

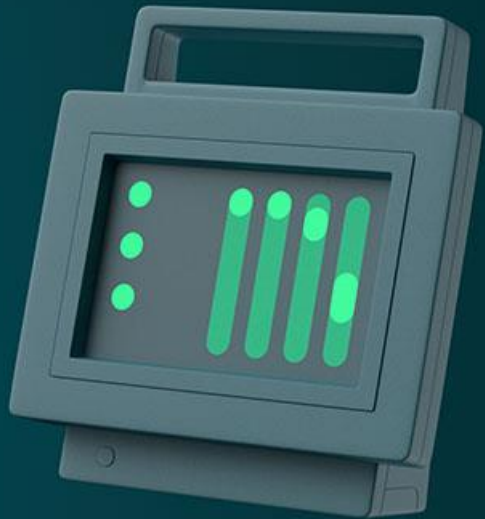
# Qt on Windows

Windows 11 Style and recent developments

5.9.2024

**Qt** Group





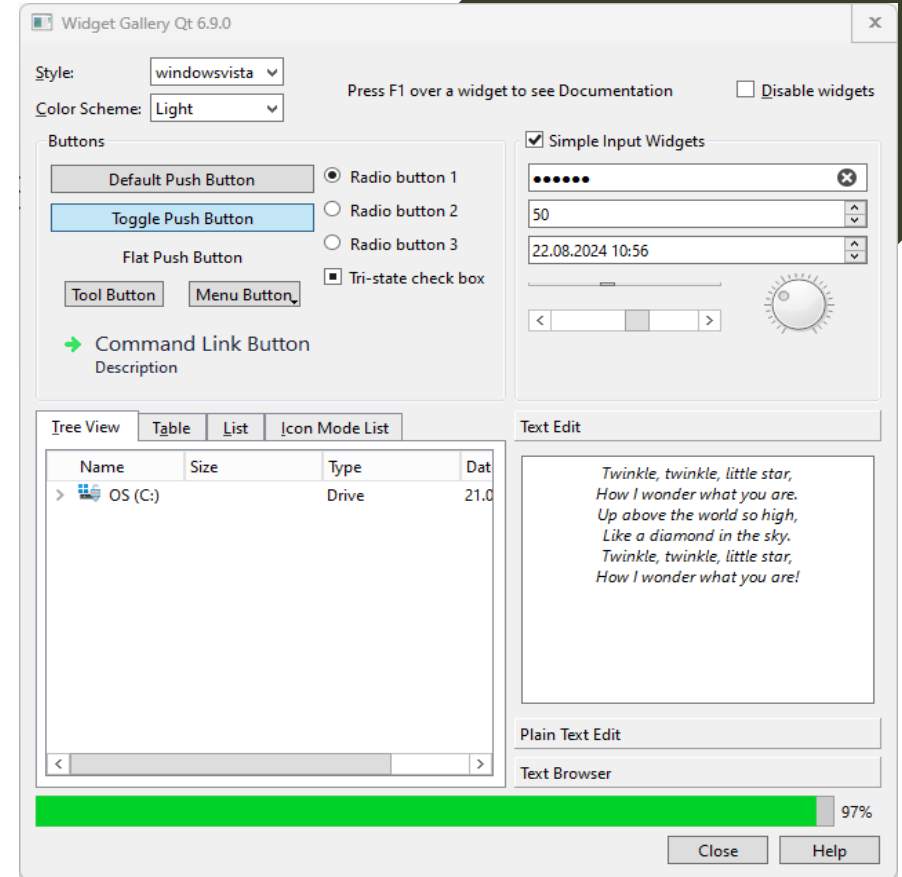
1. State before Qt 6.7
2. Windows 11 and WinUI3
3. Implementing WinUI3 Style in QtWidgets
4. Future Development

# State Before 6.7

QWindowsVistaStyle

- Pixmap based style
- Implemented through UXTheme API (deprecated)
- Ignores accent color
- No darkmode support

