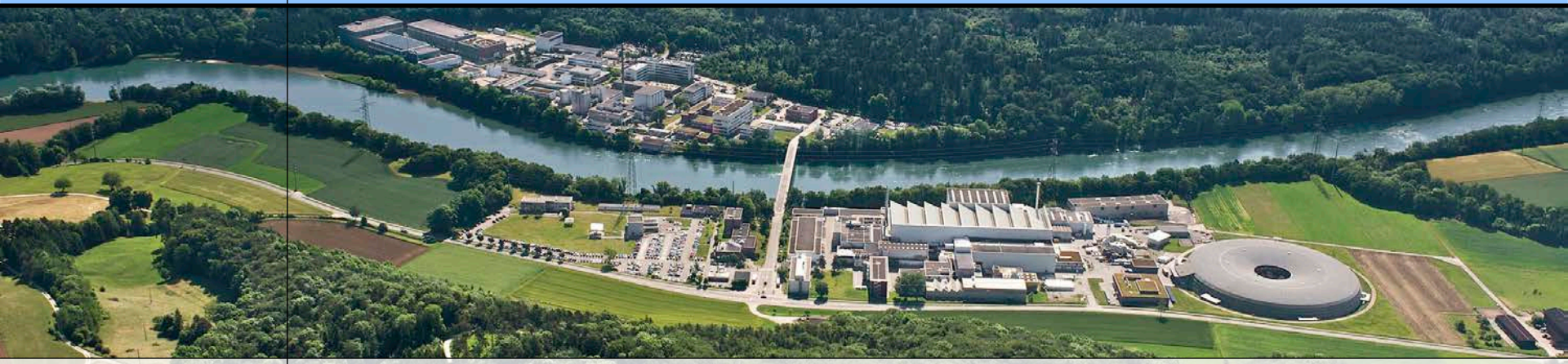
PAUL SCHERRER INSTITUT

PSI



Wir schaffen Wissen – heute für morgen

An EPICS Display manager with Qt

A.C.Mezger

Outline of the presentation

- Qt: a new system for writing graphic user interfaces.
- Qt: Architecture of MEDM and caQtdm
- Qt-Designer: what can it do.
- Translator: parsing of .adl files.
- Basic principles.
- Some hints.
- Conclusion.
- Demo.

Qt: a new system for writing graphic user interfaces

What is Qt (***definition given by Wiki***):

