
Utilisation du framework Wt pour le passage d'une application C++/Qt vers une application Web.

Laurent Garnier

22 Octobre 2012

8èmes journées Informatique IN2P3-IRFU
La londe les maures



Sommaire

-  ☐ Faire une application Web à : Pourquoi ?
-  ☐ L'application existante
-  ☐ Modèle MVC
-  ☐ Solutions potentielles
-  ☐ Wt ?
-  ☐ Wt en pratique



Pourquoi ???

📌 C'est une application qui tourne sur tout les browser Web (Safari, IE, Firefox)

📌 Et donc également sur les tablettes et téléphones

📌 **Avantages :**

- ▶ Disponible partout dès lors qu'il y a une internet
- ▶ Aucun problèmes de compatibilité Windows/Mac/Linux (*)
- ▶ Même interface graphique sur toutes les plates-formes (*)
- ▶ Rien à installer, rien à compiler
- ▶ **N'importe qui peut l'utiliser, n'importe où**
- ▶ (*) : Sauf exceptions

📌 **Inconvénients :**

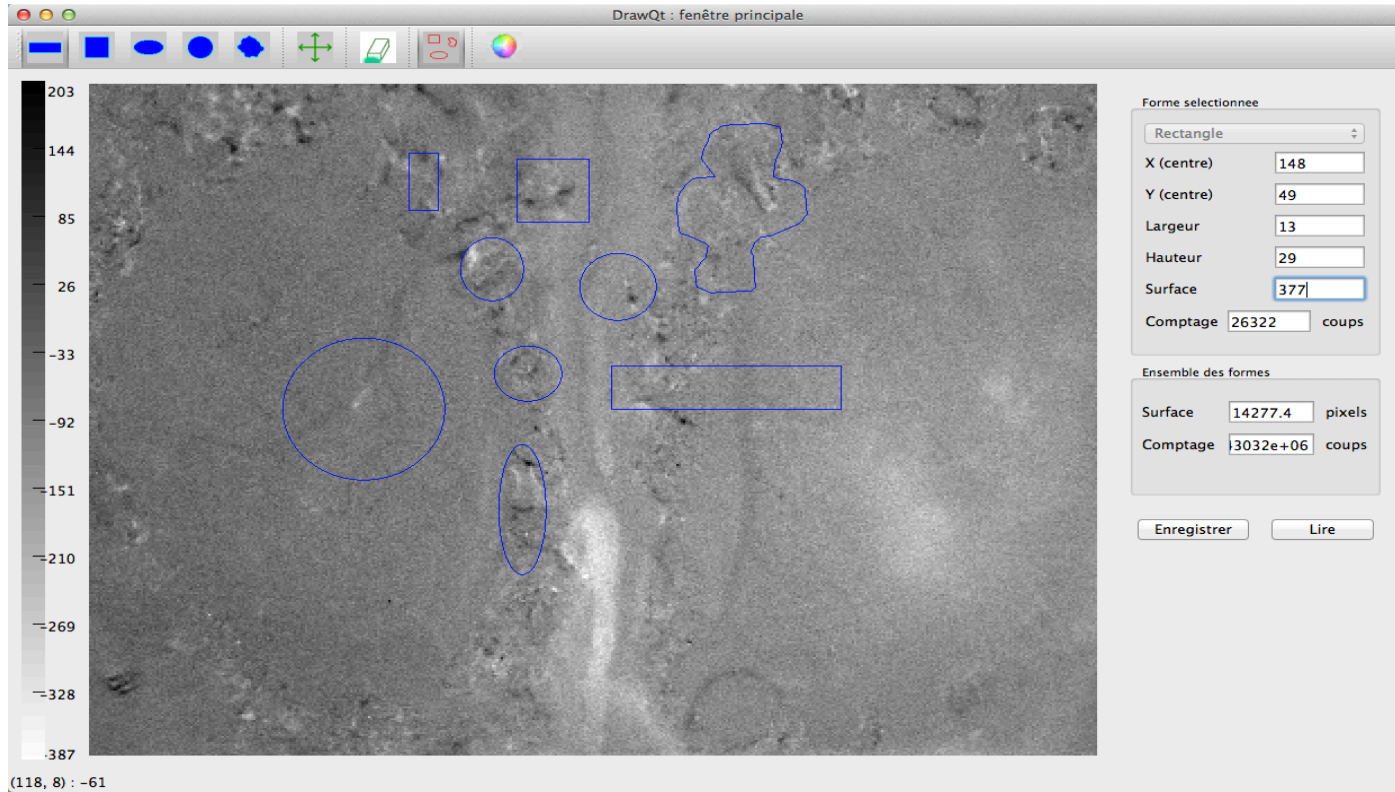
- ▶ Utilisation des langages de programmation natifs Web: HTML, Javascript ou via des plugins: Flash, Direct X, Java, ...

