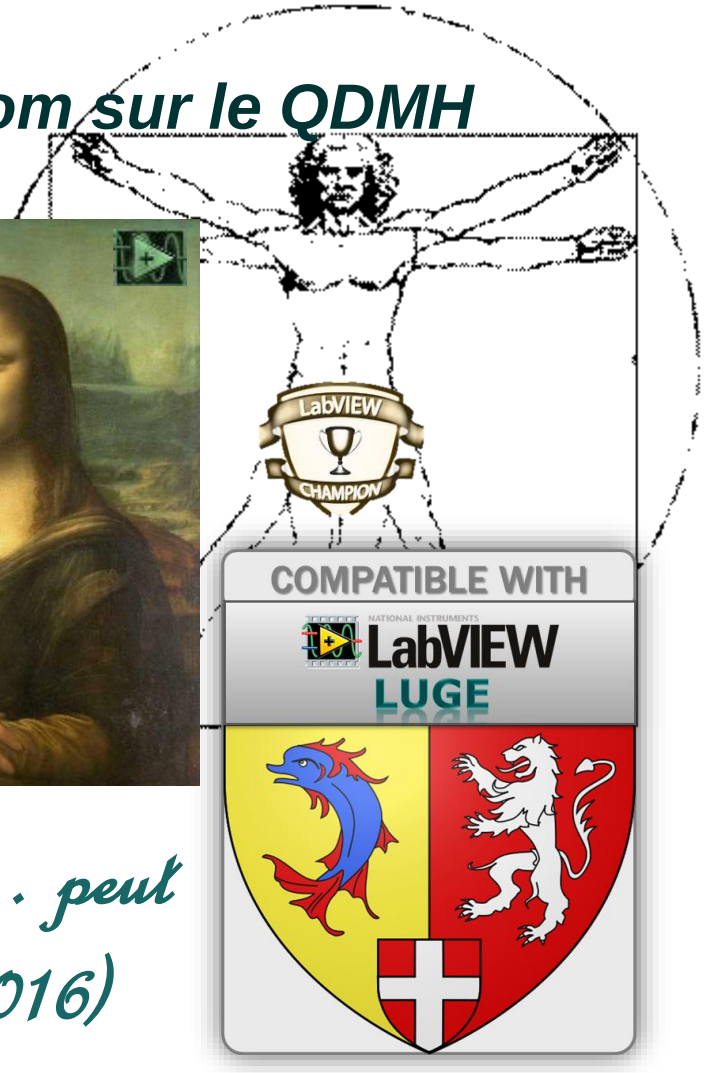
# Les modèles sous LabVIEW, zoom sur le QDMH




*Démarrer votre projet avec le bon modèle… peut tout changer (Léonard de Chambéry - 2016)*

# La société MESULOG en bref

- Activité : Développement logiciel test et mesure
- Compétences : LabVIEW (Windows, RT, PDA, DSC, FPGA),
  TestStand
  VeriStand
- Localisation : Grenoble (Moirans, 38)
- Partenaire National Instruments (2001)
- Développeurs certifiés LabVIEW et TestStand

NATIONAL INSTRUMENTS Silver Alliance Partner

NATIONAL INSTRUMENTS LabVIEW CERTIFIED ARCHITECT

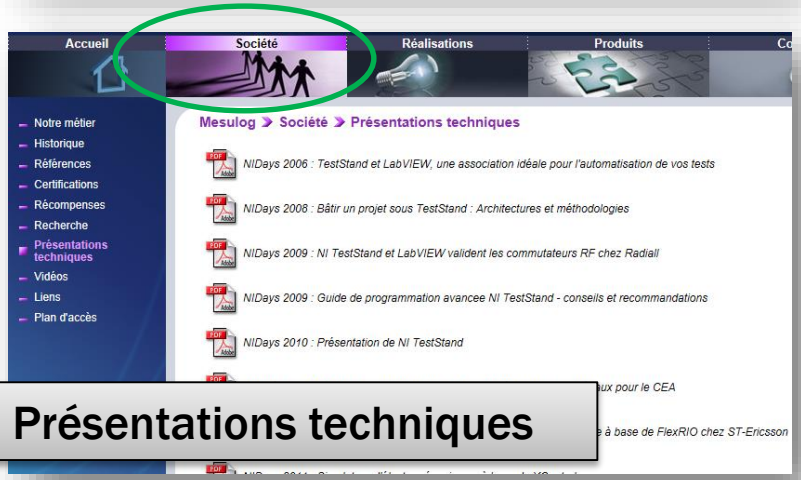
NATIONAL INSTRUMENTS CERTIFIED NI TestStand ARCHITECT

# La société MESULOG en bref

- **www.mesulog.fr**
  - Réalisations (article utilisateur)


**Réalisations**

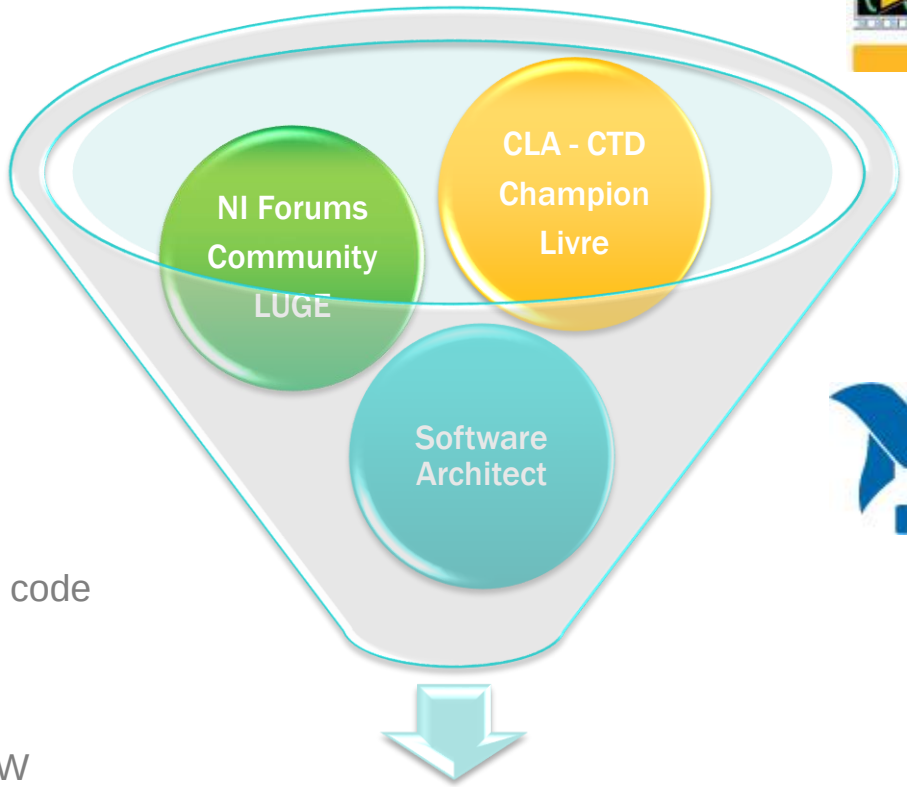
- Présentations techniques
  - LabVIEW
  - TestStand
  - VeriStand




**Société -> Présentations techniques**

# Le présentateur : Luc DESRUELLE

Desruelle_luc
Active Participant
Envois : 1 763




NATIONAL INSTRUMENTS LabVIEW CERTIFIED ARCHITECT

LabVIEW CHAMPION

NATIONAL INSTRUMENTS CERTIFIED NI TestStand DEVELOPER

Luc DESRUELLE

[Communauté NI & blog]
- Gif animé pour éviter du code
- Fenêtre LabVIEW pas rectangulaire
- Gestion IHM
- Template Projet LabVIEW
- OOP
- Modbus
- ….

