

Qt for **WebAssembly** and Dynamic Linking: Expanding Web Application Potential

Piotr Wierciński
Senior Software Engineer

What is this talk about?

Overview of Qt for WebAssembly

What is dynamic linking

Dynamic linking for WebAssembly

Shipping and deployment of dynamically linked apps

What is Qt for WebAssembly?



THE NEWEST PLATFORM



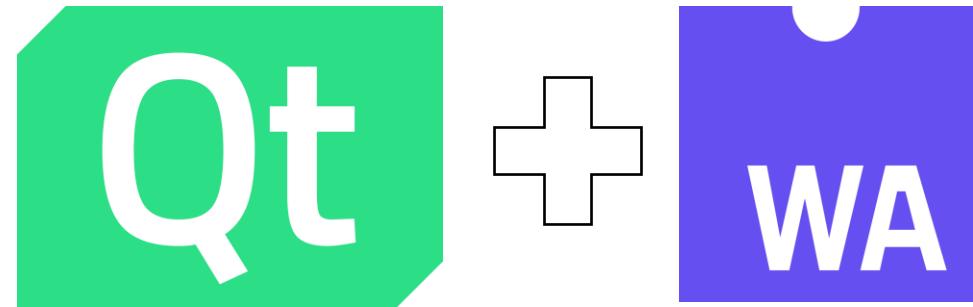
RUNS IN BROWSER



NO INSTALATION



RUN EVERYWHERE



How does Qt for WebAssembly work?

