

QSkinny - A new approach for a QtQuick framework



Who am I?

- » working on Qt since 2008
- » former QtNetwork maintainer
- » @peha23 on Twitter
- » founder of edelhirsch.io





image © AGCO GmbH



Why?
QSkinnny?



Why create QSkinny?

- » performance / memory usage
- » freedom of choice between QML / C++ (including pure C++)



Why use QSkinny?

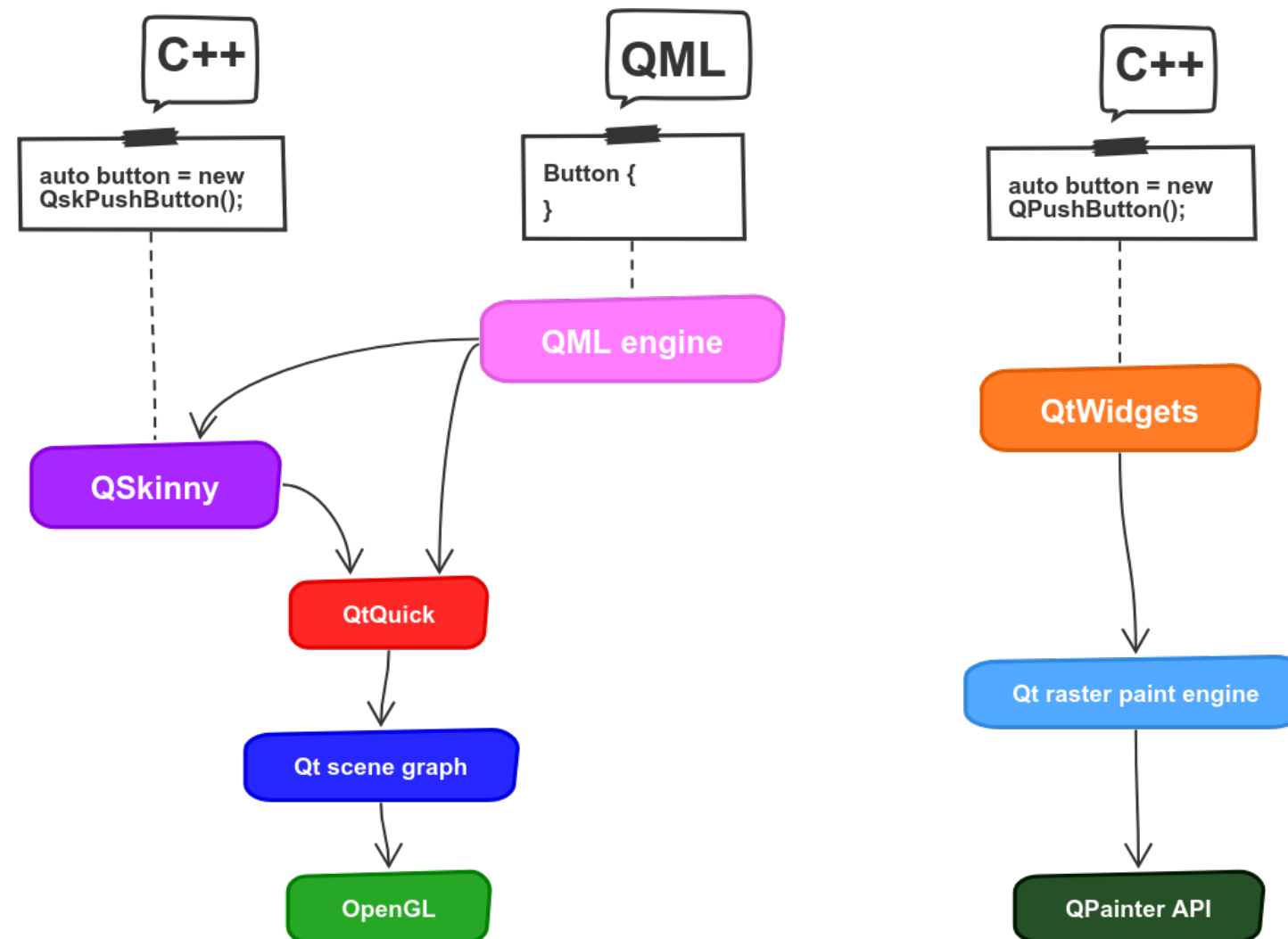
- » public C++ API
- » usable from QML
- » LGPL version 2



What
is QSkinny?