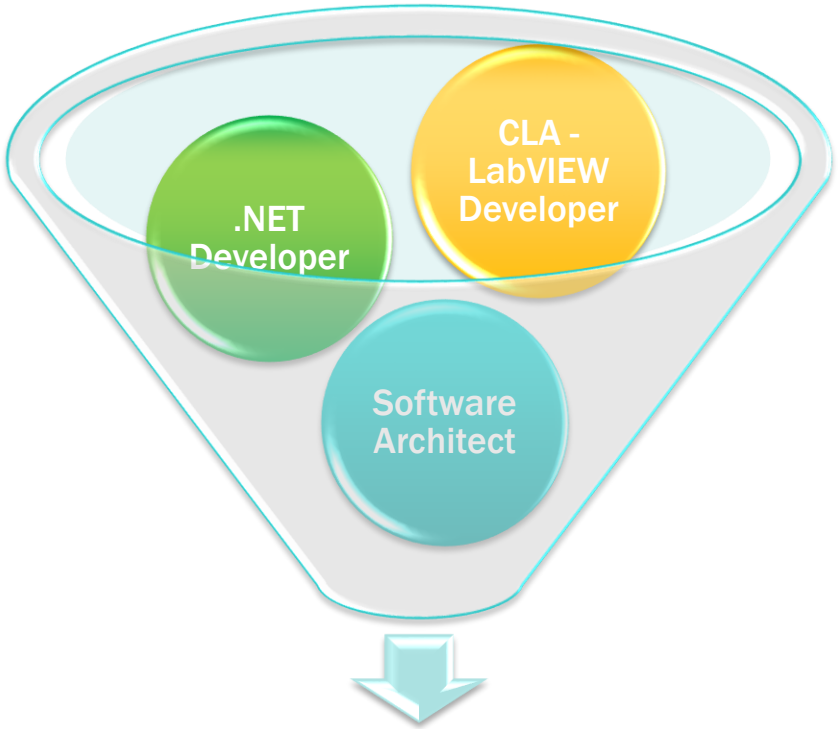

Exemples et images extraits Chapitre 3

# Le présentateur : Luc DESRUELLE

- Auteur Livre « LabVIEW : programmation et applications »
  - 3iéme Édition, Dunod 05/2015, Préface de Jeff Kodosky
  - Cette présentation est extrait du chapitre 3 du livre
  - Images et exemples ne peuvent être reproduit sans autorisation de l'éditeur
  - Lien internet du livre
    - Chapitres 1 et 2 : Prise en main de l'Environnement de développement, flux de données, code G avec des exemples simples
    - Chapitre 3 : Programmation avancée avec des techniques et architectures permettant au code d'être maintenable, évolutif, documenté et performant.
    - Chapitres 4, 5 et 6 : Acquisition, analyse et présentation des données.
    - Les + du livre
      - **Exemples concrets et tous téléchargeables gratuitement**
      - **Acquisition DAQmx, instrument VISA, traitement du signal, analyse mathématique, génération de rapport Office.**
      - **Programmation avancée, gestion des données (locale vers la DVR), variables fonctionnelles (FGV), les modèles de projet, la gestion des erreurs, les règles de styles, les outils gratuits et indispensables…**
      - **Concepts nécessaires pour l'examen Certifications LabVIEW Développeur (CLD).**

# L'assistant présentateur

Présent : NATIONAL INSTRUMENTS LabVIEW CERTIFIED ARCHITECT

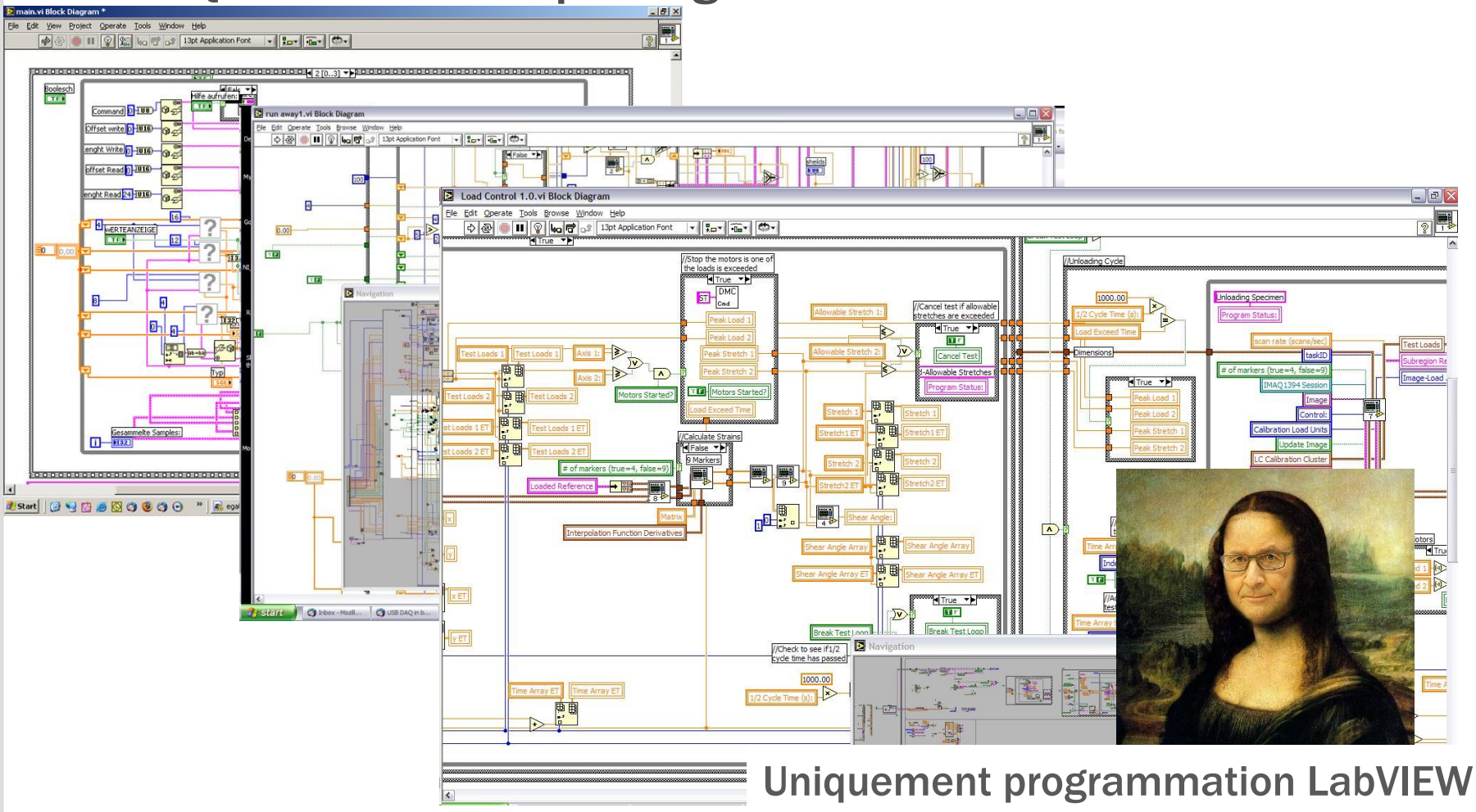



Jérémy CHIFFE

Futur:
(mais en mieux)


# Toujours le « même » constat

- Quelles sont les plus « grosses » erreurs des codes suivant ?




Uniquement programmation LabVIEW

# Le classement des erreurs « LabVIEW »

1. **Style, écriture du code**
   - Pas de style
   - Documentation


2. **Technique, Gestion des données**
   - Trop de locale – Globale
   - Flux de données

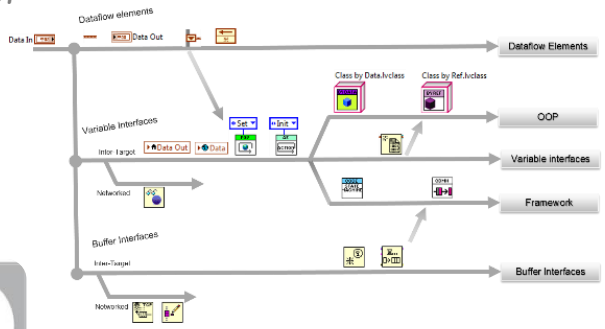
3. **Architecture, Structure application**
   - Absence modèle de conception
   - Pas gestion d'erreur, Pas gestion arrêt du logiciel


4. **Gestion Projet**
   - Version, SCC, bug, tâche

# Nous n'aborderons pas (déjà fait)


- **Nous n'aborderons pas la gestion « projet »**
  - télécharger LUGE 1.0 : Les outils qui nous veulent du bien
  - *Plus de temps pour développer en LabVIEW*

- **Nous n'aborderons pas La gestion des données**
  - LUGE 3.0 : *Contrôle vers indicateur - Locale - Globale*
  - Quand, comment, pourquoi *FGV - AE – SEQ - DVR - OOP*
  - télécharger Darwin appliqué à LabVIEW




- **Nous n'aborderons pas le « style »**
  - LUGE 5.0 (?) Avoir du style avec LabVIEW
  - Pourquoi un code est agréable?

