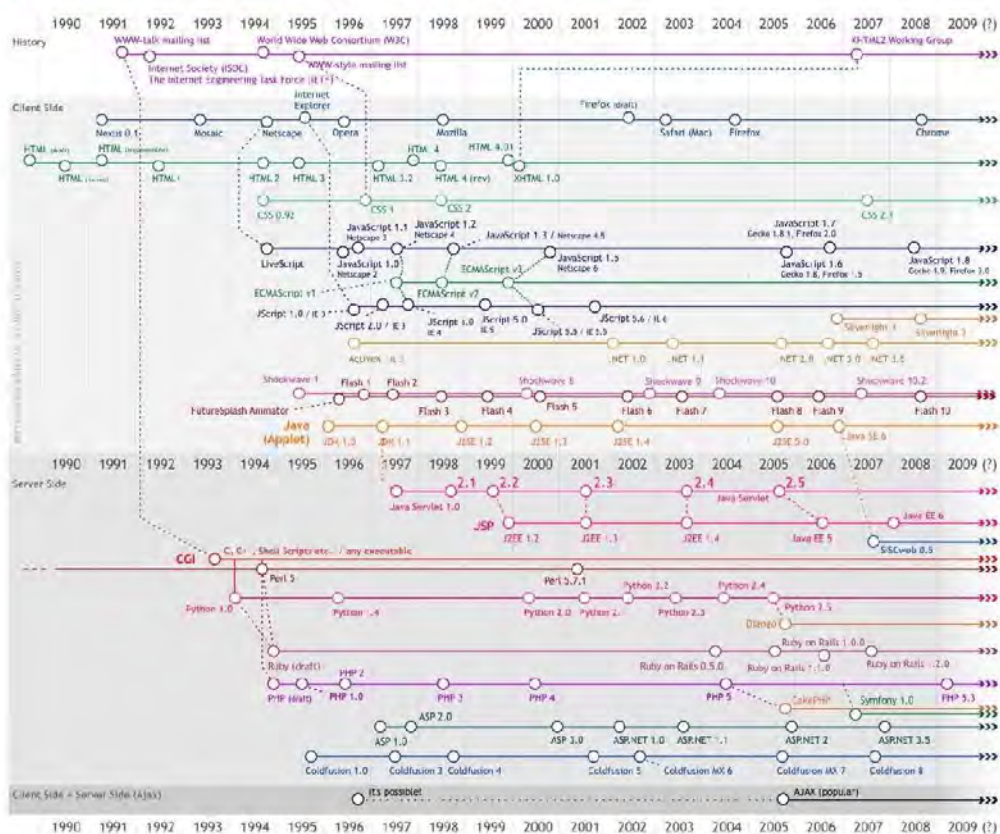
Le développement web est en continuelle évolution.

Mais comment se positionne Microsoft face à cette évolution ?

Quelle est sa vision pour les années à venir ?

Ce chapitre va essayer de trouver une réponse à ces questions.

#### Figure 4 : Histoire du développement web jusqu'en 2009



Source : <http://www.kgaut.net/Blog/index.php?post/2009/06/14/>

## 4 Etude d'autres technologies

---

La version 5 de Silverlight a été présentée le 8 décembre 2011. Selon quelques rumeurs, Microsoft n'aurait plus l'intention de poursuivre le développement de Silverlight. Si tel est le cas, pouvons – nous la remplacer par un autre plug-in ?

En outre, nous analyserons également le positionnement de Microsoft face à HTML5.

Figure 5 : Le guide pour RIA



Source [http://pro.01net.com/editorial/522394/flash-html5-silverlight-le-guide-pour-ria-mieux-choisir-\(2-6\)-silverlight-maj/](http://pro.01net.com/editorial/522394/flash-html5-silverlight-le-guide-pour-ria-mieux-choisir-(2-6)-silverlight-maj/)

## 5 Détermination / Choix

---

Lorsque cette phase d'analyse sera terminée, nous pourrons proposer une solution pour maintenir l'IHM du produit actuel.





## **7 Rapports et planning**

---

En annexe, vous trouverez un planning prévisionnel pour toute la durée du projet.

En outre, toutes les deux semaines, vous recevrez par mail un rapport de quelques lignes expliquant le déroulement du projet, les problèmes rencontrés ainsi que le planning des deux prochaines semaines.

## **8 Conclusion**

---

Les premières analyses effectuées montrent que le sujet est vaste et complexe.

Ce projet va permettre de se familiariser avec différents outils et de pouvoir apporter une réponse à toutes les questions qui se posent actuellement pour l'évolution du projet MEU.