A lightweight framework on top of the Qt scene graph



API example

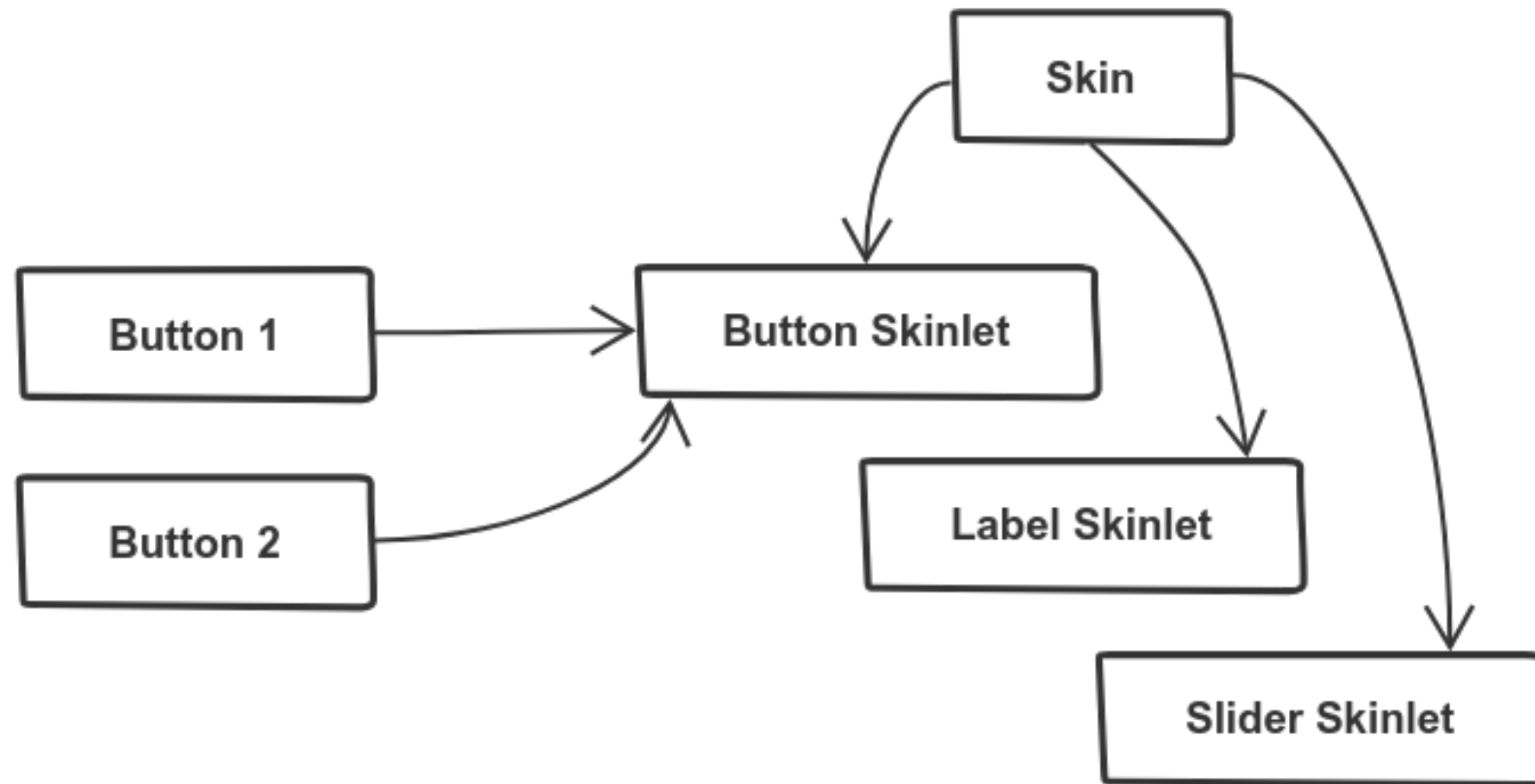
```
auto button = new QskPushButton("push me");  
auto label = new QskTextLabel("label");  
  
auto box = new QskLinearBox(Qt::Vertical);  
  
box.addItem(button);  
box.addItem(label);  
  
QskWindow window;  
window.addItem(box);  
window.show();
```



How
does QSkinny work?