# Nous aborderons : Architecture des applications

- **Modèle de conception - design pattern,**
  - Techniques reconnues et éprouvées *comme étant les bonnes façons de résoudre des problèmes donnés*.
  - Solutions standards - indépendantes de LV
  - Il existe « beaucoup de modèles »

- **Modèle de projet - framework**
  - Solutions codées en LabVIEW
  - LabVIEW fournit des exemples de modèles
  - Il existe « beaucoup d'exemples »

# But présentation

- **Suite discussion au LUGE 3.0**
  - Faire un point sur le modèle QDMH qui couvre un grand nombre d'application en restant abordable techniquement
  - Discussion sur Améliorations
  - Discussion sur Faiblesses / limitations
  - Continuer la discussion sur d'autres modèles connus, utilisées et avoir vos avis dessus.
  - Le but : Débuter avec le bon modèle…

# Sommaire : Zoom sur modèle QDMH

I. **Pourquoi un modèle?**
   - Vous convaincre d'en utiliser un

II. **Rappel rapide de quelques architectures**

III. **Le modèle de projet « QDMH »**
   - Présentation
   - Démonstration
   - Discussion sur Avantages - Améliorations - Limitations

IV. **Discussion sur d'autres modèles?**
   - Les vôtres, JKI, AE, Delacor DQMH, Autres,

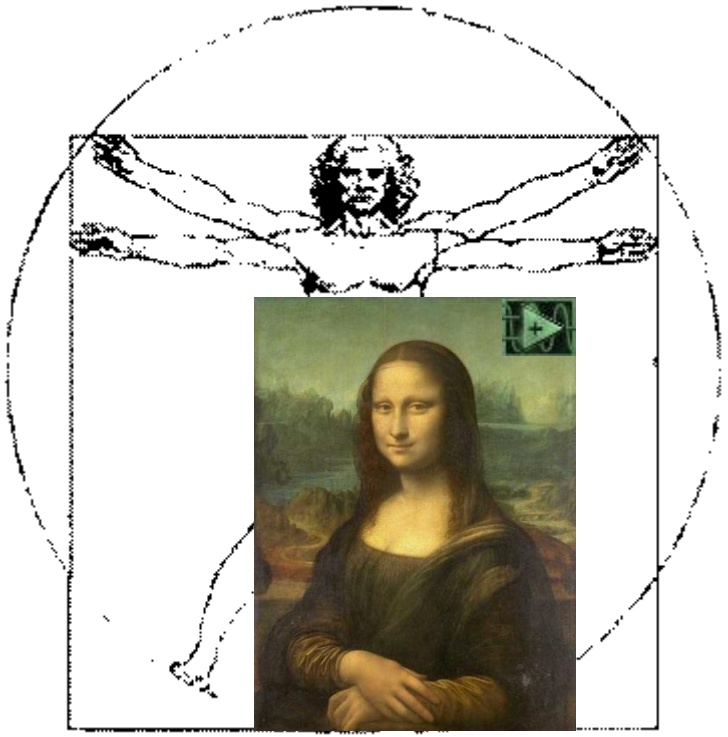
V. **Pour aller plus loin : distribuer SON modèle**

# Pourquoi un modèle?

- Compléter la phrase suivante : **sans modèle, ou avec le mauvais modèle, je ...**
  - Perds du temps
  - Réinvente la roue
  - Ne capitalise pas mon expérience
  - Perds mon savoir-faire
  - Empêche les autres de comprendre mon code
  - Crée des bugs
  - Me trompe
  - Ne documente pas l'architecture

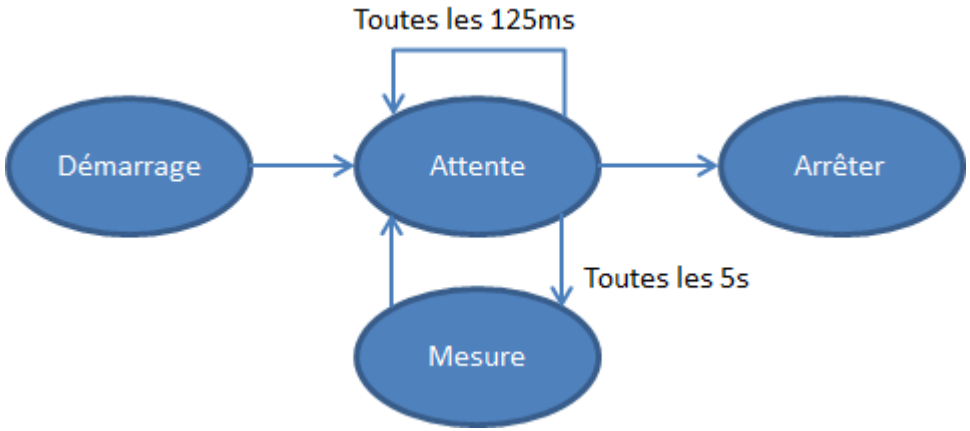
# Pourquoi un modèle?

- Qu'importe la technique il faut utiliser un modèle!!!
- Le pire est de ne pas en avoir (?)
- Mais de préférence le bon

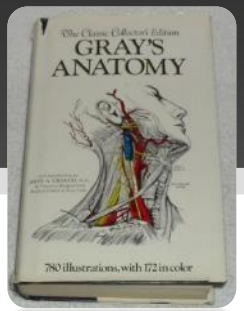


# Quelques Rappels rapides

- Une grande partie des programmes sont bien représentés par le modèle « **Machine à états** », basé sur diagramme états-transitions:
  - Etats = actions à réaliser
  - Chaque section de code détermine la transition suivante
  - Découpage clair des tâches à effectuer
  - Approche reconnue