## 9 Table des illustrations

---

FIGURE 1 : SILVERLIGHT .....	1
FIGURE 2 : ARCHITECTURE MVC.....	2
FIGURE 3 : ARCHITECTURE MVVM.....	2
FIGURE 4 : HISTOIRE DU DÉVELOPPEMENT WEB JUSQU'EN 2009 .....	3
FIGURE 5 : LE GUIDE POUR RIA .....	4
FIGURE 6 : DÉVELOPPEMENT 3D - SILVERLIGHT.....	5
FIGURE 7 : DÉVELOPPEMENT 3D - ARCGIS 3D ANALYST .....	5

## 10 Sources

---

### 10.1 Le coach Silverlight 3

<http://msdn.microsoft.com/fr-fr/silverlight/cc511510>

### 10.2 Silverlight 4 – Training

<http://msdn.microsoft.com/en-us/Silverlight4TrainingCourse>

### 10.3 Silverlight 5 – Les nouveautés

<http://msdn.microsoft.com/fr-fr/silverlight/hh184212>

### 10.4 Architecture MVVM

<http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>

### 10.5 Développement 3D

<http://www.developpez.com/actu/37751/Silverlight-5-Toolkit-la-boite-a-outils-ameliore-la-gestion-des-elements-3D-integre-de-nouveaux-modeles-de-projet-et-plusieurs-demos/>

<http://timheuer.com/blog/archive/2011/04/13/whats-new-in-silverlight-5-a-guide.aspx>

<http://www.esri.com/software/arcgis/extensions/3danalyst/demos.html>



## 11.4 Planning

[illegible]

Activités	Plan		du 26.03.2012 au 01.04.2012																
			L		M		M		J		V		S		D				
			M	A	M	A	M	A	M	A	M	A	M	A	M	A			
	TTP	14																	
Consultation des documents reçus	TP	3																	
	TE	2			2														
Préparation cahier des charges	TP	4																	
	TE	6			2		2		1	1									
Silverlight - Analyse	TP	4																	
	TE	2					1			1									
Silverlight - Les tutoriaux	TP	3																	
	TE	3							3										
Création d'une machine virtuelle	TP																		
	TE	2									2								
	TP																		
	TE	0																	
	TP																		
	TE	0																	
	TP																		
	TE	0																	
Temps total planifié		14																	
	TTE	15																	

TTP42

TTE28

Activités	Plan		du 02.04.2012 au 08.04.2012																
			L		M		M		J		V		S		D				
			M	A	M	A	M	A	M	A	M	A	M	A	M	A			
	TTP	14																	
Silverlight - Les tutoriaux (1 à 3)	TP	14																	
	TE	5			2				3										
Cahier des charges - finalisation	TP																		
	TE	1			1														
Mini-rapport - Rédaction	TP																		
	TE	1.0			0.5				0.5										
Lecture - Silverlight	TP																		
	TE	4									2						2		
Planif. des 2 semaines suivantes	TP																		
	TE	1							1										
Création d'un compte dropbox pour T	TP																		
	TE	0.25			0.25														
Bug - VM	TP																		
	TE	2							2										
	TP																		
	TE	0																	
Temps total planifié		14																	
	TTE	14.25																	

TTP56

TTE42.25

Délivrable :

- Cahier des charges

- mini-rapport

OK



Activités	Plan		du 23.04.2012 au 29.04.2012															
			L		M		M		J		V		S		D			
			M	A	M	A	M	A	M	A	M	A	M	A	M	A		
	TTP	14																
Rédaction rapport - 1 <sup>ère</sup> partie	TP	3																
	TE	8							3	2	3							
Application des concepts SL	TP	9																
	TE	8		3		5												
Concept MVVM	TP	2																
	TE	2						1				1						
	TP																	
	TE	0																
	TP																	
	TE	0																
	TP																	
	TE	0																
	TP																	
	TE	0																
	TP																	
	TE	0																
Temps total planifié		14																
	TTE	18																

TTP84

TTE93.25

Délivrable :

- Application SL

- Rapport - 1ère étape

Activités	Plan		du 30.04.2012 au 06.05.2012															
			L		M		M		J		V		S		D			
			M	A	M	A	M	A	M	A	M	A	M	A	M	A		
	TTP	14																
Concept MVVM	TP	9																
	TE	11		1					3	3	4							
Rédaction rapport	TP	4																
	TE	0																
Séance 02.05.12	TP	1																
	TE	0																
Rédaction rapport_1	TP																	
	TE	4		2		2												
Application_Modification	TP																	
	TE	2				2												
	TP																	
	TE	0																
	TP																	
	TE	0																
	TP																	
	TE	0																
Temps total planifié		14																
	TTE	17																

TTP98

TTE110.25

Délivrable :

- mini-rapports

OK

<b>Délivrable :</b>			
- Application SL - MVVM			
- Rapport - Concept MVVM			

<b>Délivrable :</b>	
- Mini-rapports	



[illegible][illegible][illegible][illegible]

<b>Délivrable :</b>					
- PV	OK				
- Mini- Rapport					



[illegible]