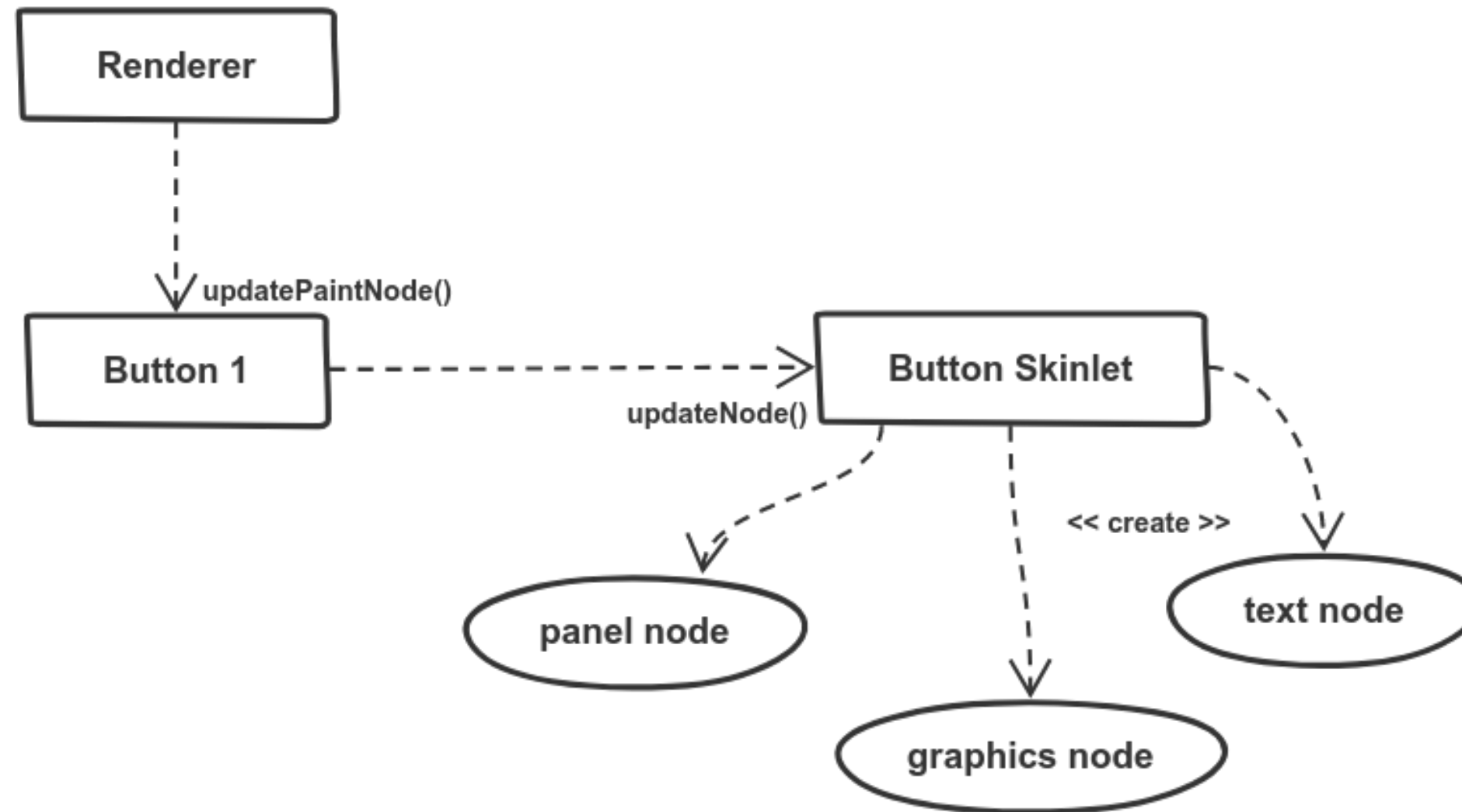
separation of implementation and styling: classes



www.sketchboard.io



separation of implementation and styling: flow



www.sketchboard.io



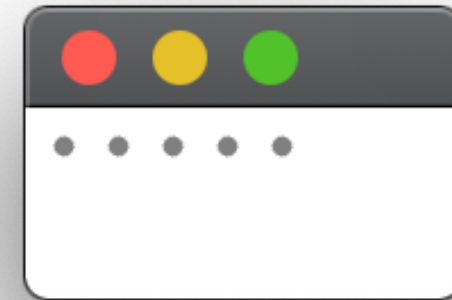
example: QSkinny page indicator

--- user code:

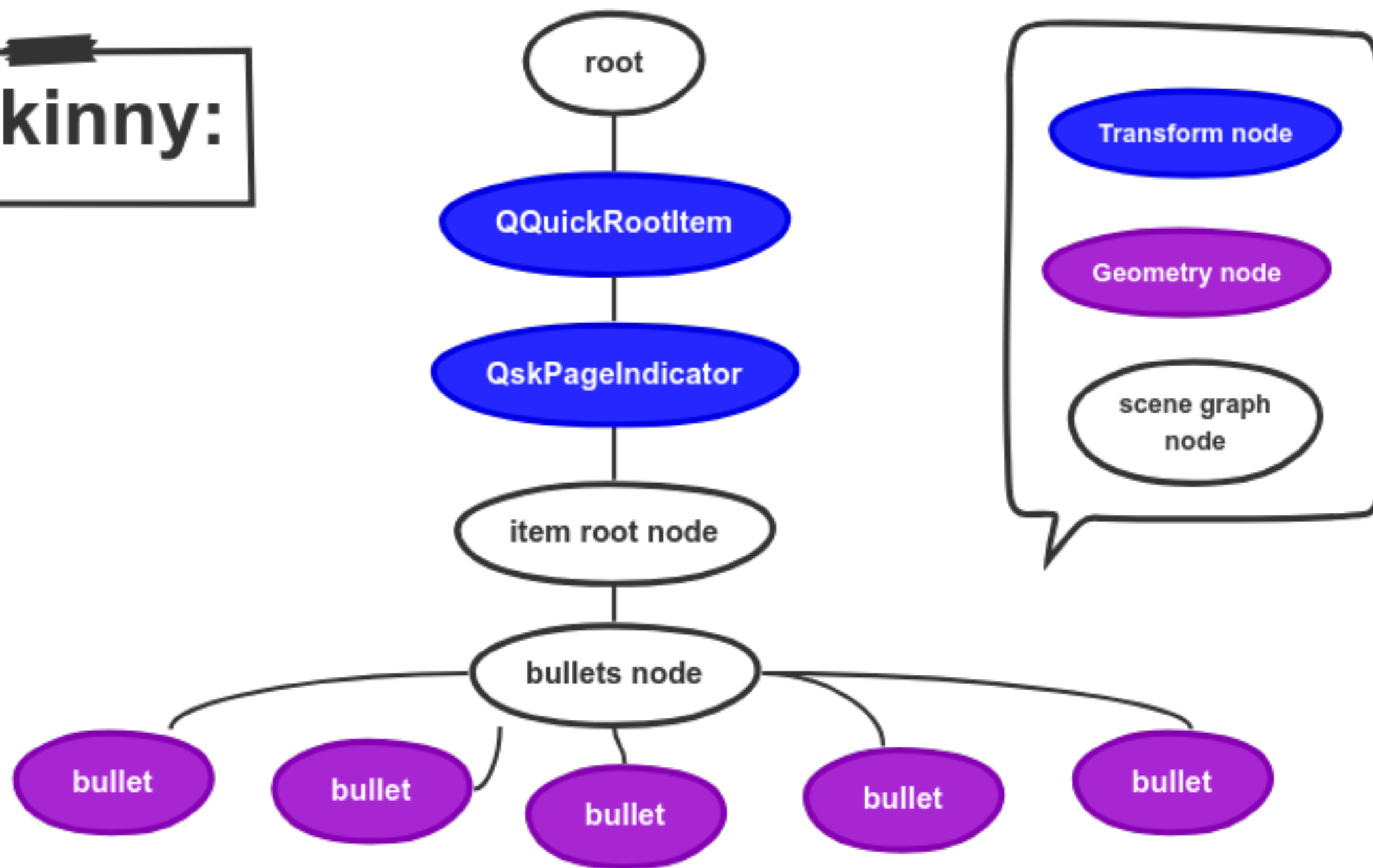
```
auto* pageIndicator = new QskPageIndicator();  
pageIndicator->setCurrentIndex( 0 );  
pageIndicator->setCount( 5 );
```

--- internal:

```
for ( int i = 0; i < indicator->count(); i++ )  
{  
    bulletNode = new QskBoxNode();  
}
```