# Machine à états en LabVIEW



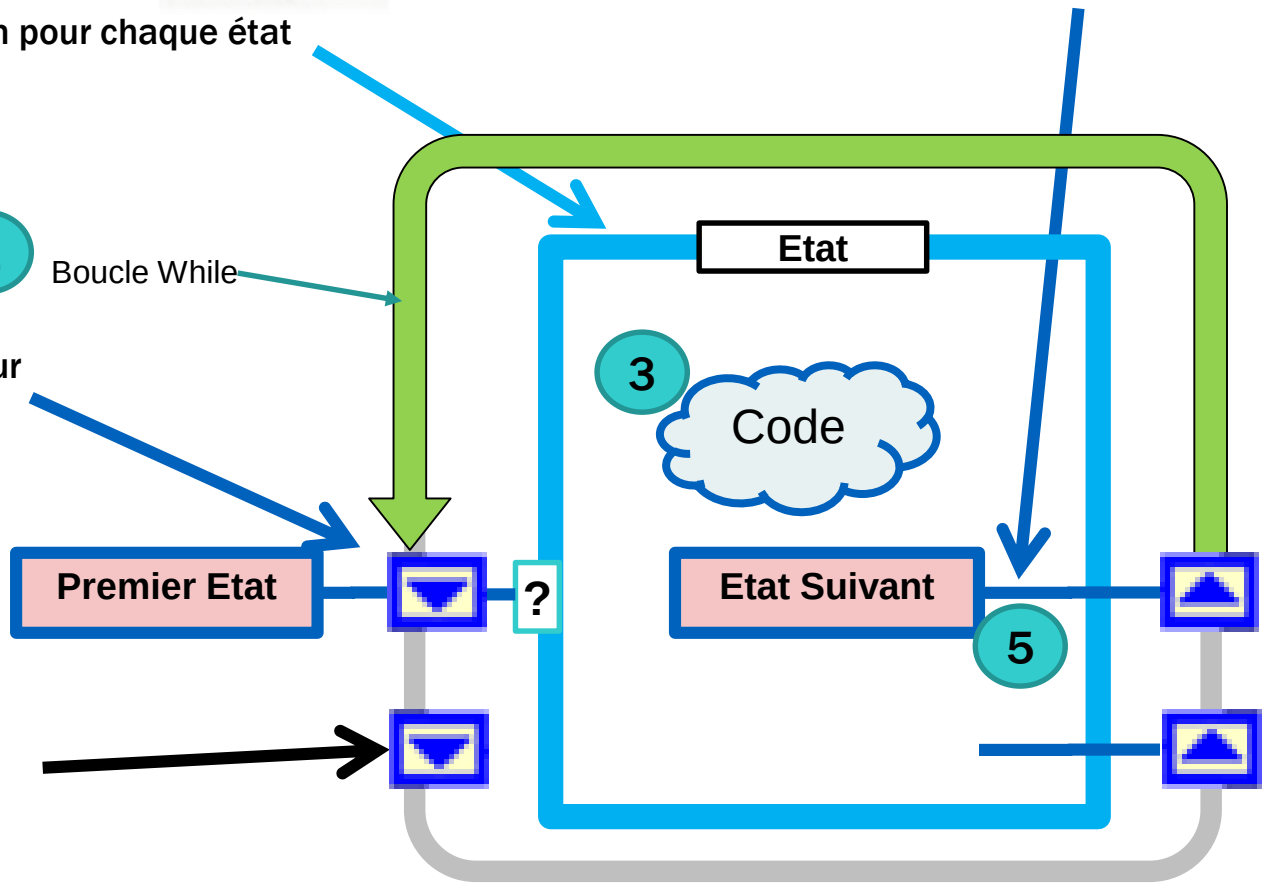

Etat suivant : transition

2 Séquence = un pour chaque état

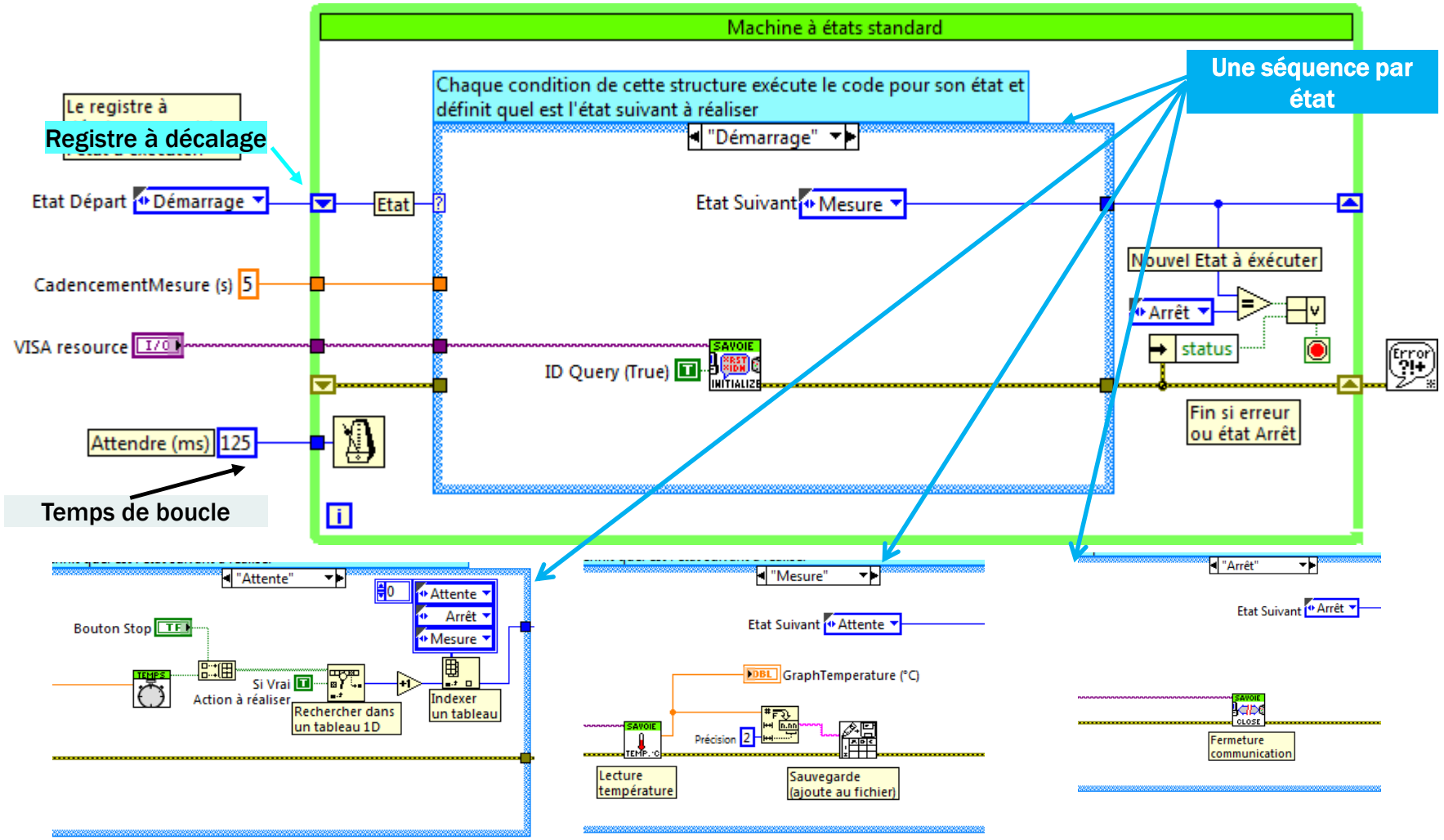
1 Boucle While

4 Registre à décalage pour les transitions d'états

6 Registre à décalage pour la mémorisation des données

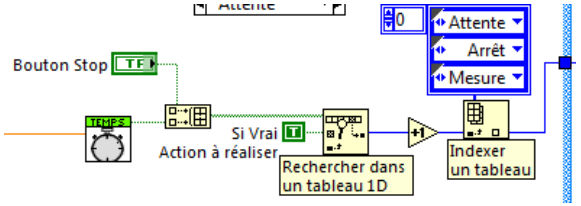


# Machine à états : Exemple



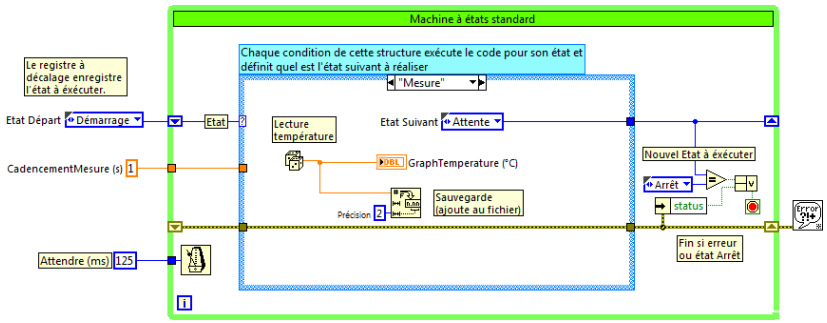
# Synthèse de la Machine à états

- **Avantages**
  - Technique Fondamentale du développement sous LabVIEW
  - Code simple
  - Documenté
  - Intuitif
- **Point faible : *Question CLAD?***
  - Peut perdre des états si détectés en même temps
- **Idées d'évolutions ?**

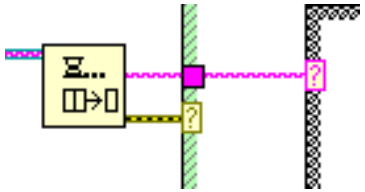


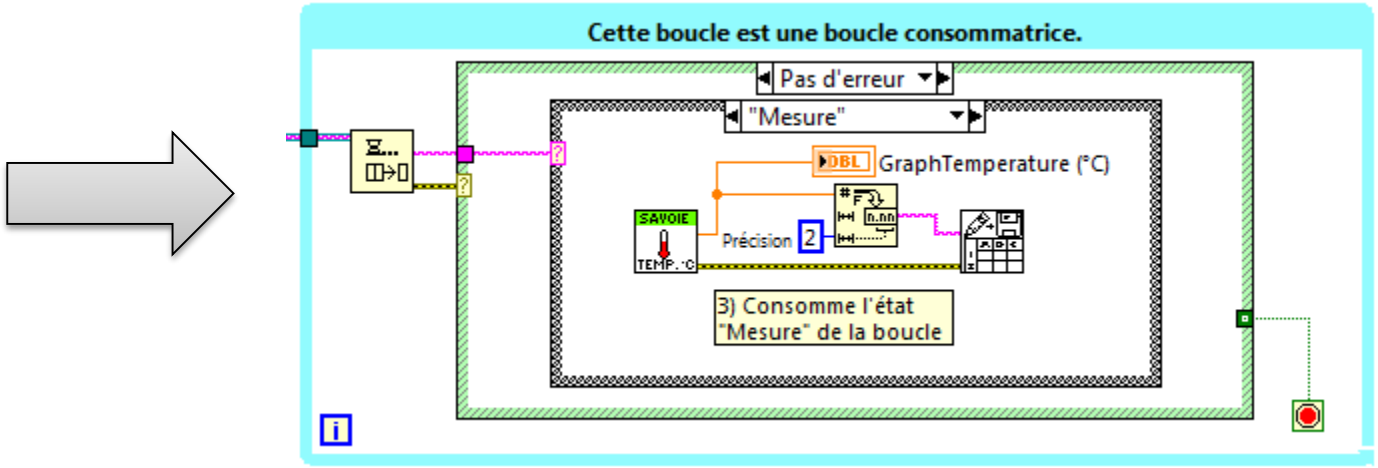

# Evolution? Queue State machine

## Machine à états




## FIFO Queue






## Queue State Machine

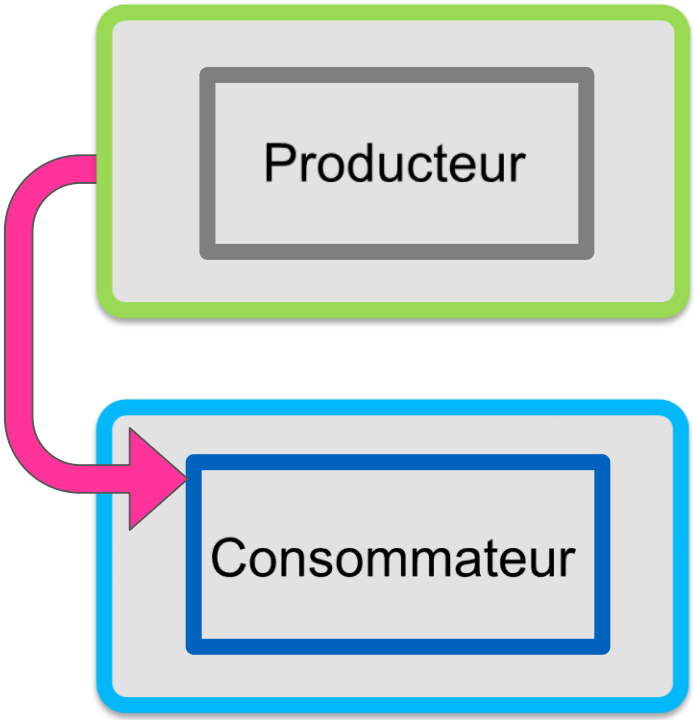
# Synthèse de la Queue State Machine

- **Avantages**
  - Capacité de mémoriser des états
  - Plus évolutifs
- **Point faible :**
  - Pas multitâches – multiprocessus
  - Si code long : bloque le programme
- **Idées d'évolutions ?**