Activités	Plan		du 18.06.2012 au 24.06.2012															
			L		M		M		J		V		S		D			
			M	A	M	A	M	A	M	A	M	A	M	A	M	A		
	TTP	0																
Microsoft - HTML5	TP	8																
	TE	0																
Concept HTML5	TP	8																
	TE	0																
Application HTML5	TP	8																
	TE	0																
Application Silverlight - HTML 5	TP																	
	TE	16		2		1		5		5	3							
Préparation environnement de travail	TP																	
	TE	4		2		1						1						
Rédaction rapport	TP																	
	TE	3				1		1					1					
Installation de VS Professional RC + Test (LightSwitch)	TP																	
	TE	1											1					
Prise de connaissance de la doc reçue vendredi soir	TP																	
	TE	3											3					
Temps total planifié		24																
	TTE	27																

TTP202

TTE234.25

Délivvable :							
--------------	--	--	--	--	--	--	--

- Applications	KO				
----------------	----	--	--	--	--

- Mini- Rapport	KO				
-----------------	----	--	--	--	--

[illegible][illegible]

Délivvable :						
--------------	--	--	--	--	--	--

- Mini-rapports	KO				
-----------------	----	--	--	--	--

- PV				OK			
------	--	--	--	----	--	--	--

[illegible]

Activités	Plan		du 02.07.2012 au 08.07.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	24														
Recommandation	TP	24														
	TE	0														
VS LightSwitch - Doc reçues	TP															
	TE	6		2		2		2								
VS LightSwitch - Appli + Rapport	TP															
	TE	0														
Rapport MVVM - Compléter	TP															
	TE	4									2	2				
OData - Rapport	TP	4														
	TE	3											3			
Windows Phone - Application	TP	4														
	TE	0														
OData - Application	TP	2														
	TE	1											1			
	TP															
	TE	0														
Temps total planifié		10														
	TTE	14														

TTP

226

TTE

268.25

- Rapport V2		MVVM	OK
- Mini-rapports			OK

[illegible]

Activités	Plan		du 09.07.2012 au 15.07.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	24														
Développement 3D - Concept	TP	12														
	TE	0														
Développement 3D - Application	TP	12														
	TE	0														
	TP															
	TE	0														
OData - Recherche info	TP															
	TE	4		2		1							1			
Windows Phone - Application	TP															
	TE	1											1			
Visual Studio Light Switch	TP															
	TE	6		1								2	3			
TechEd 2012 - Visualisation video	TP															
	TE	7			3		2				1		1			
	TP															
	TE	0														
Temps total planifié		24														
	TTE	18														

TTP

250

TTE

286.25

[illegible][illegible]

Délivrible :						
--------------	--	--	--	--	--	--

- Mini-rapports	OK		
-----------------	----	--	--

[illegible]

Activités	Plan		du 23.07.2012 au 29.07.2012														
			L		M		M		J		V		S		D		
			M	A	M	A	M	A	M	A	M	A	M	A	M	A	
	TTP	24															
Rapport	TP	24															
	TE	13		3				2			4		4				
Application Multiplateforme	TP																
	TE	6		1		1					4						
	TP																
	TE	0															
	TP																
	TE	0															
	TP																
	TE	0															
	TP																
	TE	0															
	TP																
	TE	0															
	TP																
	TE	0															
Temps total planifié		24															
	TTE	19															

TTP

298

TTE

326.25

Planning modifié																		
Activités	Plan		du 06.08.2012 au 11.08.2012														TTP	346
			L		M		M		J		V		S		D			
	M	A	M	A	M	A	M	A	M	A	M	A	M	A	M	A		
	TTP	24																
Rapport	TP	24																
	TE	6			1		2	1	2									
CD	TP																	
	TE	3									3							
Application Multiplateforme	TP																	
	TE	12		5	2		3	2										
Application Metro	TP																	
	TE	2							2									
	TP																	
	TE	0																
	TP																	
	TE	0																
	TP																	
	TE	0																
	TP																	
	TE	0																
Temps total planifié	TTP	24																
	TTE	23															TTE	366.25

- Rapports			
- CD			

## 11.5 PV

### 11.5.1 PV de la séance du 16.03.12 – N°01

#### Présents :

M. Jean-Pierre Rey

Mme Sonia Ragosta Foti

#### Lieu :

HEVS – Immeuble Bellevue à Sierre

#### 1. Plan proposé

- Compréhension du fonctionnement de Silverlight
- Compréhension de l'architecture de développement de l'application existante
  - Modèle MVVM (Model – View – View Model)
- Positionnement de Microsoft face à Silverlight par rapport au futur du développement web
- Positionnement de Microsoft face à HTML 5 (support – évolution face à HTML5)
- Recommandation pour le projet

#### Voir temps :

- Programmation 3D – Réalisation de quelques prototypes pour illustrer la faisabilité des éléments.  
Creuser profondément sur des outils, framework standard pour une modélisation en 3D sur web.  
Intégration de ces framework dans Silverlight.

#### 2. Notes de séance

MEU est un projet de planification énergétique urbain.