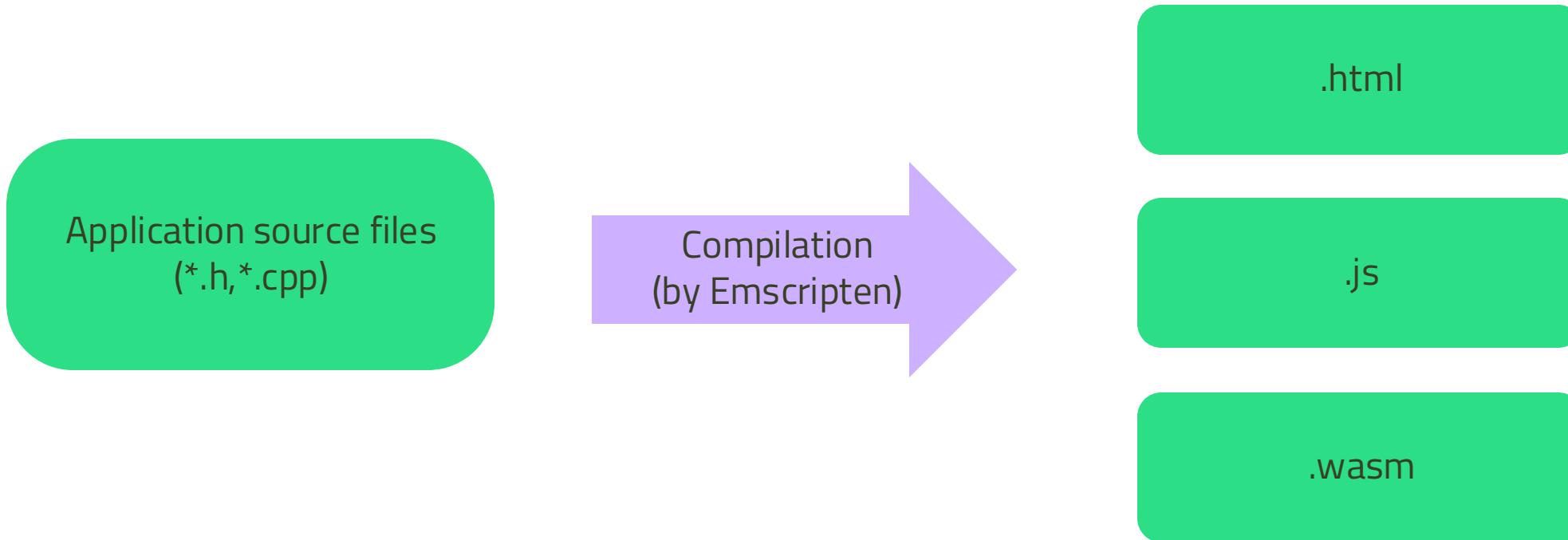
High level overview

Building

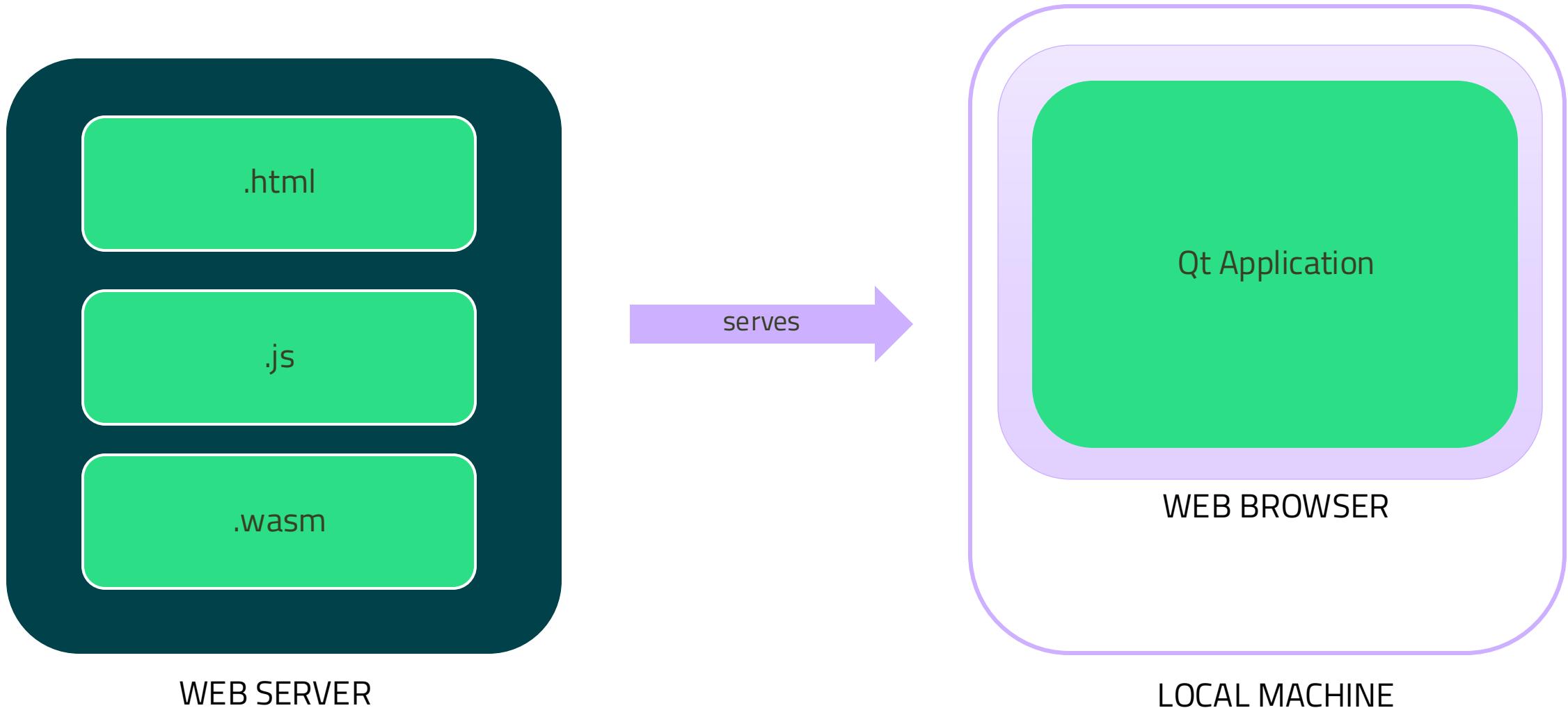
Deploying

Running

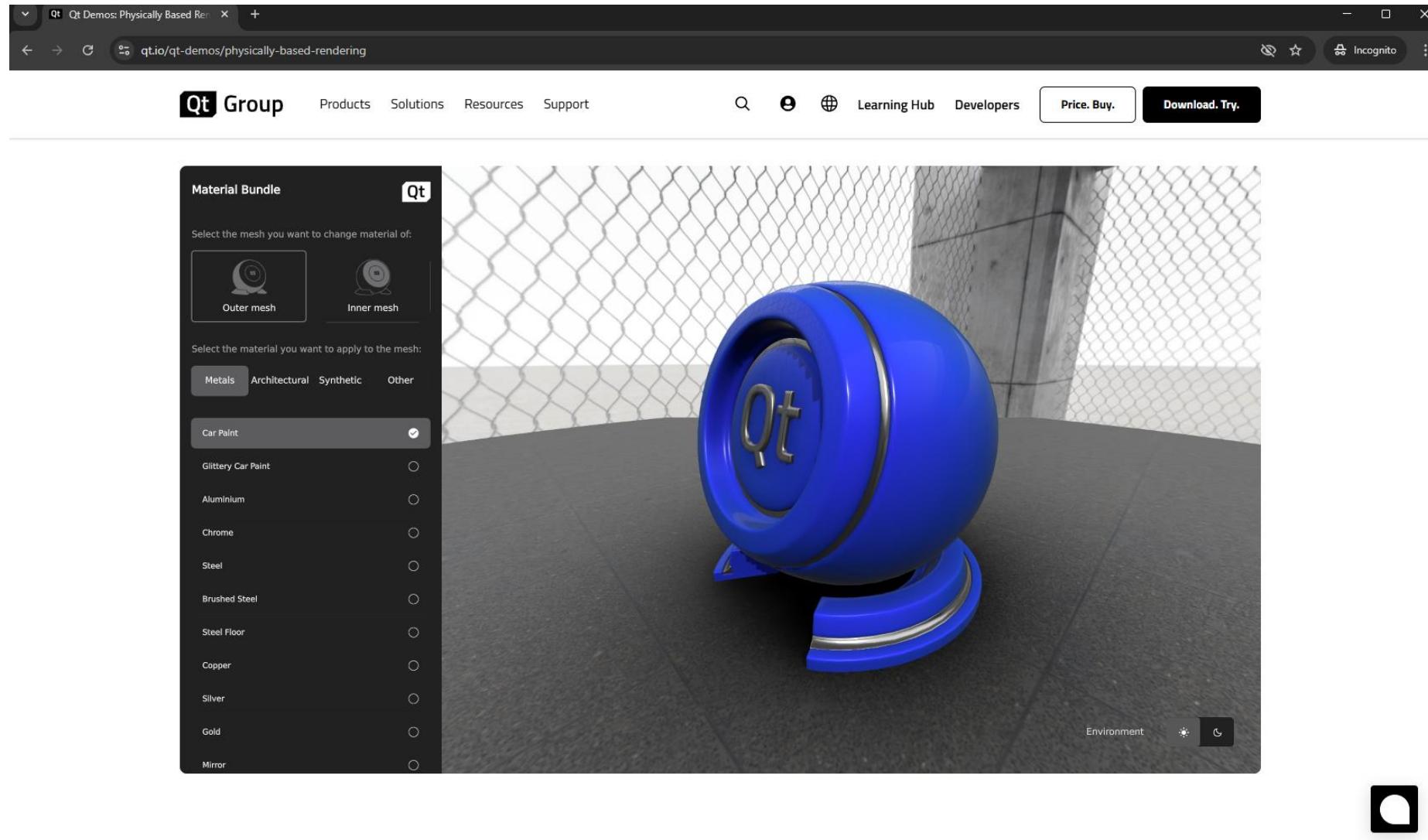
Building WebAssembly application

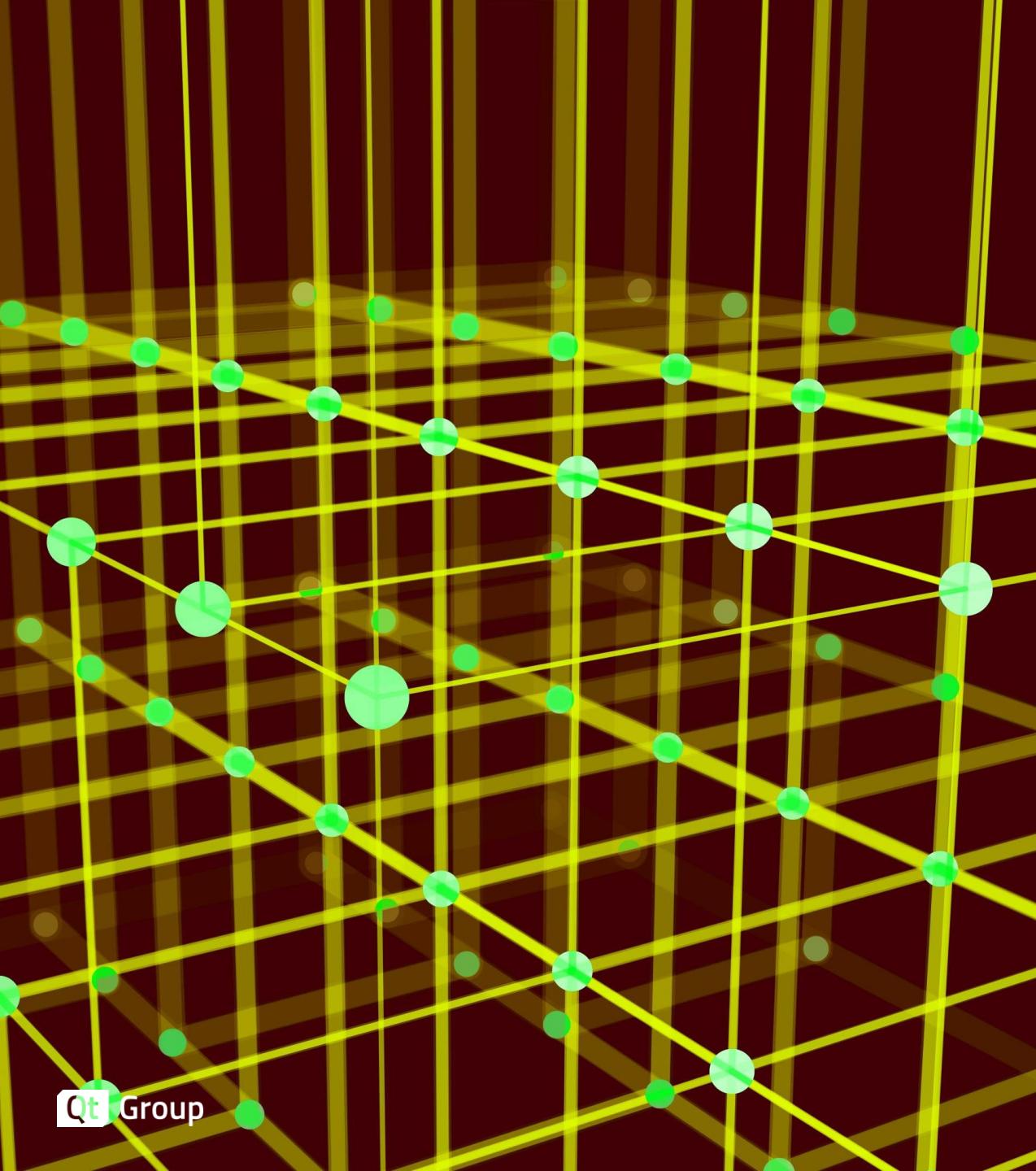


Deploying WebAssembly application



Running WebAssembly application





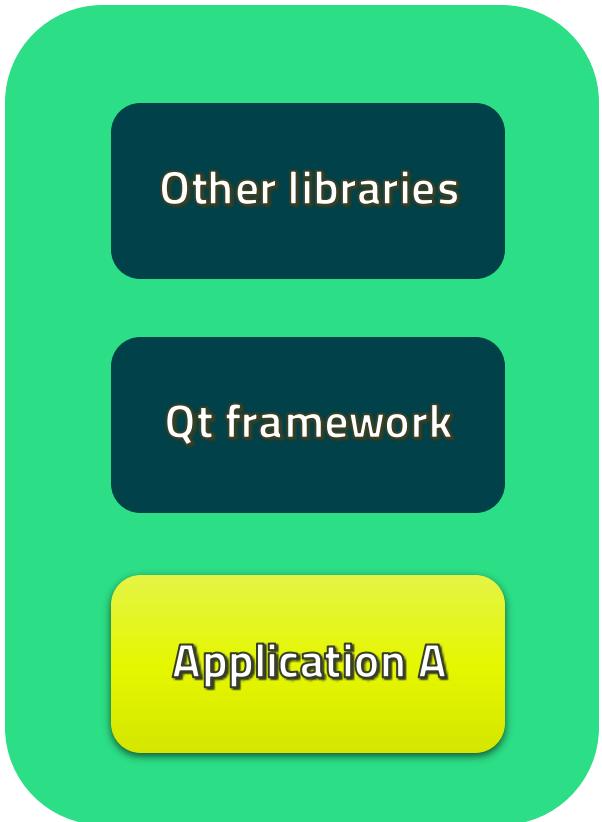