# Producteur - Consommateur

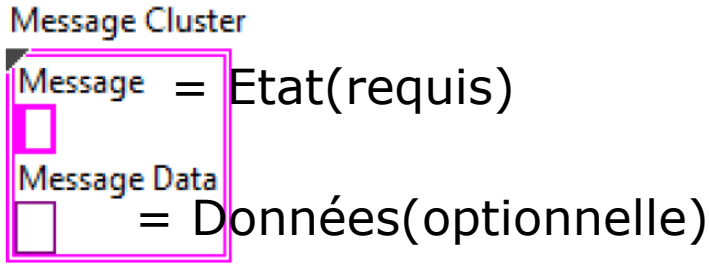
- Deux catégories de processus :
  - **Producteurs** de données
  - **Consommateurs** de données

- Communication entre processus :
  - **Message :** File d'attente « queue »

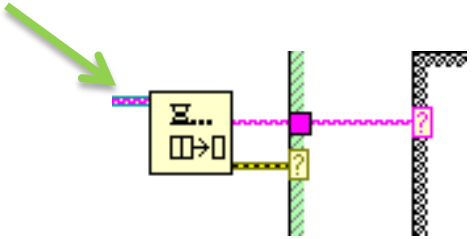


# Exemple LV : QDMH ou QMH

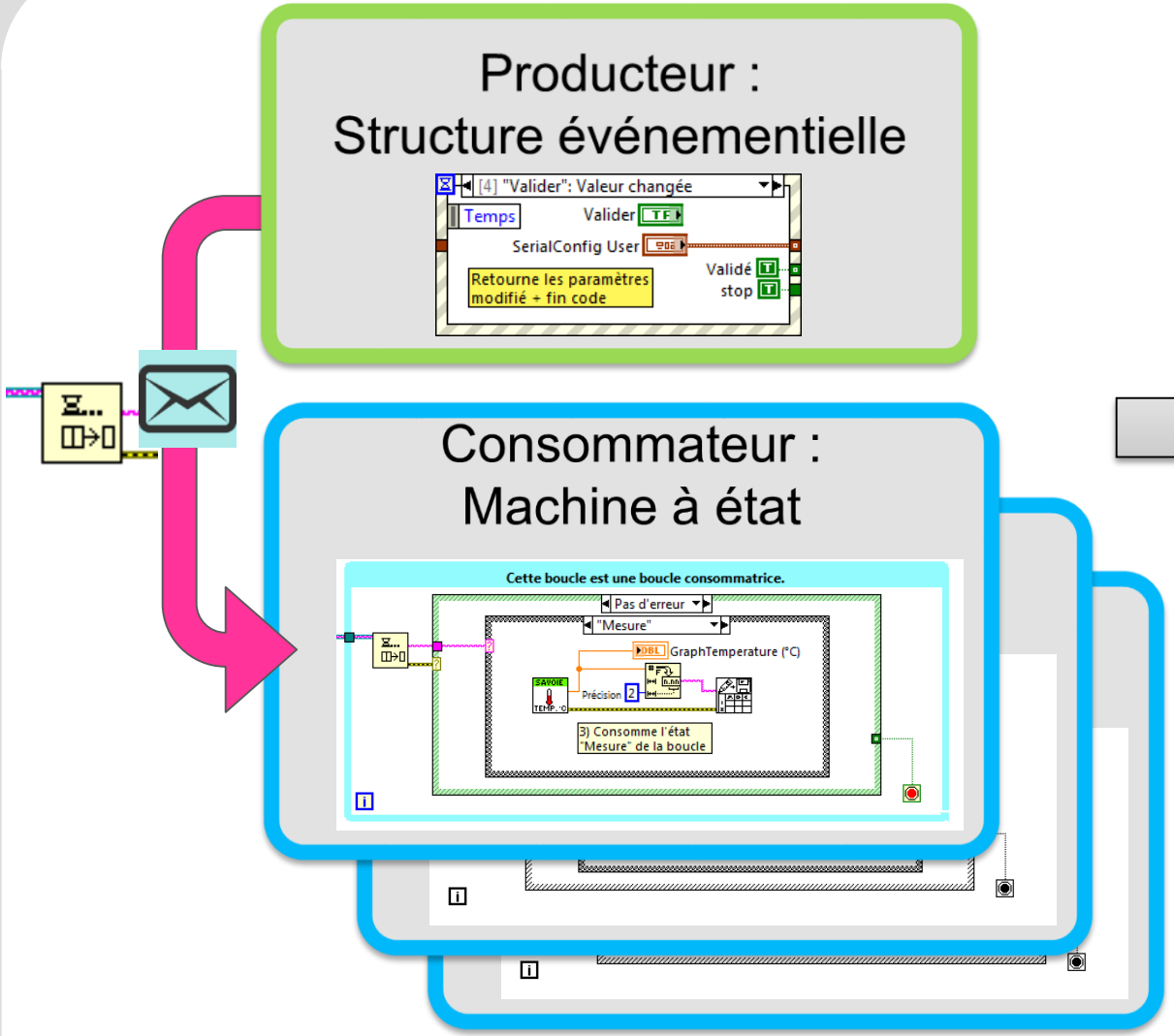
- **QDMH: Queue Driven Message Handler**
  - Modèle Producteur/Consommateur
  - Producteur : Structure événementielle (Driven)
  - Consommateur : Machine à état




- **Vue d'ensemble**
  - Plusieurs processus (task)
  - Exécutés en parallèles
  - Qui échangent des données (Message) entre eux (Queue).