A ce stade des questions se posent :

- Quel est le futur de Silverlight ?
- Quelle est sa pérennité ?
- Quels sont les avantages et les inconvénients de
  - Silverlight
  - Flex
  - Adobe

- HTML5
- GWT
- Quelle est la ligne directrice de Microsoft ?
- Quel est le positionnement de Microsoft par rapport à HTML5 ?

Il faudrait également rechercher des informations sur Windows 8.

Silverlight n'utilise pas le modèle MVC (Model – View – Controller) mais le MVVM (Model – View – ViewModel).

- Comprendre le fonctionnement de ce modèle.

Le but de ce projet est de faire une analyse complète, de donner quelques recommandations et de créer des prototypes en 3D, si le temps est suffisant.

- Modélisation 3D sur web
  - Framework
  - Outils
  - Existe-t-il des standards ?

### 3. Consignes

#### Organisationnel

- Tenir une bibliographie à jour (lien ... - consulté le ...) => références (à joindre au TB)
- Envoyer toutes les deux semaines un rapport (à joindre dans les annexes du TB) :
  - Etat de l'avancement
  - Temps
  - Le réaliser
  - Et le à réaliser dans les semaines suivantes
- Rendez-vous selon besoin

Sierre, le 23 mars 2012

## 11.5.2 PV de la séance du 22.03.12 – N°02

### Présent :

M. Jean-Pierre Rey  
M. Alain Duc  
Mme Sonia Ragosta Foti

### Lieu :

Technopôle à Sous-Géronde

### 1. Notes de séance

MEU est un projet de planification énergétique urbain. Ce projet gère un parc immobilier. Actuellement il est phase pré commerciale. Ce projet est basé sur 4 villes.

### 2. Contextes techniques

- Carte ESRI
  - Carte avec des bâtiments dans un système géographique faisant appel à des web services
  - Elle propose des API :
    - Silverlight
    - Flex
    - JavaScript
  - Cette carte est le point de départ de l'application. Lorsque l'on clique sur la carte, elle récupère les différents éléments (Id, dimensions ...) qui se trouve dans une base de données développée par une personne de l'Epfl.
  - On peut ainsi créer des calculs énergétiques de consommation ou autres.

### 3. Problématique – Boîte de dialogue

Le modèle de conception est MVVM (Model – View – ViewModel).

Le code s'écrit dans le ViewModel. La page de code behind n'existe pas.

Les pop-up sont des composantes de Silverlight, comment peut-on les fermer sans avoir de code ?



#### 4. MVVM

MVVM est un modèle non contraignant, car nous ne sommes pas obligés d'écrire du code behind. Pour bien comprendre son fonctionnement, il serait préférable d'utiliser des Diagrammes de séquences (façon de travailler).

Ensuite commencer à le tester avec la création d'un simple bouton. Pour cela il faudra d'abord :

- Comprendre le principe
- Comment implémenter différents framework

Pour Silverlight, nous avons [MVVLight Toolkit](#).

Sierre, le 23 mars 2012

### 11.5.3 PV de la séance du 19.04.12 – N°03

#### Présent :

M. Jean-Pierre Rey  
Mme Sonia Ragosta Foti

#### Lieu :

Immeuble Bellevue - Sierre

#### 1. Cahier des charges

M. Rey a apporté des modifications du cahier des charges.

#### 2. Points à développer

- Comment Silverlight gère les évènements ?
- Va-t-on rester sur Silverlight ou passer à HTML5 ?
- Tester les concepts

#### 3. Planning

Nous devons lui remettre un planning prévisionnel pour toute la durée du projet.

Des sprints ont été fixés :

- Fin avril 2012 : Compréhension du fonctionnement de Silverlight
  - Délivrables :
    - une mini-application avec les concepts appris
    - Rapport de cette première étape
- Mi-mai 2012 : Compréhension de l'architecture MVVM
  - Délivrables :
    - Application avec les concepts MVVM
    - Rapport de la 2<sup>ème</sup> étape
- Mi-juin 2012 : Positionnement de Microsoft face à Silverlight
  - Délivrable :
    - Rapport

Selon le planning prévisionnel, les sprints suivants seront :

- Fin juin 2012 : Positionnement de Microsoft face à HTML5
  - Délivrable :
    - Rapport

- Début juillet 2012 : Recommandation
  - Délivrable :
    - Rapport
- Juillet 2012 : Programmation 3D
- Août 2012 : Rapport TB

#### 4. A remettre

- Rapport 1<sup>ère</sup> partie
- Application avec les nouveaux concepts (Testing)