Quelle est l'application existante :

- Application utilisée dans le cadre de TP informatique en enseignement au LAL
- Extrait d'un logiciel développé pour POCI.
 - ▶ Sonde per-opératoire POCI : gamma-caméra
 - ▶ Aide au chirurgien dans le cadre du cancer du sein
 - ▶ Repérage des ganglions marqués par un marqueur radioactif
- environ 20 fichiers (7000 lignes de code) écrits en C++
- partie interface utilisateur écrite avec le framework Qt



DrawQt : Présentation



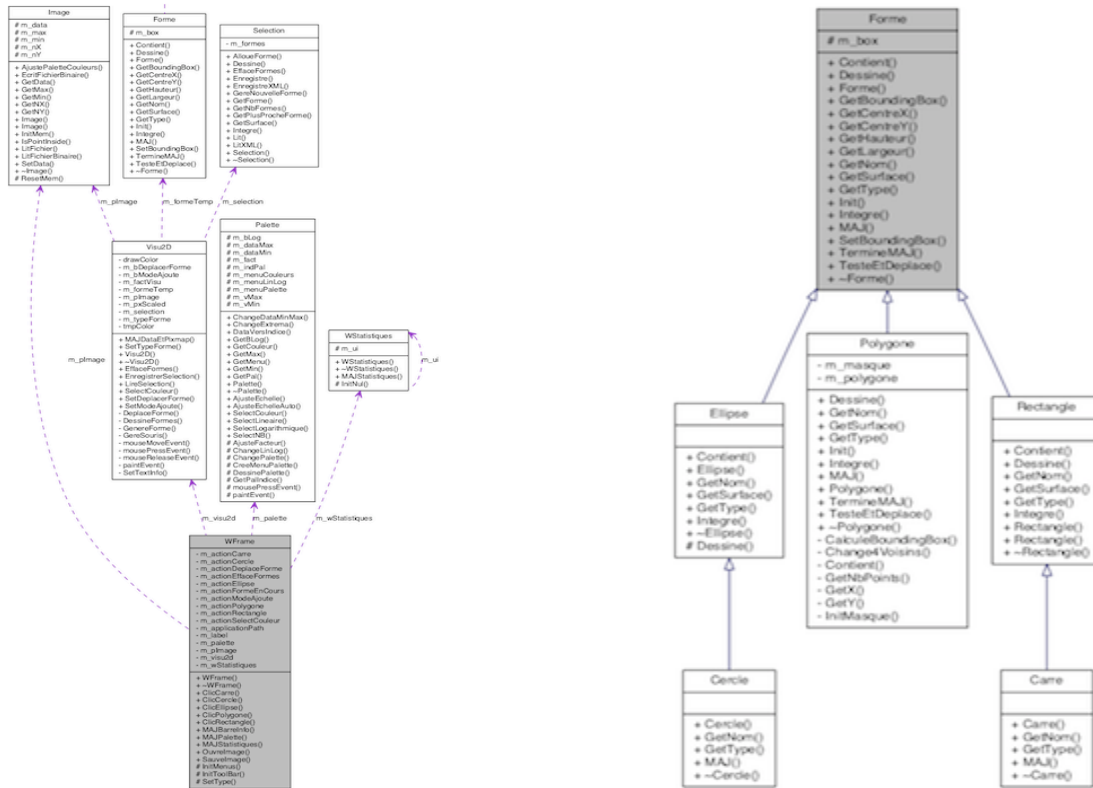


Quel sont nos contraintes

- **Ne pas refaire l'application de A à Z !**
- Faire une application ayant l'interface graphique la plus proche de l'application native
- Avoir la même interactivité
- Pour cela nous avons besoin d'outils :
 - ▶ Les outils "traditionnels" de développement Web sont orientés "pages"
 - ▶ Nos outils doivent prendre en compte les différents browser web, smartphones et tablettes
 - ▶ Etre sécurisé
- S'appuyer au maximum sur le code existant



Modèle existant : Diagramme de classes

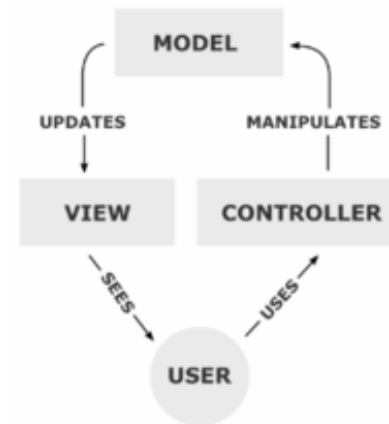
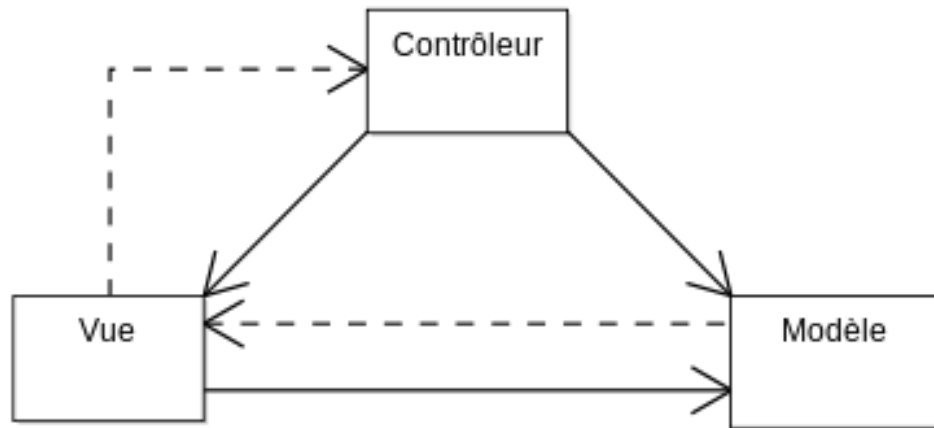




Modèle MVC ?