=> Looks outdated and not native on Windows 11 anymore



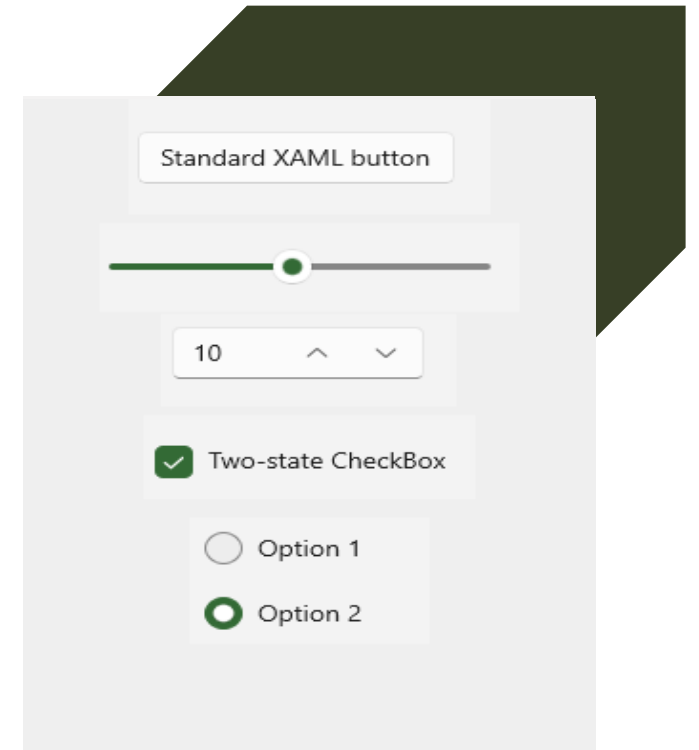
# Windows 11 and WinUI3

## WinUI3 Style

- Rounded corners
- Alpha blending used heavily
- More colors used than before
- UI elements bigger than before
- Supports light and darkmode

Detailed Style Guide as Figma Template (CC BY 4.0):

<https://www.figma.com/community/file/1159947337437047524/windows-ui-3>



# Windows 11 and WinUI3

## Native usage of WinUI3

- UI is defined through XAML
- WinUI3 focused on Universal Windows Platform (UWP)
- Win32 Apps need to use XAML Islands for Interop

```
1. <!-- Add GRID CODE -->
2.
3. <Grid>
4.     <Grid.RowDefinitions>
5.         <RowDefinition Height="Auto"/>
6.         <RowDefinition Height="*" />
7.     </Grid.RowDefinitions>
8.
9.     <Grid Margin="12">
10.        <Grid.ColumnDefinitions>
11.            <ColumnDefinition Width="*" />
12.            <ColumnDefinition Width="Auto" />
13.        </Grid.ColumnDefinitions>
14.        <!-- STEP 1: Add DropDownButton & ToggleSwitch CODE -->
15.    </Grid>
16.
17.    <Grid x:Name="Control1" Grid.Row="1" ColumnSpacing="30" RowSpacing="12"
18.        VerticalAlignment="Center" HorizontalAlignment="Center">
19.        <!-- STEP 3: Define column & row positioning -->
20.        <!-- STEP 4: Add Styling for BUTTON -->
21.        <!-- STEP 2: Add BUTTON CODE -->
22.    </Grid>
23. </Grid>
```

# Implementing WinUI3 Style in QtWidgets

## Challenges

- Qt is based on Win32 APIs
- Qt is not using the UWP platform
- XAML is Windows only