#### 5. Prochaine séance

Mercredi 2 mai 2012 à 15h15 à Bellevue

Sierre, le 19 avril 2012

#### 11.5.4 PV de la séance du 02.05.12 – N°04

##### Présent :

M. Jean-Pierre Rey

M. Alain Duc (1<sup>ère</sup> partie)

Mme Sonia Ragosta Foti

##### Lieu :

Immeuble Bellevue – salle 310 - Sierre

### 1. Présentation de MEU

M. Alain Duc présente le projet MEU.

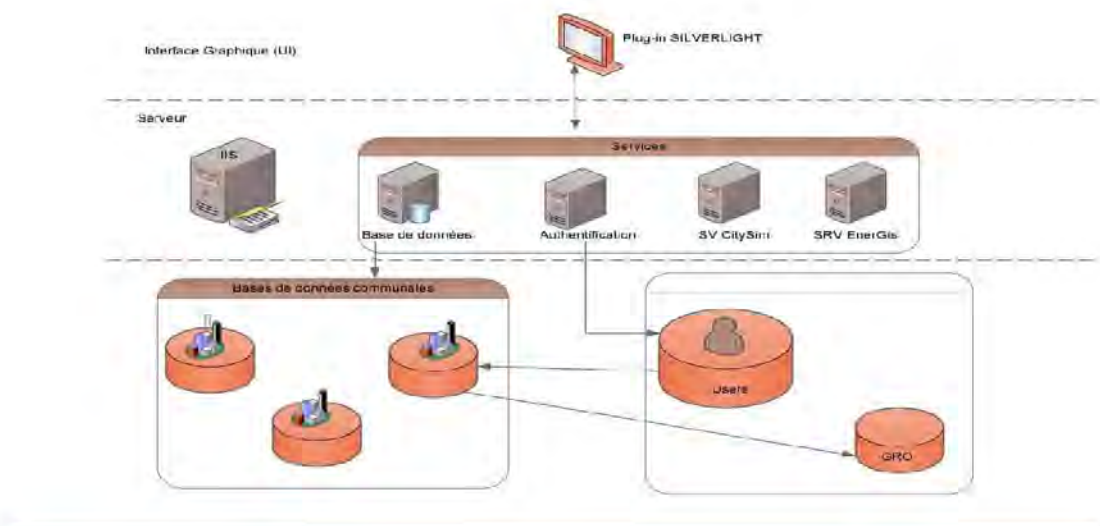
MEU est un outil pour la gestion d'un parc immobilier. C'est une aide à la décision. Par exemple, nous pouvons décider de changer le type de chauffage et de voir les conséquences que provoquerait telle ou telle solution.

Le projet a été conduit en collaboration avec l'EPFL, les instituts LENI et LESO et le CREM.

L'EPFL avait conduit un projet analogue pour la gestion de l'approvisionnement des eaux. C'est pourquoi, il a été décidé de partir sur le même principe.

MEU utilise ESRI et SIG BD, elle permet d'intégrer des données géographiques à d'autres données.

### 2. Architecture



Tout est en SOAP :

SOAP est un protocole de transmission de messages. Il définit un ensemble de règles pour structurer des messages qui peuvent être utilisés dans de simples transmissions unidirectionnelles, mais il est particulièrement utile pour exécuter des dialogues requête-réponse RPC (Remote Procedure Call). Il n'est pas lié à un protocole de transport particulier mais HTTP est populaire.

Le service BD fait appel à la BD.

Le client doit se loguer et selon son login il sera redirigé vers la base de données de sa ville.

SV CitySim simulateur de calcul.

SRV EnerGis calcule le besoin énergétique de la ville correspondante et le transmet à Silverlight.

Ils font tous appel à des Web Services :

- WS CitySim (C++)
- WS EnerGis ( matlab)

Tout se passe du côté client.

### 3. RIA

Permet d'être :

- Rapide
- Interface desktop
- Ouverture de plusieurs fenêtres avec beaucoup d'info.

### 4. Utilité de Silverlight

Silverlight permet une grande interaction. Il permet d'avoir plusieurs « Pop-up » ouverte et nous pouvons y travailler comme si on était sur le desktop mais on est sur le WEB.

### 5. A remettre

- Rapport 2ème partie
- Application du concept MVVM

### 6. Prochaine séance

Mercredi 16 mai 2012 à 15h15 à Bellevue

Sierre, le 4 mai 2012

## 11.5.5 PV de la séance du 16.05.12 – N°05

### Présent :

M. Jean-Pierre Rey

Mme Sonia Ragosta Foti

### Lieu :

Immeuble Bellevue - Sierre

### 1. MVC - MVVM

Il serait intéressant de trouver un exemple et de le développer avec le modèle MVC et ensuite d'utiliser le MVVM.

Nous pourrions ainsi voir les similitudes et les inégalités et les intégrer.

Ils existent des framework nous permettant de développer des applications utilisant le pattern MVVM, comme par exemple :

- MVVM Light Toolkit de Laurent Bugnion
- Simple MVVM Toolkit de Tony Sneed

(Liste des framework : <http://www.japf.fr/silverlight/mvvm/index.html>)

### 2. Planning proposé

Le prochain livrable fixé est :

Positionnement de Microsoft face à Silverlight et HTML5.

Approche théorique.

Si nous tenons compte de HTML5, existe-t-il des framework ?

Quel type de modèle utilise-t-il ?