# Queue Message Handler




QMH: Queue Message Handler

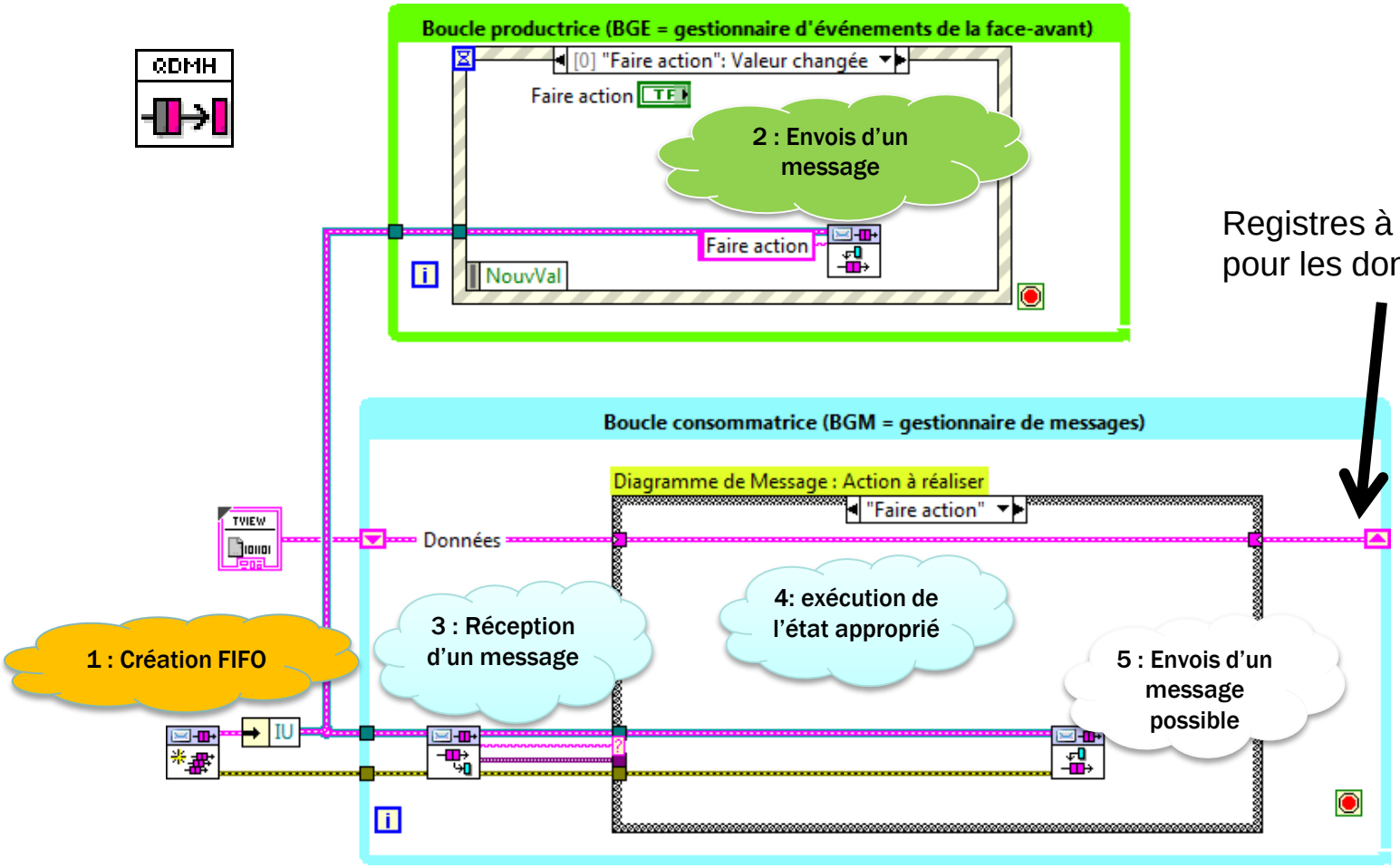
# Le modèle de projet QDMH

- La structure repose sur un modèle producteur – consommateur
  - ***la boucle productrice :*** la structure évènementielle
    - capture les actions utilisateurs, sur la face-avant, et produit le « message » via une FIFO
    - Le message est un cluster composé d'un état « case » et une donnée facultative Data de type variant
  - **la boucle consommatrice :** basée sur un modèle de machine à états, dépile sur apparition les données de la FIFO.
  - Un **message** peut être envoyé par une action utilisateur ou un autre état de la machine à états.

*Source : white-paper QMH*

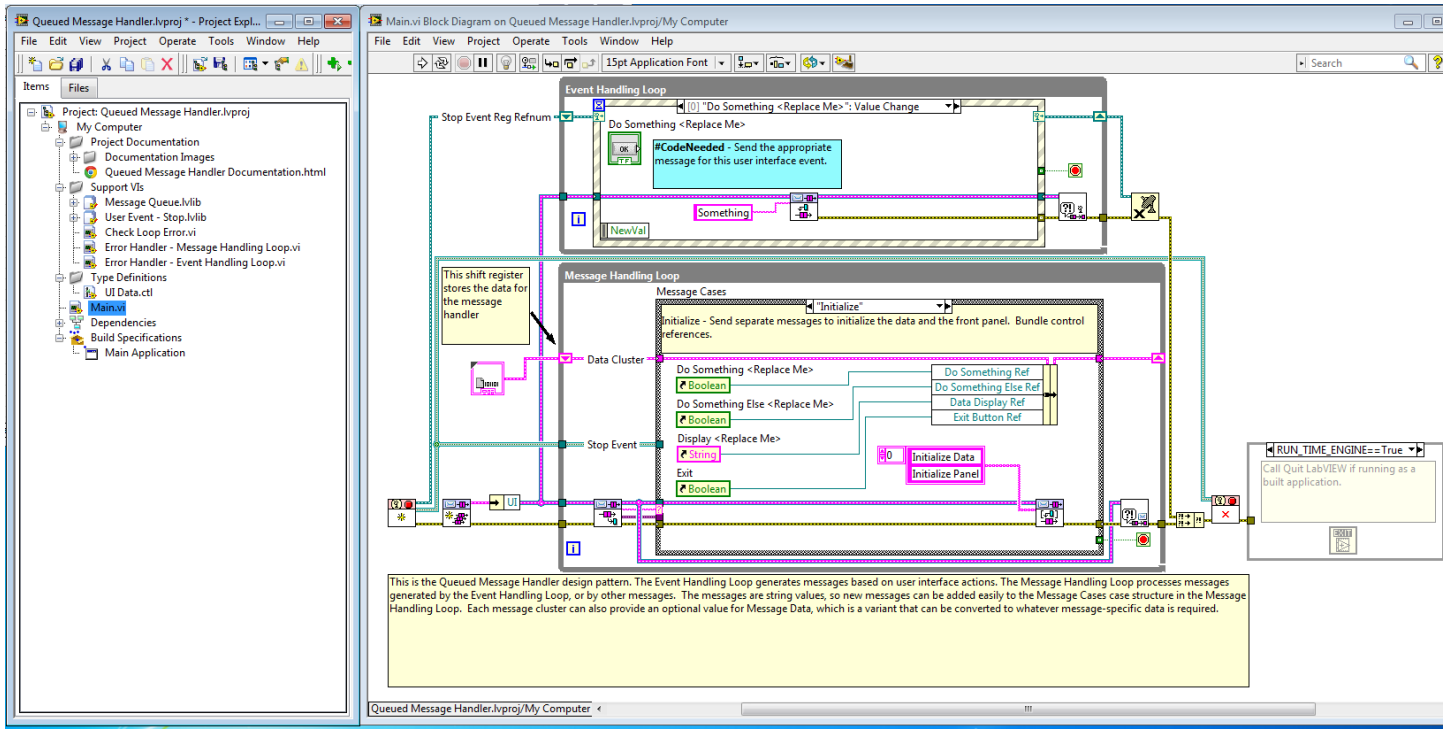
# Le modèle de projet QDMH en 5 étapes

- Démonstration



# Le modèle de projet QDMH

- LabVIEW 2012 : les modèles de projet LabVIEW via le gestionnaire de projet.



# Avantages QDMH

- A votre Avis?

# Avantages QDMH

- **UN modèle**
  - Couvre un grand nombre d'applications en restant abordable techniquement
  - Exemple par NI code **Producteur - Consommateur**
  - Un code standard, évolutif et maintenable par d'autres développeurs
  - Multitâches – multiprocess
  - Gestion de l'erreur
  - Gestion de l'arrêt des boucles – du programme
  - Projet LabVIEW lvproj
  - Standardise l'arborescence disque
  - Code documenté
  - Architecture documentée
  - La route du CLD, du CLA

# Améliorer le QDMH

- A votre Avis?

# Améliorer le QDMH : le personnaliser

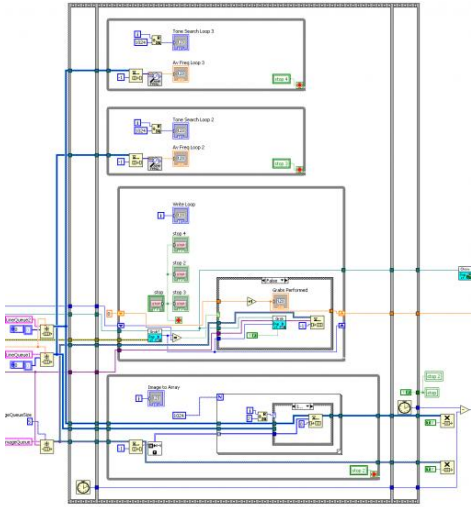
- Supprimer le code « exemple »
- Améliorer la gestion d'erreur
  - Affichage de l'erreur
  - Sauvegarde
  - Filtrage si l'utilisateur le décide.
- Ajouter une gestion de version du logiciel
  - Ajouter les VIs de gestion de version de LabVIEW
- Ajouter votre « About… », votre logo
- …

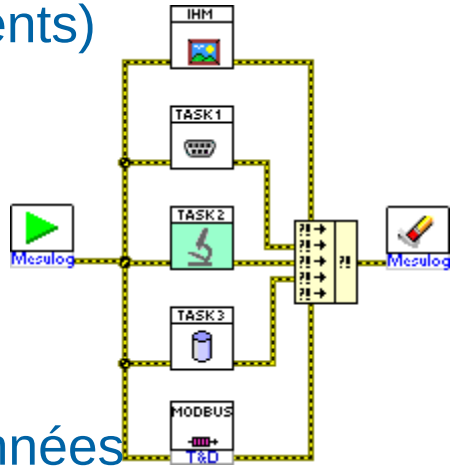
# Votre AVIS : Faiblesses? :

- **A votre Avis? Question aux Lugistes – réponses des Lugistes**
  - Nous n'utilisons pas les modèles
  - Les modèles de projets proposés ne me conviennent pas, trop simplistes, donc je ne les utilise pas.
  - Modèles de NI sont peu attirants : beaucoup de fils, difficile de s'y retrouver.
  - Quand j'en utilise un, je passe trop de temps à le rendre lisible et clair, donc j'y ai renoncé.