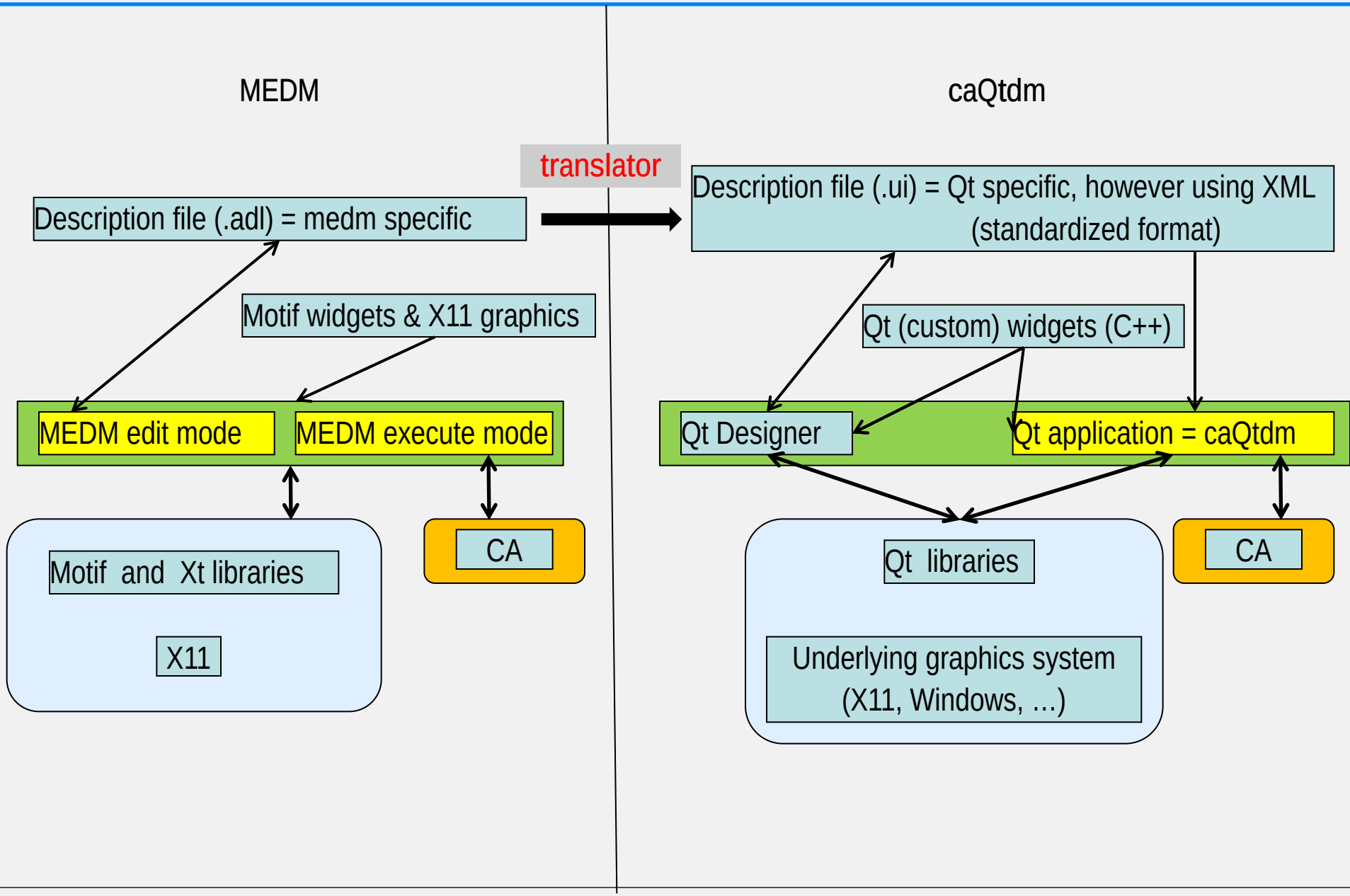
Qt is a cross-platform application framework that is widely used for developing application software with a graphical user interface (GUI) (in which cases Qt is classified as a *widget toolkit*

Qt runs on the major desktop platforms and some of the mobile platforms.

Qt comes with libraries, an integrated development environment (IDE) called QtCreator with an user interface designer (Designer) and some third party libraries like qwt for 2D graphics and qwtplot3d for 3D graphics.