Pistes :

- Liaison des données (Data Binding pour Silverlight)
- Accès aux ressources machine (stockage)
- Expérience de l'utilisateur face à l'interface graphique
- Intégration des Web Services
- Framework de développement « HTML5.net » pour faciliter le travail du développeur



### 3. A remettre

- Rapport 3<sup>ème</sup> partie – Positionnement de Microsoft face à Silverlight et HTML5

### 4. Prochaine séance

Mercredi 30 mai 2012 à 15h30 à Bellevue

### 5. But de la séance

- Validera la 1<sup>ère</sup> itération
- Définir quelques pistes de codage qui peuvent être choisies

Sierre, le 16 mai 2012



## 11.5.6 PV de la séance du 30.05.12 – N°06

### Présent :

M. Jean-Pierre Rey  
Mme Sonia Ragosta Foti

### Lieu :

Immeuble Bellevue - Sierre

### 1. Discussion

Pendant cette rencontre, nous avons discuté des dernières découvertes. Microsoft souhaite s'attaquer au marché des applications mobiles avec le lancement de Windows 8 et Metro Style Apps. Elle a introduit HTML5 dans ses langages de programmations pour attirer le plus de développeurs possible.

Elle va maintenir le plug-in Silverlight pour les applications desktop.

### 2. Définitions à chercher

- Natif
- Apps
- Desktop

### 3. Scénarii à prendre en compte

#### 3.1 N°1

Application web : application appelée à tourner que sur un navigateur.

Y a-t-il une influence avec les nouveautés de Windows 8 ?

#### 3.2 N°2

Application mobile

Est-ce que le principe de base est compliqué ?

#### 3.3 N°3

Est-ce que la demande serait d'avoir un produit qui pourrait être consommé soit sur web, soit sur web mobile et sur web tablette?

#### 4. Quelques questions

Une question que nous pourrions nous poser :

- a) Quand et pourquoi utiliser Métro ?

Ou encore :

- b) Nous avons différents types d'application
  - Application web
  - Application web pour mobile
  - Application web pour tablette

Comment passer d'une application à l'autre sans Metro ou avec Metro ?

(Peut-être qu'il n'y a qu'une feuille de style CSS qui change !)

- c) Comment tout fonctionne aujourd'hui et comment c'est 3 scénarii (§3) fonctionneront demain ?
- d) Quels sont les grands changements pour les scénarii 2 et 3 ?
- e) Doit-on penser à l'application différemment ?

#### 5. Planning proposé

Le prochain livrable fixé est :

Qu'est-ce que Métro ?

Approche théorique à approfondir.

Pistes :

- Framework
- Style
- Principe de programmation et d'utilisation
- Technique
- Architecture
- GuideLine « Règles de programmation »

#### 6. Machine virtuelle

M. David Russo a transmis le 30.05.12 une machine virtuelle avec Windows 8 et Visual Studio 11 version beta.

**7. A remettre**

- Rapport 4<sup>ème</sup> partie – Metro Style Apps

**8. Prochaine séance**

Mercredi 13 juin 2012 à 14h00 à Bellevue

Sierre, le 1<sup>er</sup> juin 2012

## 11.5.7 PV de la séance du 13.06.12 – N°07

### Présent :

M. Jean-Pierre Rey

Mme Sonia Ragosta Foti

### Lieu :

Immeuble Bellevue - Sierre

### 1. Approfondissement du rapport 4<sup>ème</sup> partie

Il serait intéressant d'approfondir l'impact de Metro Style Apps et Metro sur le design du contenu, c'est-à-dire la grandeur des écrans des divers outils (Tablette – Téléphone – Ecran ...) est différente donc qu'elle sera la manière de développer?

Autres points à approfondir :

- Quel est le lien entre Silverlight et WPF et WinRT?
- Est-ce que Microsoft propose quelque chose pour passer de Silverlight à HTML5 (Metro) pour les web Services?

### 2. Planning proposé

Le prochain livrable fixé est :

Quelques print screen d'application en Silverlight et HTML5.

1<sup>ère</sup> échéance : Mercredi 20 juin 2012.

### 3. Machine virtuelle

La machine que David Russo m'a fourni n'était pas compatible avec mon ordinateur.

M. David Russo m'a transmis un lien pour créer une machine virtuelle à 32 bits.

### 4. A remettre

- Des exemples de codage.

### 5. Prochaine séance

Mercredi 27 juin 2012 à 13h30 à Bellevue

Sierre, le 14 juin 2012

## 11.5.8 PV de la séance du 27.06.12 – N°08

### Présent :

M. Jean-Pierre Rey

Mme Sonia Ragosta Foti

### Lieu :

Immeuble Bellevue - Sierre

### 1. Objectifs de départ

M. Rey a repris les objectifs qui avaient été fixé au départ du travail de Bachelor.

Il en sort différents points :

- Compléter la partie MVVM et créer un exemple
  - Utilisation de PRISM
- Reprendre la documentation et mettre à jour les différentes parties
- Mettre de l'ordre dans la documentation
- Finaliser le code
- Sauvegarder les applications

### 2. Planning proposé

Le prochain livrable fixé est :

Recherche sur :

- PRISM + exemple
- OData – Services disponibles
- Application pour Windows Phone 7
- Faire des recherches : quelles technologies développer pour quel type de scénario ?