# Faiblesses?

- **Tous les process sont dans le même VI**
  - Difficilement lisible si plus de 3 boucles
- **Tous les process sont dans le même thread**
  - Thread UI
- **Evolutions**
  - Séparer les process (dans des VIs/thread différents)
  - Le VI HMI n'est pas le Top Level
  - Loader
  - …
- **Mise en garde**
  - Si QMH ne suffit pas = attention gestion des données
  - Accès concurrent, copie mémoire, privatisation, partage



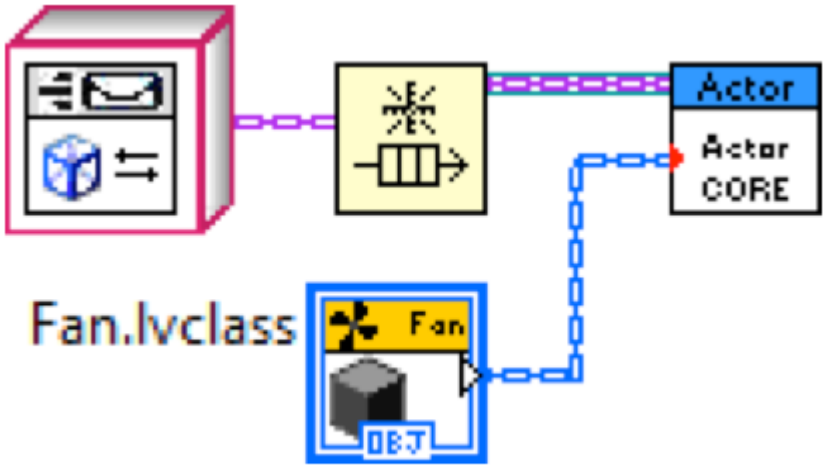


# Autres modèles

- Les vôtres?
  - Merci d'avoir partagé votre code

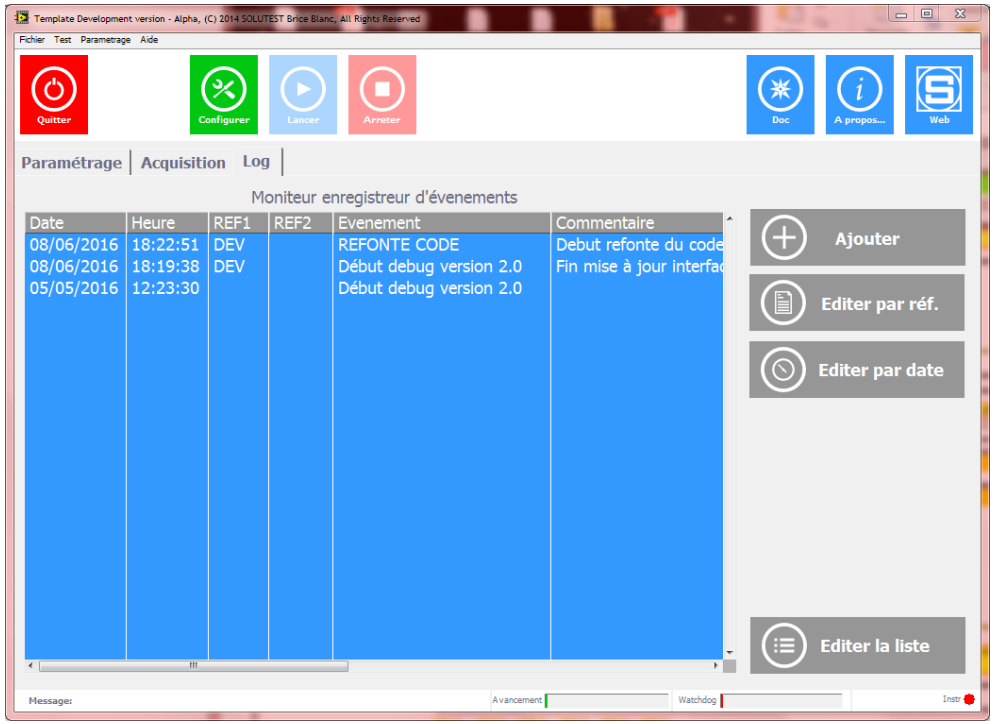
# Autres modèles

- Actor Framework
  - LUGE 1.0 par OJO
  - Ne revient pas dessus



# Autres modèles : anonyme N1

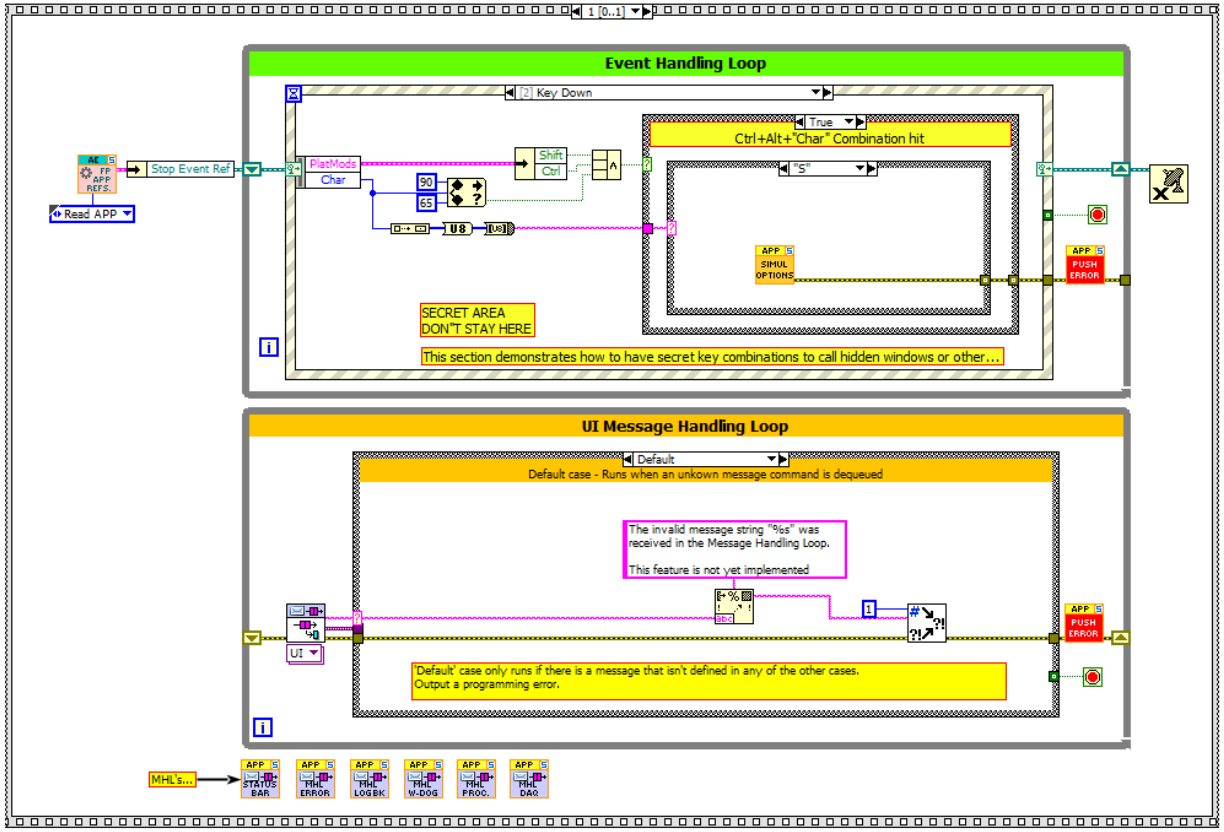
L'auteur désire rester anonyme




- Menu
- Boutons
- Id utilisateur
- Barre de message
- Progress bar
- Indicateurs instruments
- Gestion logbook