Static linking

Static linking - idea



Single binary

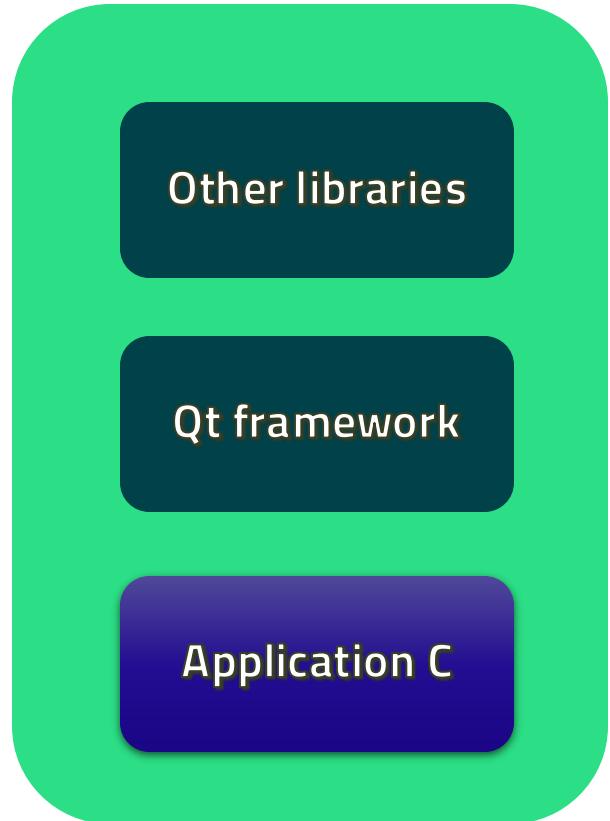
Static linking - deployment



Binary A



Binary B



Binary C

Dynamic linking

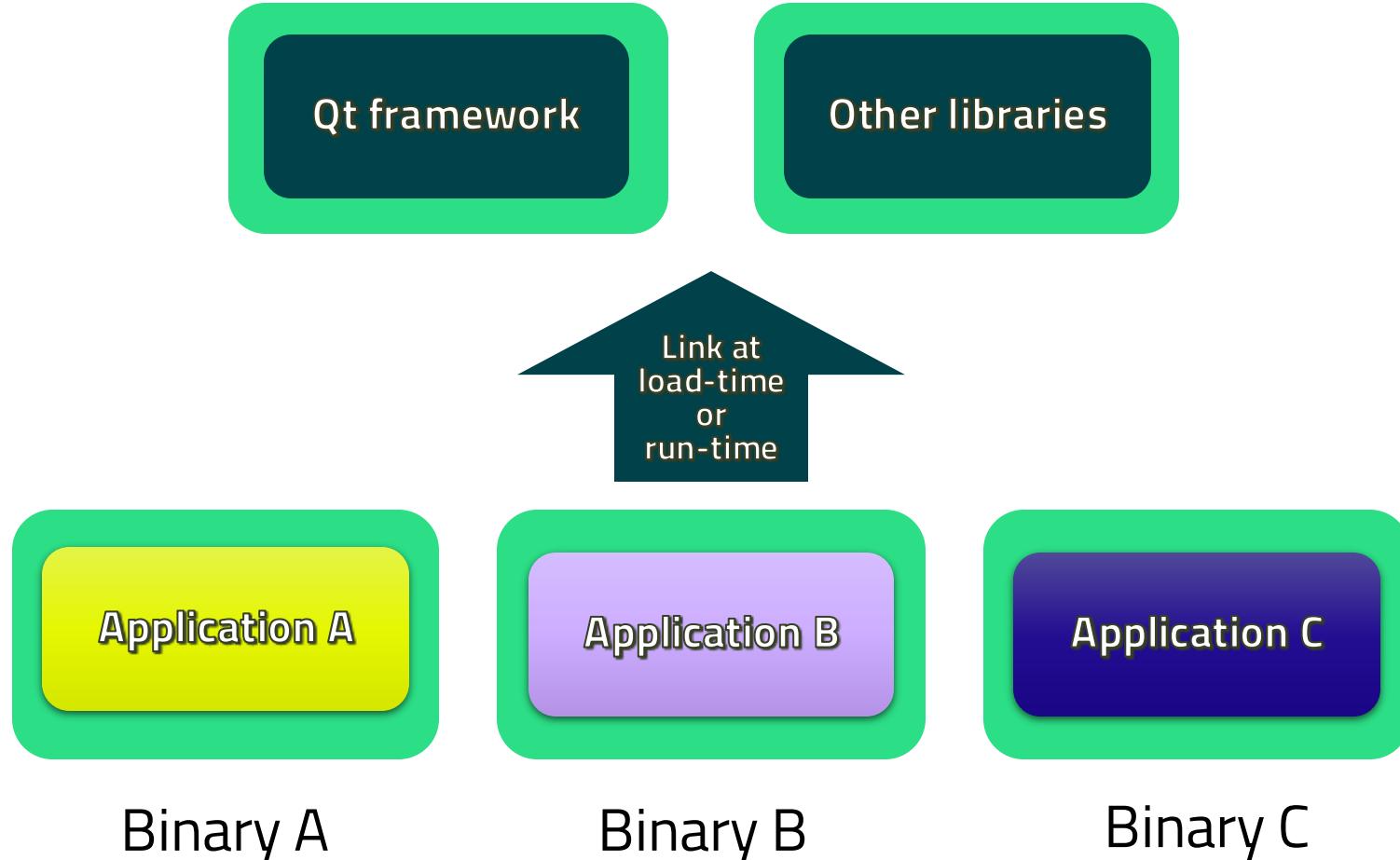


Dynamic linking - idea



Multiple binaries

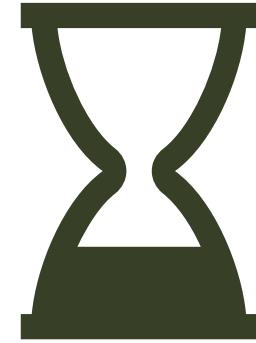
Dynamic linking - deployment



Dynamic linking comes in two flavors