### 3. A remettre

- Rapports OData et PRISM
- Développer les points 1 – 3 – 4 – 6 du Guideline Microsoft : patterns – practices – Upcoming Release Roadmap (<http://msdn.microsoft.com/en-us/practices/bb232643>).

#### 4. Remarques

- M. Rey sera en vacances du 14 juillet 2012 au 20 août 2012, mais il reste à disposition en cas de problème.
- La défense a été préfixée au 29 août 2012. Attendre la confirmation.

#### 5. Prochaine séance

Mercredi 11 juillet 2012 à 15h00 à Bellevue, si nécessité.

Sierre, le 28 juin 2012

## 11.6 Rapports hebdomadaires

### 11.6.1 Semaines du 9 au 22 avril 2012

#### Semaine 1

Activités	Plan		du 09.04.2012 au 15.04.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	0														
Silverlight 3 - Tutoriaux	TP	14														
	TE	14			3		3		2	4	2					
Silverlight - Lecture	TP															
	TE	2.5									1				1.5	
Mini-rapport - Rédaction	TP	0.5														
	TE	0														
Silverlight - Microsoft Blend	TP															
	TE	2					2									
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
Temps total planifié		14.5														
TTE		18.5														

Suite des Tutoriaux Silverlight Version 3.

#### Problèmes rencontrés :

Gestion des VM.



## Semaine 2

Activités	Plan		du 16.04.2012 au 22.04.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	14														
Mini-rapport - Rédaction	TP	0.5														
	TE	0.5									0.5					
Tutoriaux - Silverlight	TP	4														
	TE	6		2				4								
Modification - Cahier des charges	TP	0.5														
	TE	0.5								0.5						
Application des concepts SL	TP	4														
	TE	3									3					
Rédaction rapport - 1 <sup>ère</sup> partie	TP	4														
	TE	0														
PV Séance	TP															
	TE	0.5								0.5						
Planification du projet	TP	1.0														
	TE	1								1						
Environnement de test	TP															
	TE	3									3					
Temps total planifié		14														
	TTE	14.5														

### Délivrable :

- mini-rapports
- Cahier des charges corrigé
- Planning prévisionnel

Suite des tutoriels.

Mise en place d'un environnement pour le testing.

Recherche d'informations pour la rédaction de la première étape.

### Problèmes rencontrés :

La gestion des VM. Je crois que le problème vient de l'ordinateur.

### Délivrables fournis

- Cahier des charges corrigé
- Planning prévisionnel
- Rapports hebdomadaires

## 11.6.2 Semaines du 23 avril au 6 mai 2012

### Semaine 1

Activités	Plan		du 23.04.2012 au 29.04.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	14														
Rédaction rapport - 1 <sup>ère</sup> partie	TP	3														
	TE	8							3	2	3					
Application des concepts SL	TP	9														
	TE	8		3		5										
Concept MVVM	TP	2														
	TE	2						1			1					
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
Temps total planifié		14														
	TTE	18														

Cette semaine s'est déroulée sans trop de problèmes. J'avais prévu de rendre la première partie du rapport pour la fin de la semaine, mais ma planification a été mauvaise.

### Problèmes rencontrés :

Gestion du temps.

## Semaine 2

Activités	Plan		du 30.04.2012 au 06.05.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	14														
Concept MVVM	TP	9														
	TE	11		1					3	3	4					
Rédaction rapport	TP	4														
	TE	0														
Séance 02.05.12	TP	1														
	TE	0														
Rédaction rapport_1	TP															
	TE	4		2		2										
Application_Modification	TP															
	TE	2				2										
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
Temps total planifié		14														
	TTE	17														

La compréhension du pattern MVVM s'annonce plus difficile que prévue. La distinction entre les différentes parties est nette, mais le ViewModel me pose problème.

### Problèmes rencontrés :

Compréhension du modèle MVVM.

### Délivrables fournis

- Rapports hebdomadaires

### 11.6.3 Semaines du 7 mai au 27 mai 2012

#### Semaine 1

Activités	Plan		du 07.05.2012 au 13.05.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	14														
Rapport - MVVM	TP	5														
	TE	3										3				
Application - MVVM	TP	9														
	TE	2										2				
MVVM - Théorie	TP															
	TE	7			2		1		1	2	1					
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
Temps total planifié		14														
	TTE	12														

Délivable :  
 - Application SL - MVVM  
 - Rapport - Concept MVVM

#### Problèmes rencontrés :

Le concept MVVM a été difficile à cerner.