SOLUTEST
LA SOLUTION TEST & MESURE

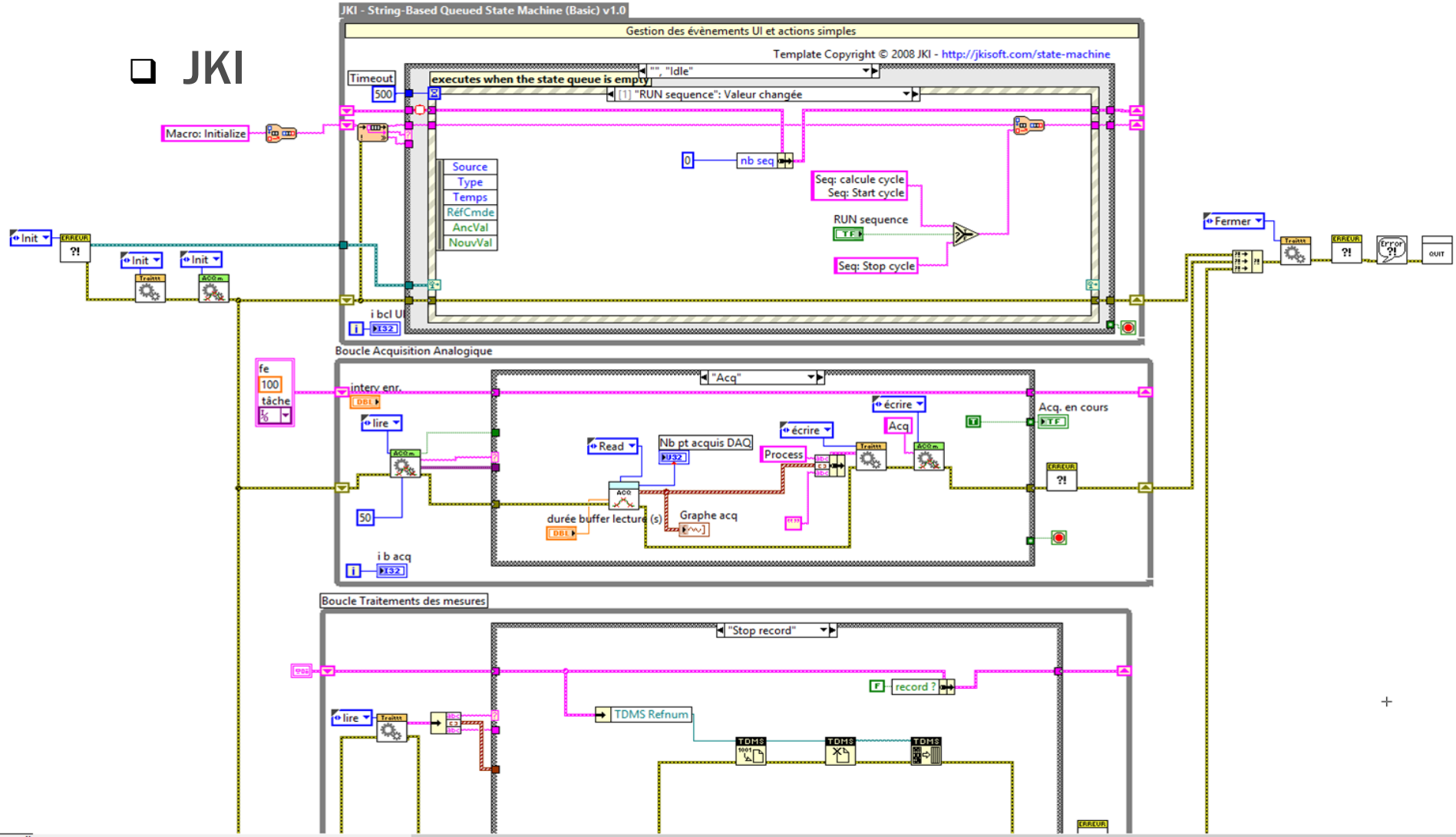
# Anonyme N°1 : Handling loop




- Event
- User Interface
- Status Bar
- Error
- Log Book
- Watchdog
- Process
- DAQ

# Autres modèles : Anonyme N°2 : JKI

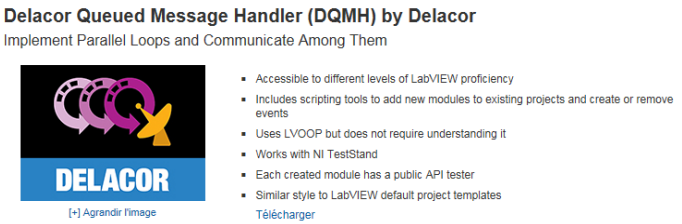
- JKI



# Autres modèles?

- D'autres modèles dans LabVIEW
  - Actor Framework

- LabVIEW Tools Network
  - JKI,
  - Delacor DQMH
    - LabVIEW Tools Network Product of the Year – 2016



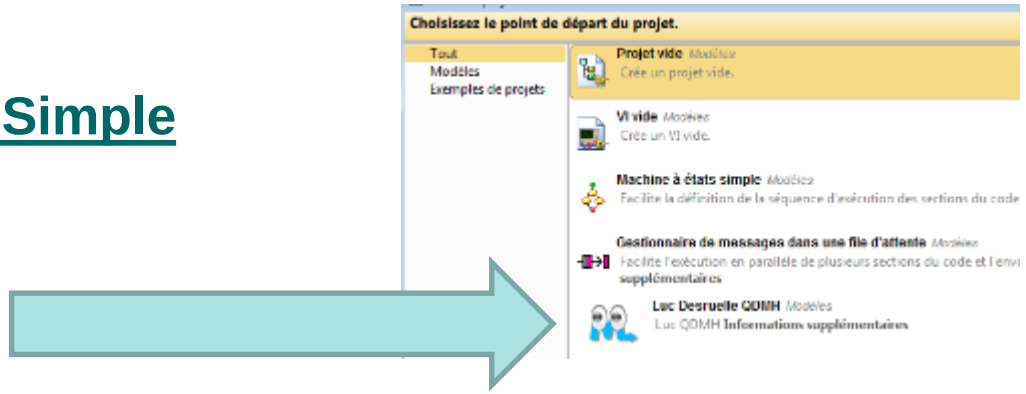

- Les vôtres?

# Conclusions : pourquoi un modèle?

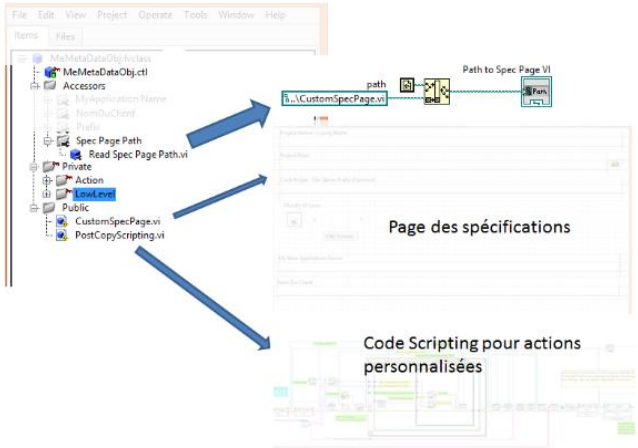
- **UN modèle, pour permettre**
  - Une lecture plus rapide du code.
  - Coder plus rapidement, projet
  - Un code standard, évolutive et maintenable par d'autres développeurs.
  - Code & Architecture documentés
  - La route du CLD - CLA
- **QDMH : un modèle puissant pour beaucoup d'applications**
  - Personnalisation simple
  - Evolutions simples
- **VOTRE modèle, que vous réutilisez d'un projet à un autre,**
  - Personnalisation de la structure (version, gestion erreur, about, version validée ou pas…)
  - Standardisation de l'arborescence disque

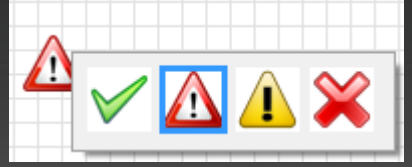
# Pour aller plus loin : Distribuer Votre Modèle

- Distribuer votre modèle avec le gestionnaire de projet LabVIEW
  - **Partie 2/3 : Version Simple**




  - **Partie 3/3 : Version distribution personnalisée**





# Liens

## Plus de présentations techniques

- www.mesulog.fr/presentations_techniques
- Mon blogue LabVIEW : Blogue sur le site National Instruments
- Tutoriels developpez.com
- Luc Desruelle | LinkedIn

## Plus de livres

"LabVIEW programmation et applications" 3iéme édition, Dunod

Exemples et images extraits du chapitre 3

## Plus de National Instruments Francophone

- Forum francophone NI LabVIEW
- Forum francophone Autres produits NI
- Communauté Francophone

# Où trouver de l'aide ?

- La communauté travaille pour vous
  - 2 Forums National Instruments Francophone
    - http://forums.ni.com/t5/Discussions-au-sujet-de-NI/bd-p/4171
    - http://forums.ni.com/t5/Discussions-au-sujet-des-autres/bd-p/4170
  - 1 communauté Francophone
    - https://decibel.ni.com/content/community/regional/france/labview
    - Rencontre développeurs : cf LUGE
  - Mais aussi… l'ensemble des forums et communautés de NI
    - **LabVIEW Development Best Practices**
    - Large Applications : ni.com/largeapps

# Questions?