MVC : Modèle-Vue-Contrôleur

Patron de conception destiné à répondre aux besoins des applications interactives en séparant la partie représentation (Vue) de la partie gestion des données (Modèle).





Le modèle :

- ▶ Représente le coeur de l'application : Traitement des données, interaction avec les bases de données, ...



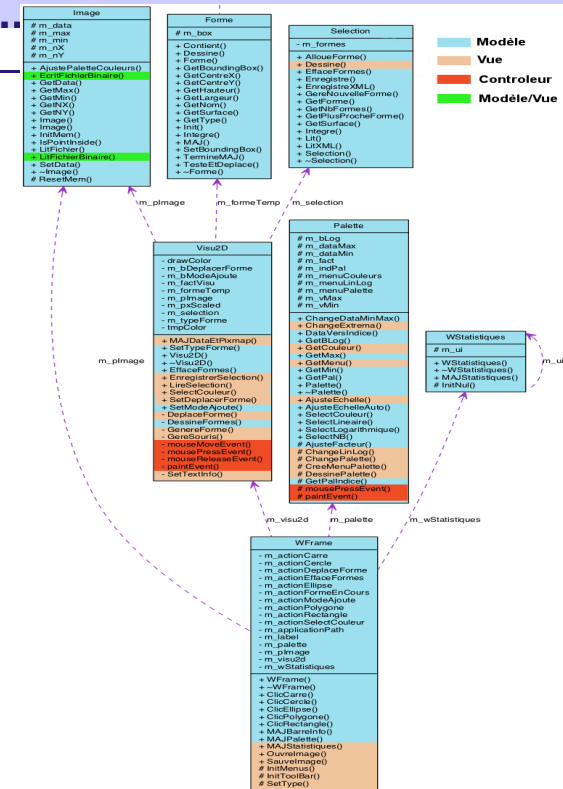
La vue :

- ▶ Composante "présentation" de l'application, peut-être graphique/web/console/...
- ▶ Ce avec quoi l'utilisateur interagit par des événements : clic souris/touches/trackpad/...
- ▶ Plusieurs vues sont possibles au sein d'une même application, par exemple dans le cas d'une application multi-plateformes



Le contrôleur :

- ▶ Prend en charge la gestion des événements pour mettre à jour la vue et le modèle
- ▶ Reçoit les événements utilisateur et lance les actions à effectuer





Quelles solutions ?

- Tout réécrire "from scratch" en Web 2.0 ?
- Utiliser un framework existant avec le moins de changements possibles pour notre code
- Quelques exemples de framework (merci wikipedia)



Quelles solutions possibles ?

ASP.NET

See also: [ASP.NET](#) and [List of CLI languages](#)

Project	Language	Current stable version	Release date	License
ASP.NET MVC Framework (ASP.NET MVC)	ASP.NET	4.0 Beta	2012-02-15	Apache v2 ^[1]
Base One Foundation Component Library (BFC)	ASP.NET	7.43	2011-01-15	Proprietary
Component-based Scalable Logical Architecture (CSLA)	C#, Visual Basic .NET	4.3.10	2012-03-04	Proprietary freeware, no limit on use type
DotNetNuke	ASP.NET	6.01.02	2011-07-25	MIT
Kentico	ASP.NET	6.0	2011-10-10	Proprietary
MonoRail	ASP.NET	2.1 RC	2011-03-17	Apache
OpenRasta	ASP.NET	2.0 RC	2009-11-04	MIT
Nancy	C#, Visual Basic .NET	0.6.0	2011-06	MIT

C++

See also: [C++](#)

Project	Current stable version	Release date	License
CppCMS	1.0.2	2012-08-14	LGPL, Proprietary
Tntnet	2.1	2012-04-02	LGPL
Wt	3.2.2-p1	2012-07-27	GPL, proprietary

ColdFusion Markup Language (CFML)

See also: [ColdFusion Markup Language](#) and [ColdFusion](#)

Project	Current stable version	Release date	License
ColdBox Platform	3.5.2	2012-08-01	Apache v2
ColdFusion on Wheels	1.1.7	2011-12-14	Apache v2
ColdSpring	1.2	2008-09-12	Apache v2
Fusebox	5.5.1	2008-03-08	Apache
FW/1	1.2	2010-10-18	Apache v2
Mech-II	1.8.1	2010-09-20	GPLv3 with Classpath exception
Model-Glue	3.1.185	2009-08-11	Apache
onTap	3.3	2009-11-07	BSD



Quelles solutions possibles ?