#### Semaine 2

Activités	Plan		du 14.05.2012 au 20.05.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	14														
Rapport - MVVM	TP	9														
	TE	3		2			1									
Application - MVVM	TP	5														
	TE	0														
PV	TP															
	TE	1					1									
Positionnement de MS	TP															
	TE	5									3	2				
Séance	TP															
	TE	1					1									
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
Temps total planifié		14														
	TTE	10														

#### Problèmes rencontrés :

Les informations étaient nombreuses et parfois elles se contredisaient. Alors je devais chercher des vérifications et souvent je me perdais dans beaucoup de blabla qui ne voulaient à la fin rien dire.



## Semaine 3

Planning modifié																
Activités	Plan		du 21.05.2012 au 27.05.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	14														
Microsoft - > Silverlight et HTML5	TP	9														
	TE	7		2		4		1								
Rapport - positionnement	TP	5														
	TE	5.5						1.5		3	1					
Tech Days 2012 - Visualiser	TP															
	TE	5									3				2	
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
Temps total planifié		14														
	TTE	17.5														

**Délivrable :**  
 -PV  
 - Rapport 3ème partie  
 - Mini-Rapport

## Problèmes rencontrés

Idem que la semaine 2

## Délivrables

- PV remis le 24.05.2012
- Rapport positionnement de Microsoft
- Mini-Rapports

## 11.6.4 Semaines du 28 mai au 10 juin 2012

### Semaine 1

Activités	Plan		du 28.05.2012 au 03.06.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	14														
Microsoft -> Silverlight et HTML5	TP	7														
	TE	0														
Rapport - positionnement	TP	7														
	TE	9			5		4									
Tech Days 2012 - Visualiser	TP															
	TE	1				1										
Séance	TP															
	TE	1						1								
Recherche Doc - Metro Style	TP															
	TE	3										1	2			
Lecture docs reçus	TP															
	TE	3											3			
PV séance du 30.05.2012	TP															
	TE	1										1				
	TP															
	TE	0														
Temps total planifié		14														
	TTE	18														

### Problème rencontré :

Rien à signaler

### Semaine 2

Planning modifié																
Activités	Plan		du 04.06.2012 au 10.06.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	24														
Microsoft -> Silverlight	TP	12														
	TE	0														
Rapport - positionnement	TP	12														
	TE	0														
Microsoft -> Metro - Rapport	TP															
	TE	12		1		1		1			2	2	1	4		
Recherche informations (Metro)	TP															
	TE	5	1		1			3								
Les Web Services	TP															
	TE	3				1		1			1					
	TP															
	TE	0														
	TP															
	TE	0														
	TP															
	TE	0														
Temps total planifié		24														
	TTE	20														

### Problème rencontré :

Il est difficile de trouver des exemples de Web Service avec HTML5.

Les premières recherches me renvoient sur de la programmation avec JQuery.

Les premiers essais de codage ne sont pas satisfaisants.

J'espère pouvoir présenter des exemples à la séance de mercredi 13.06.12.

### Délivrable :

- PV de la séance du 30.05.2012
- Mini-rapports









## 11.6.7 Semaines du 9 juillet au 22 juillet 2012

### Semaine 1

Planning modifié																
Activités	Plan		du 09.07.2012 au 15.07.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	24														
Développement 3D - Concept	TP	12														
	TE	0														
Développement 3D - Application	TP	12														
	TE	0														
	TP															
	TE	0														
OData - Recherche info	TP															
	TE	4		2		1								1		
Windows Phone - Application	TP															
	TE	1												1		
Visual Studio Light Switch	TP															
	TE	6		1								2	3			
TechEd 2012 - Visualisation vidéo	TP															
	TE	7				3		2				1		1		
	TP															
	TE	0														
Temps total planifié		24														
	TTE	18														

De nouvelles informations apparaissent tous les jours.

Immersive Windows est le nom que donne Microsoft pour les applications Metro Style.

### Problèmes rencontrés :

RAS

## Semaine 2

Planning modifié																
Activités	Plan		du 16.07.2012 au 22.07.2012													
			L		M		M		J		V		S		D	
			M	A	M	A	M	A	M	A	M	A	M	A	M	A
	TTP	24														
Développement 3D - Concept	TP	12														
	TE	0														
Développement 3D - Application	TP	12														
	TE	0														
Odata Rapport	TP															
	TE	6							2		1	3				
RoadMap Microsoft	TP															
	TE	1		1												
Rapport final	TP															
	TE	10				3		1		3	2	1				
Application WP	TP															
	TE	2		1							1					
Application Mutliplateforme	TP															
	TE	2										2				
	TP															
	TE	0														
Temps total planifié		24														
	TTE	21														
			Délivable :													
			- Mini-rapports													
			OK													

Test pour la création d'une application multiplateforme (Metro – Silverlight – WP).

### Problèmes rencontrés :

Problème à l'installation de Windows Phone 7.1.1 Update.

### Délivable :

- Rapports hebdomadaires