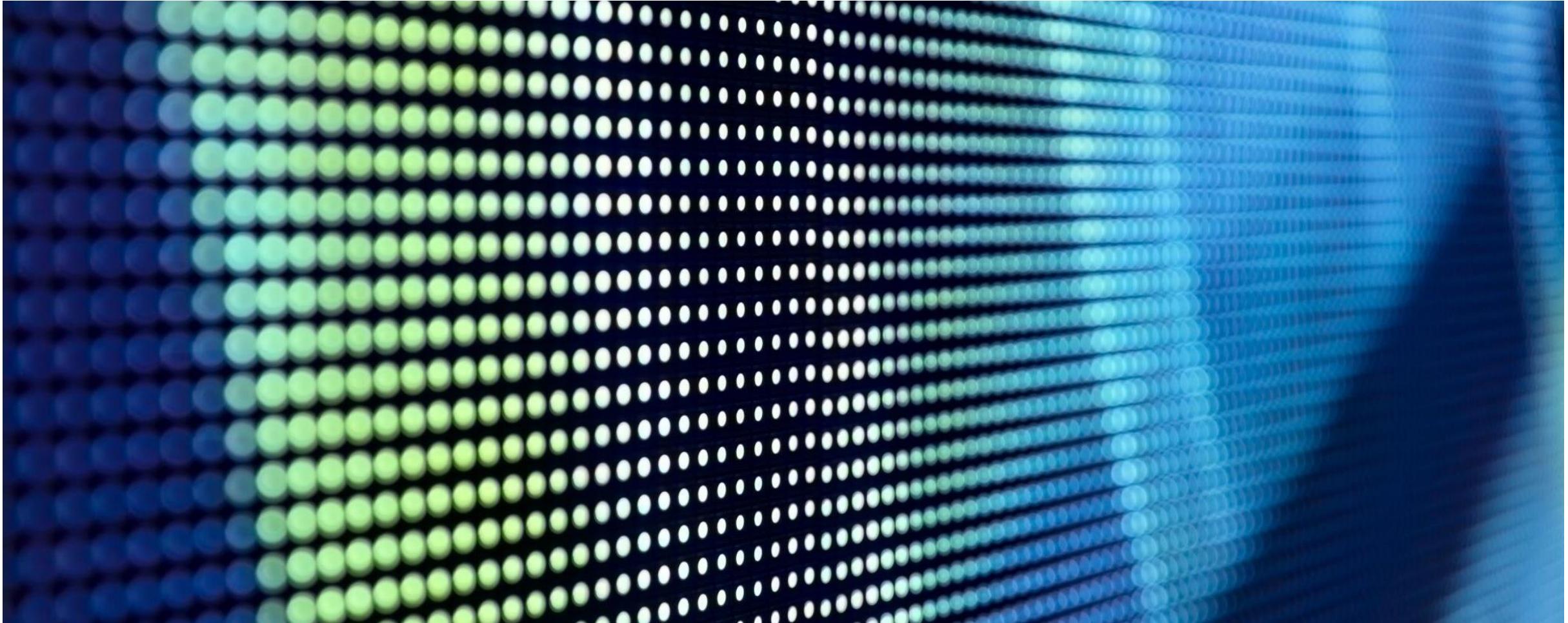


At load-time



At run-time

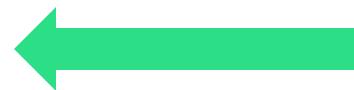
Back to WebAssembly...



Qt for WebAssembly – static linking is default

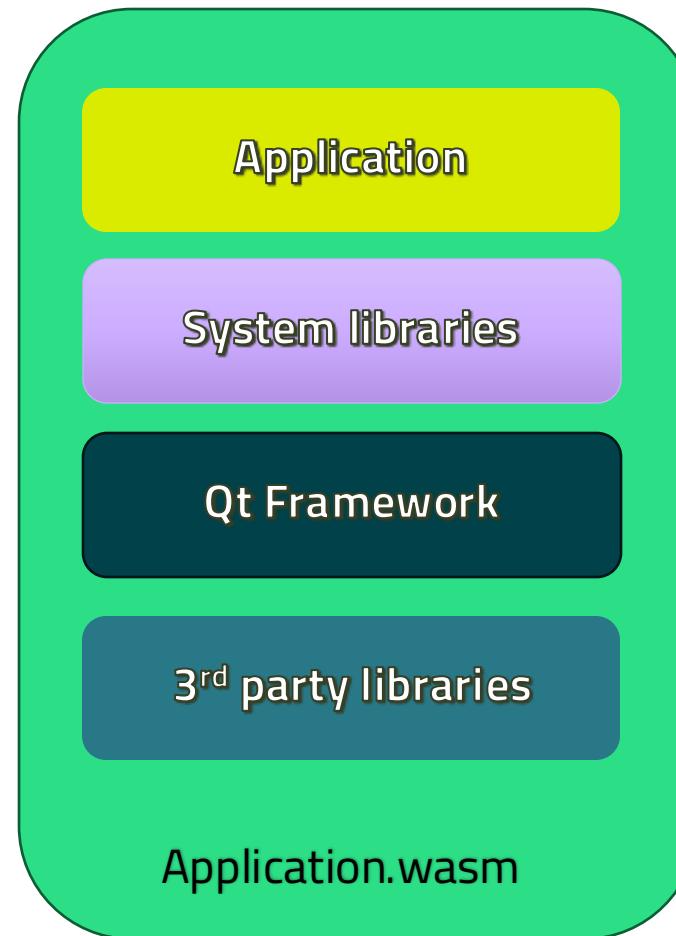


WebAssembly application



Contains compiled code.

Qt for WebAssembly – static linking is default



galleryexample

127.0.0.1:8000/galleryexample.html