ECMAScript

See also: [ECMAScript](#)

Project *	Language *	Current stable version *	Release date *	License *
Flex	ActionScript, MXML	4.8	2012-07-25	Apache

Java

See also: [Java \(programming language\)](#) and [Java \(software platform\)](#)

Project *	Current stable version *	Release date *	License *
Apache Click	2.3.0	2011-03-27	Apache Software License 2.0 (ASL 2.0)
Apache OFBiz	10.04.02	2010-04-30	Apache Software License 2.0 (ASL 2.0)
Apache Shale	1.0.4	2007-12-19	Apache
Apache Sling	6	2011-04-18	Apache 2.0
Apache Struts	2.3.4	2012-05-12	Apache 2.0
Apache Tapestry	5.3.5	2012-08-30	Apache
Apache Wicket	6.0	2012-09-05	Apache
AppFuse	2.1	2011-04-04	Apache
Eclipse RAP	1.4.1	2011-09-23	Eclipse
FormEngine	2.0.1	2012-05-08	MIT , Proprietary
Grails	2.0.3	2012-04-03	Apache
Google Web Toolkit	2.4.0	2011-09-08	Apache
Hamlets	1.7	2011-03-11	BSD
ItsNat	1.2	2011-05-24	LGPL , proprietary
JavaServer Faces	2.1	2010-10-22	Implementation dependent
JBoss Seam	3.1.0 final	2012-01-13	LGPL
Jspix-bay	1.1.0	2011-02-17	Apache 2.0
JVx WebUI	0.9	2011-04-01	Apache 2.0
ManyDesigns Portofino	3.1.10	2011-03-22	GPLv3
OpenLaszlo	4.9.0	2010-10-21	CPL
OpenXava	4.5	2012-07-19	LGPL
Oracle ADF	11.1.1.5	2011-05-06	Oracle Technology Network Developer License
Play!	2.0.3	2012-08-03	Apache 2.0
RIFE	1.6.1	2007-07-14	CDDL , LGPL
SmartClient	8.0	2011-01-05	LGPL
Sonic	3.1.2	2012-07-05	Apache



Quelles solutions possibles ?

JavaScript

See also: [JavaScript](#)

Project *	Current stable version *	Release date *	License *
multiple frameworks based on node.js	0.8.9	2012-07-11	MIT
qooxdoo	1.5	2011-07-13	LGPL or Eclipse
SproutCore	1.8 (beta 2.0)	2012-03-07	MIT
WaveMaker (client side)	6.3.3	2011-09-22	Apache

Scala

See also: [Scala \(programming language\)](#)

Project *	Current stable version *	Release date *	License *
Play!	2.0.3	2012-08-03	Apache 2.0
Lift	2.4	2012-01-12	Apache 2.0

Perl

See also: [Perl](#)

Project *	Current stable version *	Release date *	License *
Catalyst	5.90012	2012-05-19	Artistic, GPL
Dancer	1.3095	2012-04-01	Artistic, GPL
Mason	2.19	2012-05-02	Artistic, GPL
Maypole	2.13	2008-04-18	Artistic, GPL
Mojolicious	3.0	2012-06-26	Artistic
Reaction	0.002005	2011-03-08	Artistic, GPL



Quelles solutions possibles ?

PHP

See also: *PHP*

Project	Start date	Current stable version	Release date	Framework type	License
Agevi	2005-05	1.0.7	2011-12-22	Web application framework	LGPL
Alki Framework	2009-09	0.9.0	2011-12-24	Web application framework	AGPLv3
AppFlower	2009-02	1.2	2011-12-21	Web application framework	GPLv3
CakePHP	2005-08	2.2.3	2012-10-08 ^[2]	Web application framework	MIT
CodeIgniter	2006-01	2.1.1	2012-06-13 ^[3]	Web application framework	OSLv3
Cotonti	2009-02	0.9.7	2012-02-01	Content management framework	BSD
Directus	2006-04	5.1	2012-03-20	Content management framework	GPL
Drupal	2000-05	7.15	2012-08-01	Content management framework	GPLv2 or later
e107	2004-11	1.0.1	2012-03-30	Web application framework	GPLv2
eZ Publish	2001-01	4.7	2012-05-15	Content management framework	GPLv2, proprietary license
Flow3	2007-01	1.1	2012-05-15	Web application framework	LGPLv3
FuelPHP	2010-10	1.2.1	2012-06-03	Web application framework	MIT
Fusebox	1997-11	5.5.1	2008-03-01	Web application framework	Apache
Horde	2002-02	4.0.8	2012-05-29 ^[4]	Web application framework	GPL, LGPL, BSD, Apache
Joomla! Platform	2005-09	12.1	2012-05-09	Content management framework	GPLv2
Kajona	2004-06	3.4.2	2012-07-18 ^[5]	Content management framework	LGPL
Laravel	2011-06	3.2.7	2012-09-03 ^[6]	Web application framework	MIT
Lithium	2009-10	0.10	2011-06-18	Web application framework	BSD
Midgard	1999-05	10.05.06	2012-03-06 ^[7]	Content management framework	LGPL
MODX	2005-03	2.2.2-pl	2012-05-02 ^[8]	Content management framework	GPL
Qcodo	2005-10	0.4.22	2011-08-15 ^[9]	Web application framework	MIT
SilverStripe (Sapphire)	2005-11	2.4.5	2011-02-02	Content management framework	BSD
Seagull	2003-10	0.6.8	2010-12-03	Web application framework	BSD
Symfony	2005-10	2.0.16	2012-07-11	Web application framework	MIT
Xyster Framework	2007-09	02 Build 01	2010-10-18	Web application framework	BSD
Yii	2008-01	1.1.12	2012-08-19	Web application framework	New BSD
Zend Framework	2006-03	2.0.0	2012-09-05 ^[10]	Web application framework	New BSD
Zeta Components (formerly eZ Components)	2005-11	eZ Components 2009.2.1	2010-03-22 ^[11]	Web application framework	Apache



Quelles solutions possibles ?