=> Qt can't use WinUI3 APIs directly

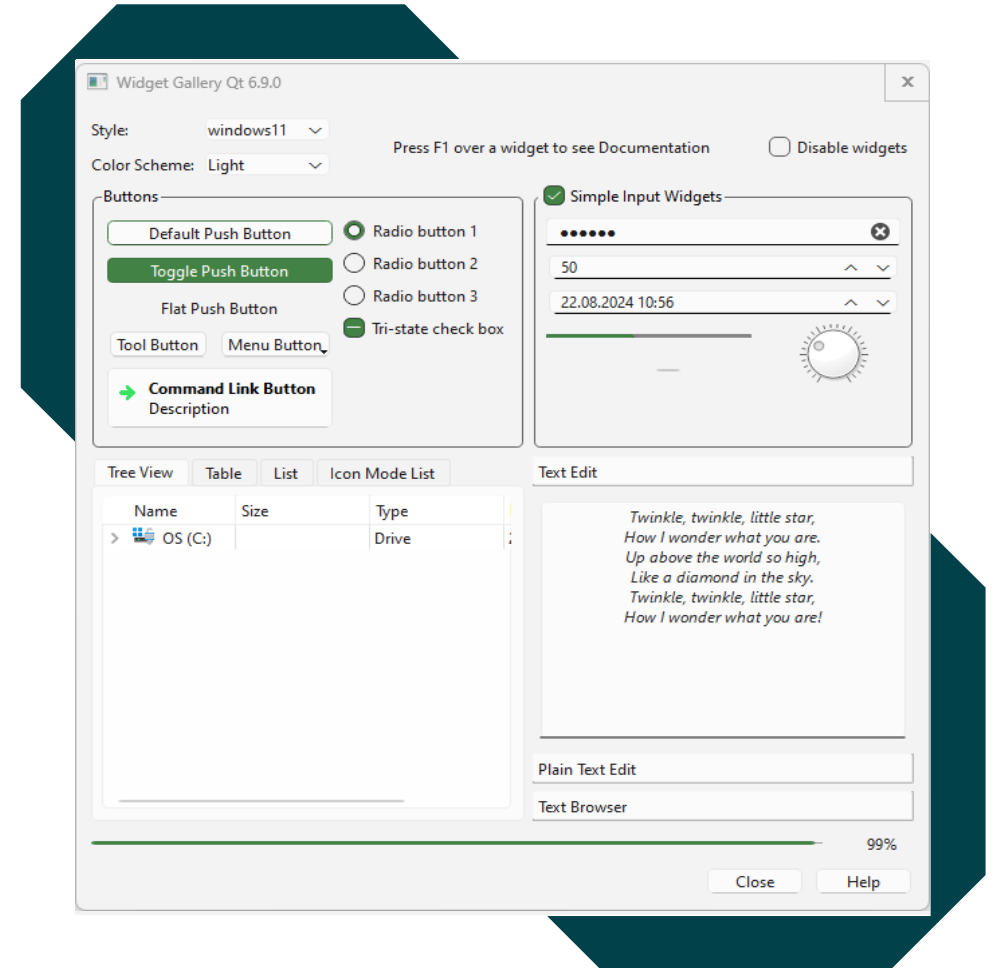
Solution:

Create a new Style based on QWindowsVistaStyle that mimicks WinUI3

# Implementing WinUI3 Style in QtWidgets

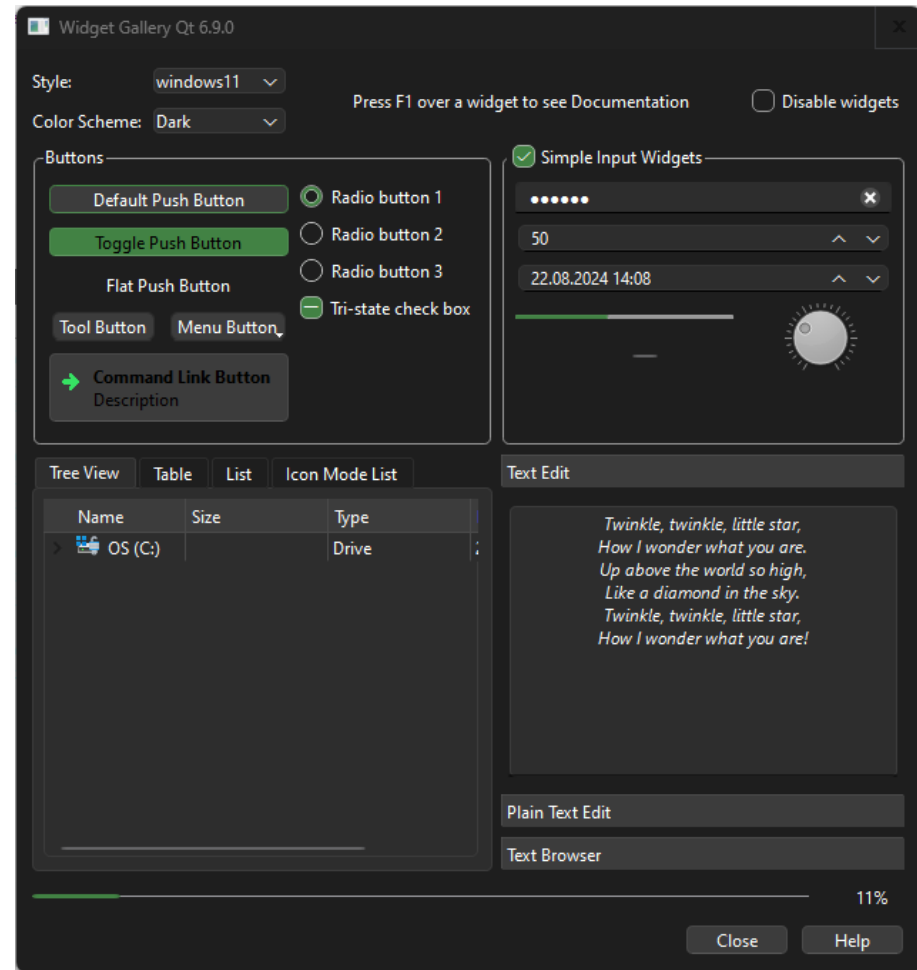
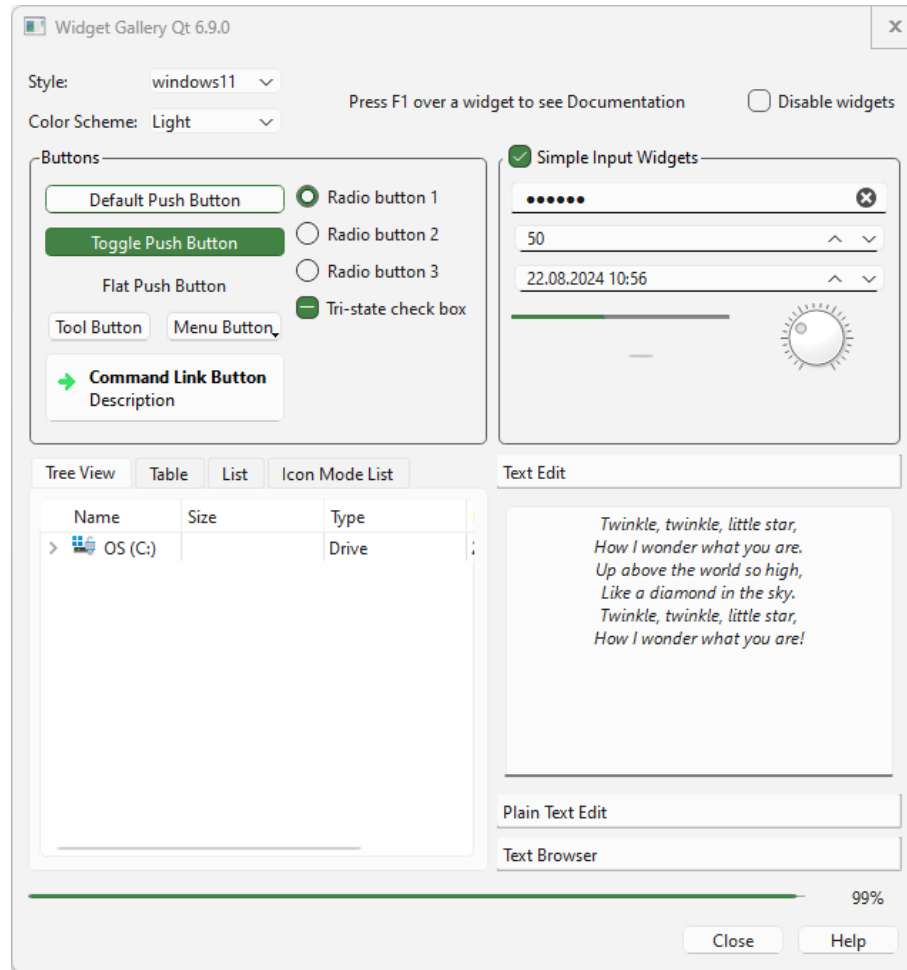
## QWindows11Style Challenges

- WinUI3 uses more colors than QPalette offers
  - Create a color lookup in QWindows11Style for dark/light theme
- WinUI3 uses rounded edges and alpha blending heavily
  - Set `Qt::WA_TranslucentBackground` on UI elements
  - Set `Qt::FrameLessWindowHint` on UI elements
- WinUI3 uses an Segoe MDL2 Assets as AssetFont
  - Lookup needed assets UTF-8 code in charmap
- Keeping compatibility with older Styles
  - Do the palette overrides in `QStyle::polish` to retain original palette