QSkinny:



www.sketchboard.io



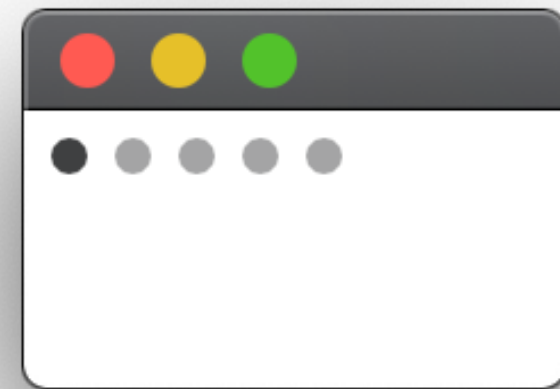
example: QML page indicator

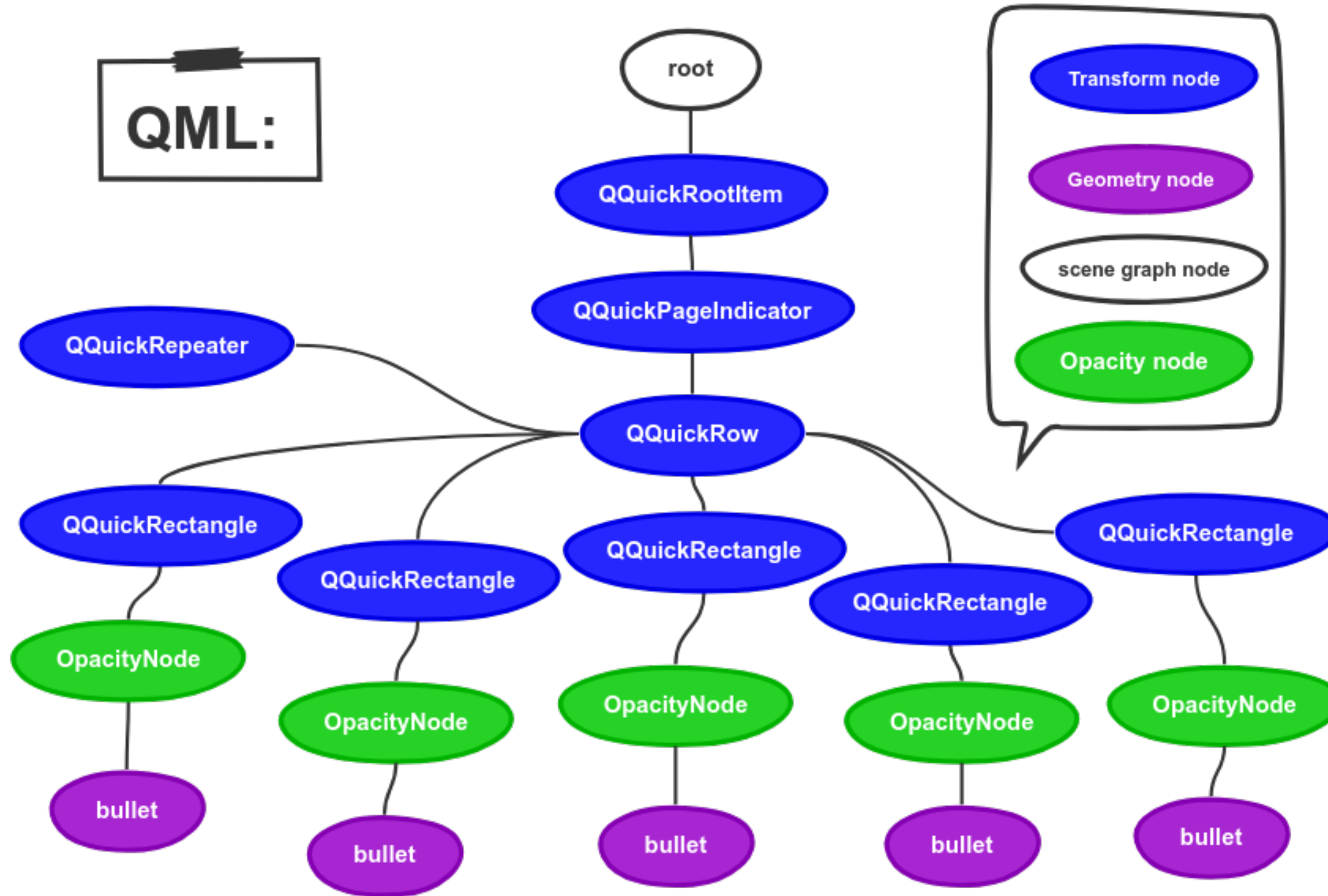
--- user code:

```
PageIndicator {  
    currentIndex: 0  
    count: 5  
}
```

--- internal:

```
T.PageIndicator {  
    delegate: Rectangle {  
    }  
}
```





www.sketchboard.io



Thanks!

<https://github.com/uwerat/qskinny>

<https://github.com/peter-ha/qtworldsummit2019-presentation>

peter@edelhirsch.io

support@qskinny.org