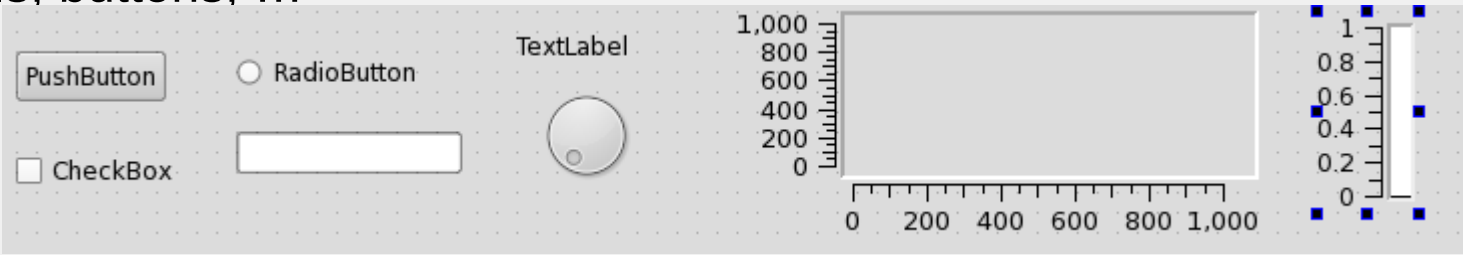
The actual version used is Qt 4.8.2

Architecture of MEDM and caQtdm

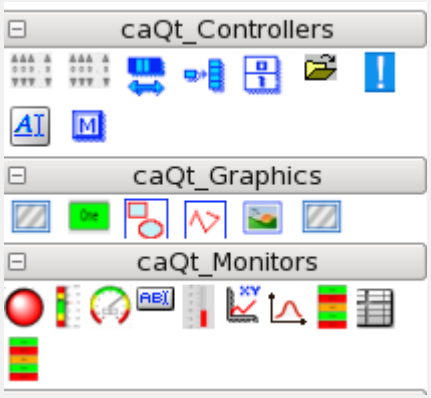


Qt-Designer, what can it do

- a tool for building graphic user interfaces with widgets like sliders, knobs, labels, buttons, …



- **and you can introduce your own widgets:** stripchart, cartesian plot, composite file, related display, …



- **while widgets are used, you can only use widgets to draw objects.**
  - Drawing widget for rectangles, circles, ovals, lines, triangles can be created.
  - Polylines, Polygones widget has an integrated editor for drawing with the mouse.

The .adl files: translation

Translator (adl2ui) is based on parser code of MEDM and generates ui files. Two modes exist:

1. Parse all composite files from a main .adl file and produce one flat file:
    - Normally not very useful

2. Parse the .adl file and produces “includes”, you have also to parse the underlying ui files:
    - The designer will show all the includes, but you have to be in the directory where these are situated.

Problems:

- The widget ordering (in front or behind) is not always correct due to the widget principle.
- In Qt, the width of a text widget has to fit the text, but in MEDM the width was not always adjusted

Basic principles

1. Stylesheets: Qt uses stylesheets at several levels and inheritance of styles (CSS), One can define styles at application level, at file level (mainwindow) and widget level

   Example: color for window Qwidget#centralWidget {background: rgba(187, 187, 187, 255);}

2. Building a DM synoptic:

   - Translate an existing medm file. A default stylesheet will be automatically included. The DM uses the same stylesheet. Only few definitions are implemented (window color, table color and font, default caLineEdit and caTextEntry colors, QPushButton colors)
   - Copy an existing file and modify it with the designer
   - Start from scratch with the designer:

   ❑ Copy and paste the contents of the default stylesheet to the mainwindow stylesheet, change the color definition inside.

   ❑ Drop the objects to your window and customize them (colors, channels,)

3. Load the file into the DM (the reload function is convenient during edition)

Tips and Tricks

1. Use transparent background for files that will be included:
   Qwidget#centralWidget {background: rgba(187, 187, 187, 0);}
2. Follow the style guide of Andreas Luedeke: (in particular about where the author of the display is mentioned)
3. Use the reload function when developing screens with the designer
4. Be careful with in front and behind widgets. When drawing a rectangle in front of a widget that should be active. It will not be active.
5. You may use also other widgets than just the control widgets: split windows, tabbed windows, dock windows.
6. Start designer with the command qtdesigner, Start caQtDM with the command startDM. The appropriate architecture will be used (32, 64 bits, SL5 or SL6)
7. If you use designer and caQtDM remote, use –graphicssystem native on the command line, otherwise response is slow
8. Maximum display rate is 5Hz, but can be changed for individual channels by specifying {“monitor”:{“maxdisplayrate”:20}} after pv name

Conclusion

- the .adl translator produces correct .ui files. Some few files have to be edited after translation.
- caQtDM reproduces correctly the behavior of MEDM. However some «bugs» will still be found.
- More widgets are easily integrated. Actually table, led widgets, cameras and internal channels have been added.
- The Qt designer presents a good editor tool.
- Resizing windows is possible, but needs a good knowledge of layouts and can be used for simple layouts.
- Demo

Acknowledgments:

The authors of MEDM for their code.
Elettra people for providing some widgets.

- Thank you for your attention