Python

See also: [Python \(programming language\)](#)

Project	Current stable version	Release date	License
BlueBream	1.0	2011-01-18	ZPL
CherryPy	3.2.2	2011-10-28	BSD
CubicWeb	3.15.2 ^[12]	2012-07-24	LGPL
Django	1.4.1	2012-07-30 ^[13]	BSD
Flask	0.9	2012-07-01 ^[14]	BSD
Grok	1.4	2011-04-04	ZPL
Nagare	0.4.1	2012-01-18	BSD
Plone	4.2	2012-07-05	GPL
Pyjamas	0.8	2012-06-01	Apache
Pylons	1.0.0	2010-05-28	BSD
Pyramid	1.3	2012-03-21	BSD
Tornado	2.3 ^[15]	2012-05-31	Apache
TurboGears	2.2	2012-08-23	MIT , LGPL
web2py	1.99.7	2012-03-04 ^[16]	LGPLv3
Webware	1.1	2011-08-03	Python
Zope 2	2.13.15	2012-06-22 ^[17]	ZPL

Ruby

See also: [Ruby \(programming language\)](#)

Project	Current stable version	Release date	License
Camping	2.1	2010-08-20	MIT
Ruby on Rails	3.2.8	2012-08-09	MIT , Ruby
Ramaze	2011.07.25	2011-07-25	Ruby
Sinatra	1.3.3	2012-08-19	MIT
Rango	0.2	2009-03	
Merb	1.1.2	2010-06-17	MIT
Nitro	0.41	2006-12-12	BSD
PureMVC	2.0.4	2008-08-14	CC +Attribution



Quelles solutions possibles ?

Others

Project *	Language *	Current stable version *	Release date *	License *
AIDA/Web	Smalltalk	6.5	2012-08-27	MIT
Application Express	PL/SQL	4.2	2012-10-15	Proprietary freeware
Grails	Groovy (JVM)	2.0.1	2012-02-14	Apache
Kepler	Lua	1.1.1	2009-03-11	MIT
Morfik	C#, BASIC, Pascal, Java	3.0.8.1	2010-12-13	Proprietary freeware
Noir	Clojure	1.2.2	?	Eclipse
Ocsigen	OCaml	2.0.4	2012-03-09	LGPL
Opa	Opa	1.0.5	2012-08-02	MIT and AGPL
OpenACS	Tcl	5.7.0	2011-07-17	GPL
Seaside	Smalltalk	3.0	2010-09-13	MIT
Compojure	Clojure	1.0.1	2012-01-08	Eclipse
Zotonic	Erlang	0.8.1	2012-08-11	Apache



Beaucoup de candidats potentiels !

PHP, Javascript, Python, JAVA, Perl, Ruby, ... C'est bien, mais...

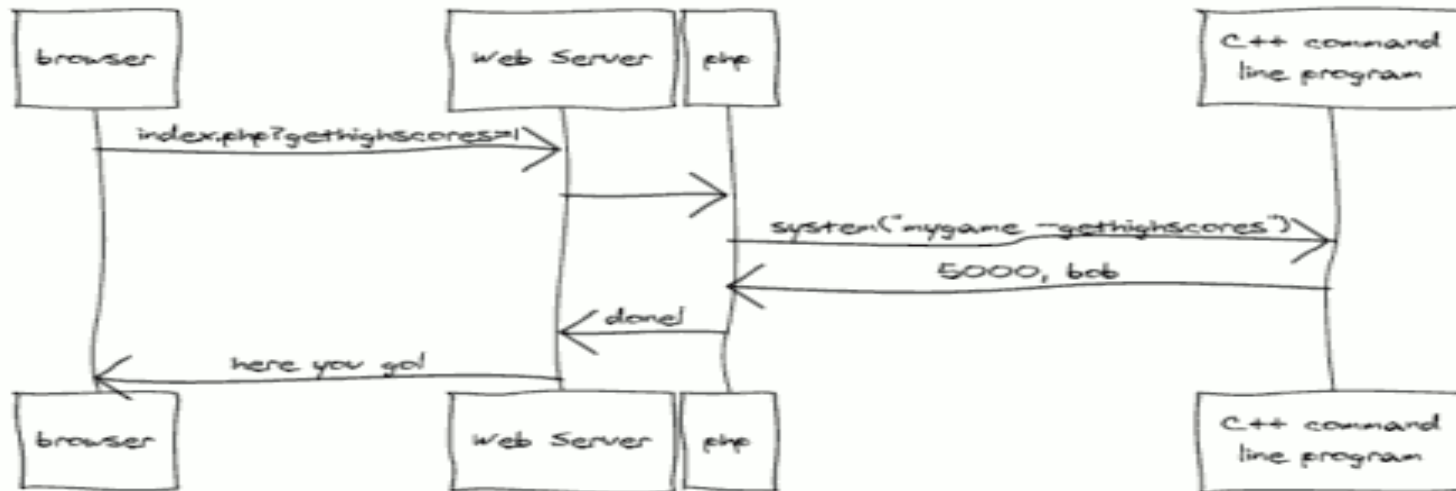
Pourquoi ne pas utiliser directement un framework en C++ ??