# Implementing WinUI3 Style in QtWidgets

QWindows11Style





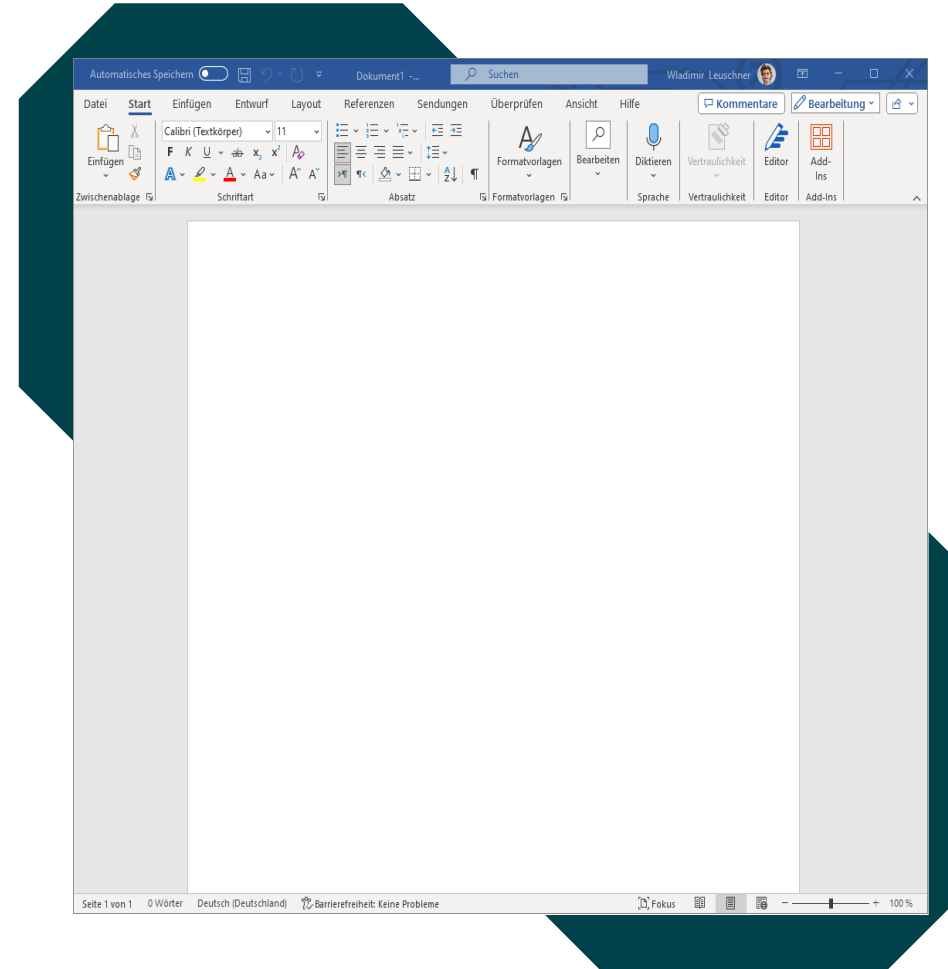
# Future Development

- Custom Titlebars
- Refactoring COM to winrt
- Windows on ARM
- Transparency effects
- Windows packaging (msix)
- .NET with Qt
- Windows App SDK (WebView2)

# Future Development

## Custom Titlebars

- Placing UI elements into the titlebar is a recent UI trend
- Creating a custom titlebar is currently a lot of manual work
- Drawing Widgets and Quick UI into titlebar area
- Prototype implementation available:
  - Widgets: <https://codereview.qt-project.org/c/qt/qtbase/+/567970>
  - Quick UI: <https://codereview.qt-project.org/c/qt/qtdeclarative/+/578518>
- Wladimir Leuschner: [wladimir.leuschner@qt.io](mailto:wladimir.leuschner@qt.io)



# Future Development

Refactoring COM to winrt

- Using newer winrt APIs to replace COM implementations
  - Increase maintainability
  - Make error handling more consistent
  - Make consistent use of best practices
- Bluetooth LE as first submodule to see refactor
  - <https://codereview.qt-project.org/c/qt/qtconnectivity/+/564449>
- Miguel Costa: miguel.costa@qt.io
- Oliver Wolf: oliver.wolff@qt.io