Gallery

Network

Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other

Name	Status	Type	Initiator	Size	Time
qtloader.js	200	script	galleryexample.html:76	4.6 kB	3 ms
galleryexample.js	200	script	Other	350 B	23 ms
galleryexample.js	200	script	Other	350 B	29 ms
galleryexample.js	200	script	Other	350 B	22 ms
galleryexample.js	200	script	Other	350 B	24 ms
galleryexample.wasm	200	wasm	galleryexample.js:9	10.5 MB	237 ms
data:image/svg+xml;...	200	svg+xml	galleryexample.js:9	(memory cache)	0 ms
data:image/svg+xml;...	200	svg+xml	galleryexample.js:9	(memory cache)	0 ms
data:image/svg+xml;...	200	svg+xml	galleryexample.js:9	(memory cache)	0 ms
data:image/svg+xml;...	200	svg+xml	galleryexample.js:9	(memory cache)	0 ms

13 requests | 10.6 MB transferred | 35.1 MB resources | Finish: 1.11 s | DOMContentLoaded: 32 ms | Load: 66 ms

Console AI assistance What's new

Default levels ▾ 1 issue: 1 | 2 hidden

Qt::ColorScheme::Dark

qrc:/gallery.qml:48:5: QML Shortcut: Shortcut: Only binding to one of multiple key bindings associated with 1. Use 'sequences: [<key>]' to bind to all of them.