C++?!/



Navigateur Web -> C++ (ligne de commande ?)

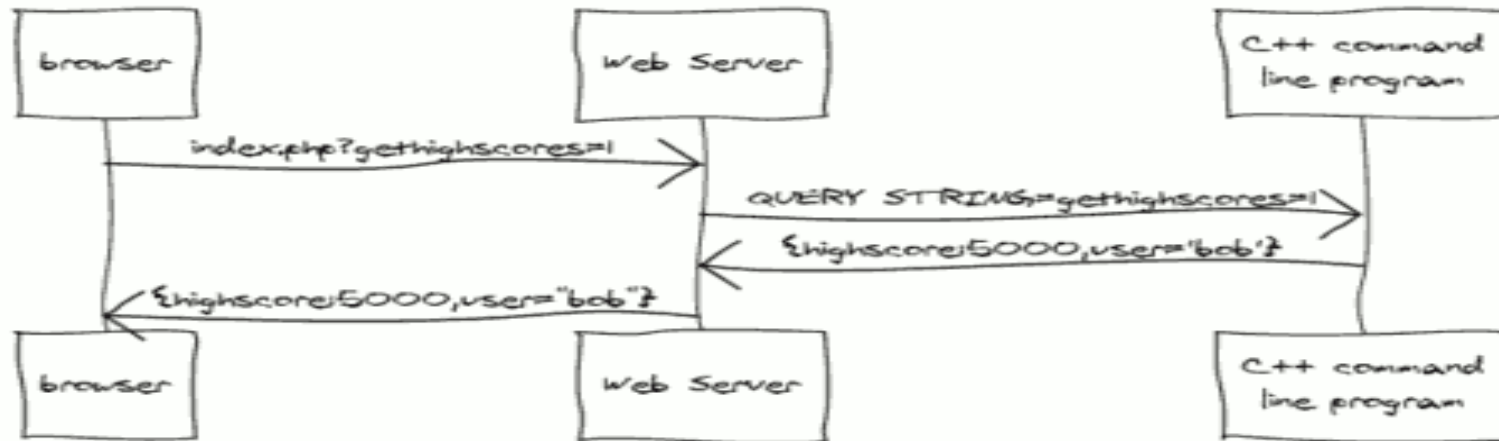
Call a command line program from server program





Navigateur Web -> C++ (script CGI ?)

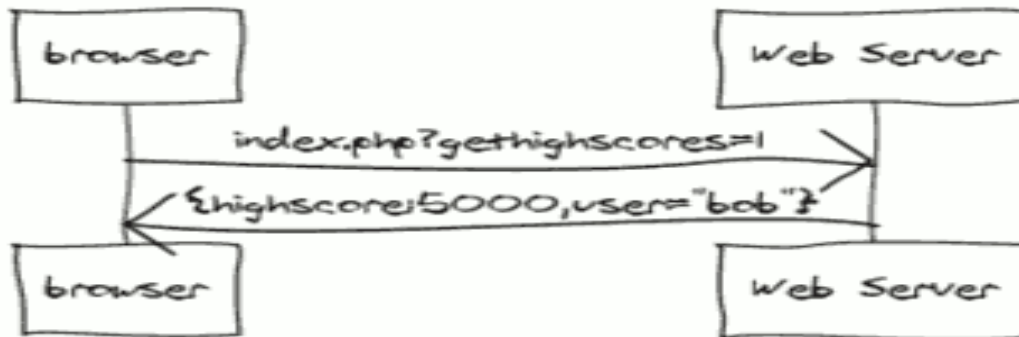
C++ CGI script called directly from web page





Navigateur Web -> C++ (Intégration du serveur Web dans l'appli C++ ?)

C++ program integrated with web server





Wt, un framework Web en C ++

Wt est une bibliothèque de classes C++ pour développer des applications Web

- ▮ Contient tout le nécessaire pour faire toute la "Vue" du modèle MVC, orientée Web
- ▮ Nombreux composants graphiques, composants interactifs, gestion des feuilles de style, layout
- ▮ Mécanisme de **signal/slot** pour la gestion des événements
- ▮ Composants de dessin, intégration de composants **WebGL**, ...
- ▮ Utilise la librairie boost
- ▮ Serveur Web **s~~é~~curis~~é~~** intégré dans l'application
- ▮ **Tout les composants de la bibliothèques Wt sont le reflet de la bibliothèque Qt !**
Pour dessiner un Bouton dans Qt: **QPushButton** , dans Wt: **WPushButton** !



Wt dans la vraie vie

- Très peu de travail pour passer une application écrite en C++ avec le framework Qt
- => Remplacer les "**Q**" des classes Qt par des "**W**", c'est "presque" tout :-)
- Possibilité de mixer du Qt/Wt sans soucis pour toutes les classes non graphiques