# Future Development

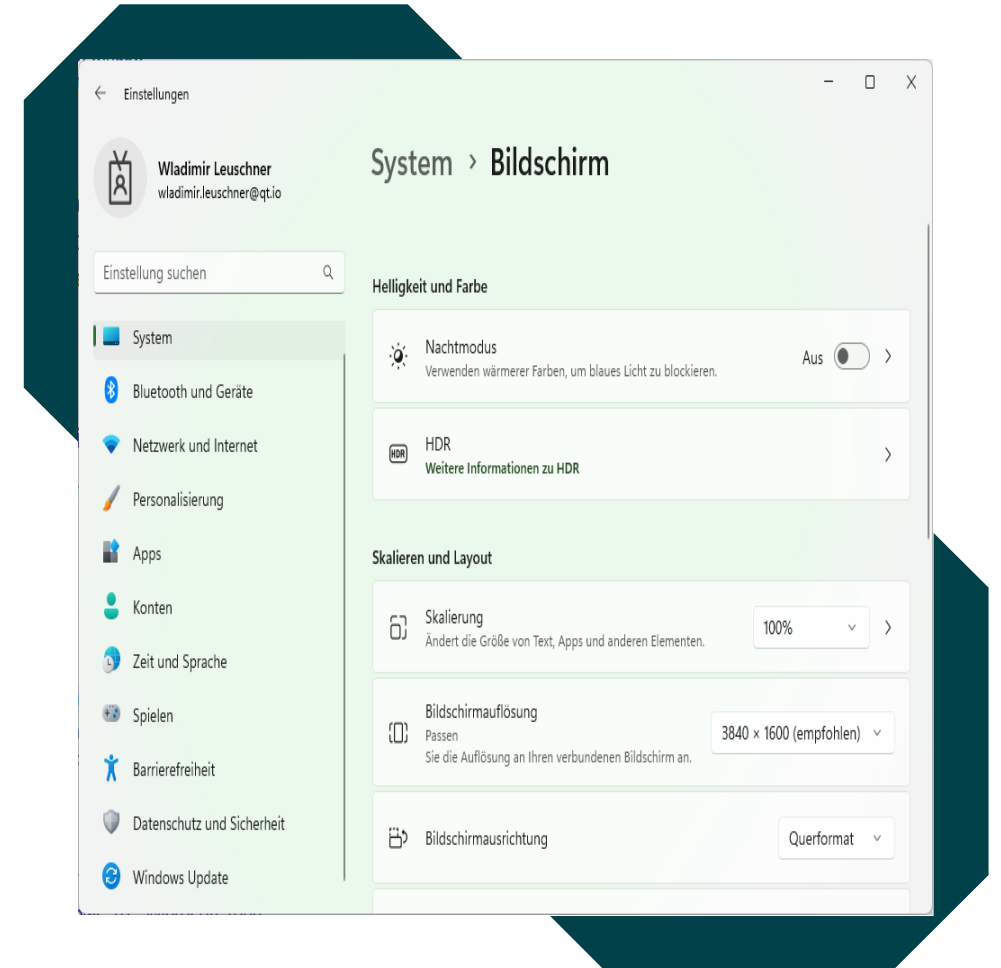
## Windows on ARM

- Support Windows on ARM (WoA) as target
  - Create a native WoA installer
  - Create a QtCreator package for WoA
  - Support the ARM64EC layer on WoA
- 
- Oliver Wolf: [oliver.wolff@qt.io](mailto:oliver.wolff@qt.io)

# Future Development

## Transparency Effects

- Microsoft uses transparency effects since Windows Vista
- Current Qt Apps on Windows always draw opaque
- Evaluation how to enable transparency effects in Qt Apps



# Future Development

## Windows packaging

- Add support for MSIX packaging
- Add support for Microsoft Store
  
- Evaluation whether using windeployqt or cmake deployment

# Future Development

.Net with Qt

- Better Documentation
- Make it easier use Qt in existing .Net code
  - Using QML interface with a C# backend
- Tooling to generate boilerplate InterOp code from existing C# Libraries
  
- Miguel Costa: [miguel.costa@qt.io](mailto:miguel.costa@qt.io)

# Future Development

## Windows App SDK Support

- Microsoft provides newer functionality with Windows App SDK
  - WebView2
  - WinUI3
  - App lifecycle Management
- Challenges:
  - How to integrate into CMake
  - Making things work with Win32 API
- Evaluation how to integrate Windows App SDK into Qt



# Feedback, Questions and Discussion

Wladimir Leuschner

E-Mail: [wladimir.leuschner@qt.io](mailto:wladimir.leuschner@qt.io)