DrawWt : classe Rectangle

```
Rectangle::Rectangle() : Forme()
{
}

Rectangle::Rectangle(BoundingBox& box)
{
    m_box = box;
}

Rectangle::~~Rectangle()
{
}

TYPE_FORME Rectangle::GetType() const{
    return RECTANGLE;
}

std::string Rectangle::GetNom() const {
    return "RECTANGLE";
}

void Rectangle::Dessine(QPainter* p, int fact)
{
    p->drawRect(m_box.GetOrigineX() * fact,
                m_box.GetOrigineY() * fact,
                (m_box.GetLargeur()) * fact,
                (m_box.GetHauteur()) * fact);
}

double Rectangle::Integre(Image* pData)
{
    double s = 0;
    for (int y = m_box.GetOrigineY(); y < m_box.GetFinY(); y++)
        for (int x = m_box.GetOrigineX(); x < m_box.GetFinX(); x++)
            s += (double)pData->GetData(x, y);

    return s;
}

double Rectangle::GetSurface() const
{
    return (double)(GetLargeur() * GetHauteur());
}

bool Rectangle::Contient(int x, int y) const
{
    return m_box.Contient(x, y);
}
```

→ 1

```
Rectangle::Rectangle() : Forme()
{
}

Rectangle::Rectangle(BoundingBox& box)
{
    m_box = box;
}

Rectangle::~~Rectangle()
{
}

TYPE_FORME Rectangle::GetType() const{
    return RECTANGLE;
}

std::string Rectangle::GetNom() const {
    return "RECTANGLE";
}

void Rectangle::Dessine(WPainter* p, int fact)
{
    p->drawRect(m_box.GetOrigineX() * fact,
                m_box.GetOrigineY() * fact,
                (m_box.GetLargeur()) * fact,
                (m_box.GetHauteur()) * fact);
}

double Rectangle::Integre(Image* pData)
{
    double s = 0;
    for (int y = m_box.GetOrigineY(); y < m_box.GetFinY(); y++)
        for (int x = m_box.GetOrigineX(); x < m_box.GetFinX(); x++)
            s += (double)pData->GetData(x, y);

    return s;
}

double Rectangle::GetSurface() const
{
    return (double)(GetLargeur() * GetHauteur());
}

bool Rectangle::Contient(int x, int y) const
{
    return m_box.Contient(x, y);
}
```



DrawWt : classe WFrame

```
/*
 * Initialisation de la barre de boutons : sélection des formes et des options relatives
 */
void WFrame::InitToolBar()
{
    QString strDirIcones = m_applicationPath+"/../Resources/data/icons";    // Pour MAC
    QDir dir(strDirIcones);
    if (!dir.exists())    // Pour Windows
        strDirIcones = "../data/icons";

    // Création d'une toolbar
    QToolBar *toolBarForme = new QToolBar(this);
    addToolBar(Qt::TopToolBarArea, toolBarForme);

    // Ajout d'un bouton rectangle
    m_actionRectangle = toolBarForme->addAction(QIcon(strDirIcones+"/Rect.bmp"), "Forme rectang", this);
    m_actionRectangle->setShortcut(tr("Ctrl+R"));
    m_actionRectangle->setCheckable(true);

    // Ajout d'un bouton carré
    m_actionCarre = toolBarForme->addAction(QIcon(strDirIcones+"/Carre.bmp"), "Forme carrée", this);
    m_actionCarre->setShortcut(tr("Ctrl+S"));
    m_actionCarre->setCheckable(true);

    // Ajout d'un bouton ellipse
    m_actionEllipse = toolBarForme->addAction(QIcon(strDirIcones+"/Ellipse.bmp"), "Forme ellipt", this);
    m_actionEllipse->setShortcut(tr("Ctrl+E"));
    m_actionEllipse->setCheckable(true);

    // Ajout d'un bouton cercle
    m_actionCercle = toolBarForme->addAction(QIcon(strDirIcones+"/Cercle.bmp"), "Forme circulai", this);
    m_actionCercle->setShortcut(tr("Ctrl+C"));
    m_actionCercle->setCheckable(true);

    // Ajout d'un bouton polygone
    m_actionPolygone = toolBarForme->addAction(QIcon(strDirIcones+"/Polygone.bmp"), "Tracé d'un polygone", this);
    m_actionPolygone->setShortcut(tr("Ctrl+P"));
    m_actionPolygone->setCheckable(true);
}
```

```
/*
 * Initialisation de la barre de boutons : sélection des formes et des options relatives
 */
void WFrame::InitToolBar(WVBoxLayout* boxR)
{
    WContainerWidget *ex = new WContainerWidget();

    Ext::ToolBar *toolBarForme = new Ext::ToolBar(ex);
    std::string strDirIcones = "icons";

    m_actionRectangle = toolBarForme->addButton(strDirIcones+"/Rect.bmp", "R", this, &WFrame::ClickRectangle);
    m_actionRectangle->setCheckable(true);

    // Ajout d'un bouton carré
    m_actionCarre = toolBarForme->addButton(strDirIcones+"/Carre.bmp", "C", this, &WFrame::ClickCarre);
    m_actionCarre->setCheckable(true);

    // Ajout d'un bouton ellipse
    m_actionEllipse = toolBarForme->addButton(strDirIcones+"/Ellipse.bmp", "E", this, &WFrame::ClickEllipse);
    m_actionEllipse->setCheckable(true);

    // Ajout d'un bouton cercle
    m_actionCercle = toolBarForme->addButton(strDirIcones+"/Cercle.bmp", "C", this, &WFrame::ClickCercle);
    m_actionCercle->setCheckable(true);

    // Ajout d'un bouton polygone
    m_actionPolygone = toolBarForme->addButton(strDirIcones+"/Polygone.bmp", "P", this, &WFrame::ClickPolygone);
    m_actionPolygone->setCheckable(true);

    // Ajout d'un séparateur
    toolBarForme->addSeparator();

    // Ajout d'un bouton déplacer
    m_actionDeplaceForme = toolBarForme->addButton(strDirIcones+"/Deplacer.bmp", "D", this, &WFrame::ClickDeplaceForme);
    m_actionDeplaceForme->setCheckable(true);
    connect(m_visu2d, &WFrame::ClickDeplaceForme, this, &WFrame::ClickDeplaceForme);

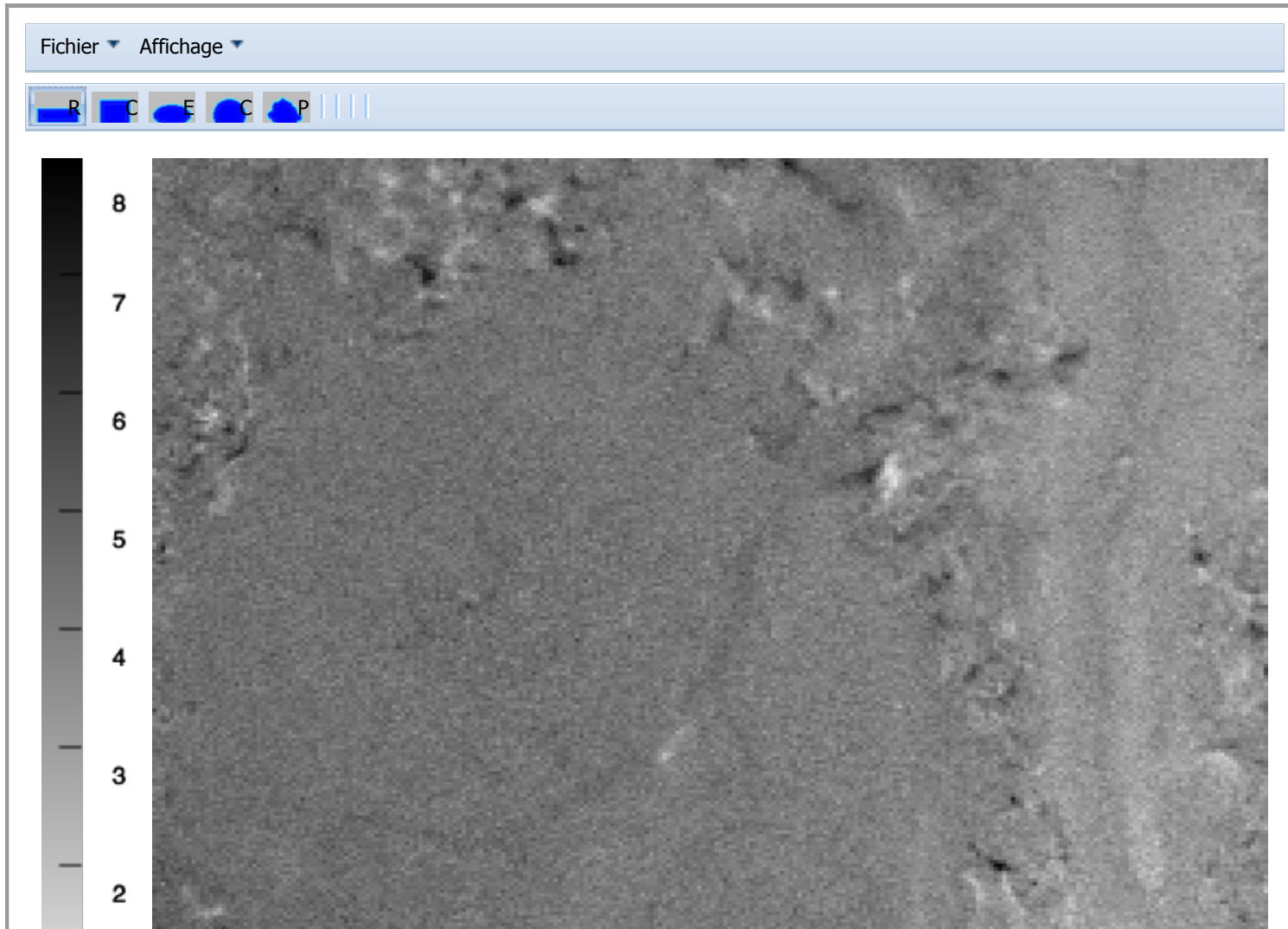
    // Ajout d'un séparateur
}
```



Wt : Comment ça marche ?

-  **Installation** : très simple, tout est sur le site web (2 packages : boost + wt)
-  **Programmation** : Wt est un framework C++, si vous connaissez le C++, vous pouvez coder votre appli web sans effort. Si en plus votre application utilise la bibliothèque Qt, encore moins d'efforts !
-  **Démarrer votre serveur Web** : La création d'une application Wt va inclure un serveur web dans votre application. Il suffit de lancer votre application et de lancer votre browser

Démo ! : ../Darwin/DrawWt.exe --docroot ../data --http-address 0.0.0.0 --http-port 8080





Références

- Wt : <http://www.webtoolkit.eu/wt/features>, [fiche plume](#)
- DrawQt ([enseignement M2 au LAL](#))
- Wikipedia : [Comparison of web application frameworks](#)