[GroupMarkerNotSet(cbug.com/242999)]:A0F0FB013C340000]Automatic fallback to software WebGL has been deprecated. Please use the --enable-unsafe-swiftshader flag to opt in to lower security guarantees for trusted content.

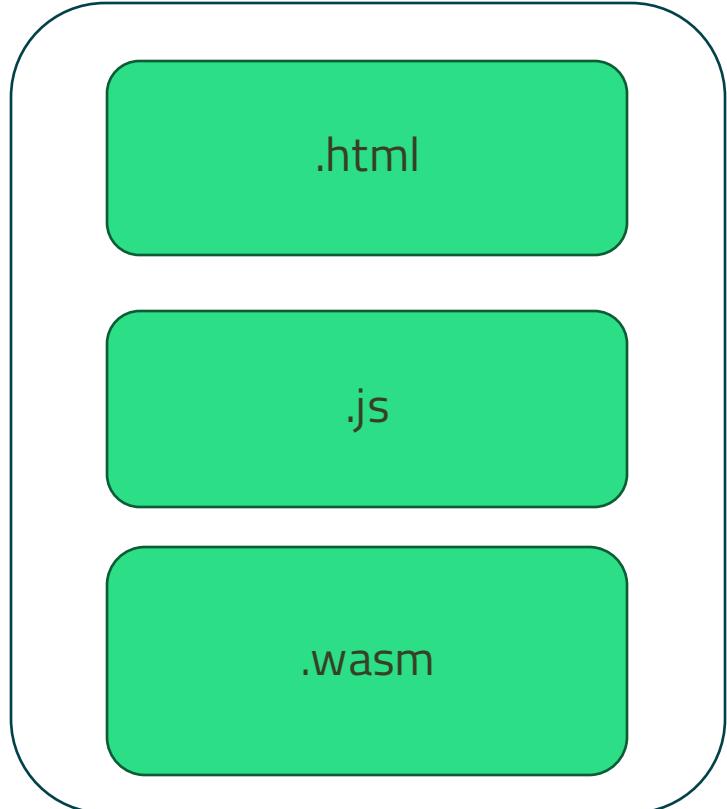
Qt Quick Controls provides a set of controls that can be used to build complete interfaces in Qt Quick.



Name	Status	Type	Initiator	Size	Time
galleryexample.html	200	document	Other	1.6 kB	3 ms
qtlogo.svg	200	svg+xml	galleryexample.htm	1.3 kB	43 ms
galleryexample.js	200	script	galleryexample.htm	73.4 kB	6 ms
qtloader.js	200	script	galleryexample.htm	4.6 kB	3 ms
galleryexample.js	200	script	Other	350 B	23 ms
galleryexample.js	200	script	Other	350 B	29 ms
galleryexample.js	200	script	Other	350 B	22 ms
galleryexample.js	200	script	Other	350 B	24 ms
galleryexample.wasm	200	wasm	galleryexample.js:9.	10.5 MB	237 ms
data:image/svg+xml;...	200	svg+xml	galleryexample.js:9. (memor...		0 ms
data:image/svg+xml;...	200	svg+xml	galleryexample.js:9. (memor...		0 ms
data:image/svg+xml;...	200	svg+xml	galleryexample.js:9. (memor...		0 ms
data:image/svg+xml;...	200	svg+xml	galleryexample.js:9. (memor...		0 ms

13 requests | 10.6 MB transferred | 35.1 MB resources | Finish: 1.11 s | DOMContentLoaded: 32 ms |

Qt for WebAssembly – static vs dynamic



Static linking



Dynamic linking

Qt for WebAssembly – dynamic linking



There can be:

- single MAIN_MODULE (.wasm)
- multiple SIDE_MODULE (.so)

Qt for WebAssembly – dynamic linking

Example “quickgallery”:



The screenshot shows a web browser window titled "galleryexample" displaying the URL "127.0.0.1:8000/galleryexample.html". The page content is the "Gallery" section of the Qt Quick Controls 2 documentation, featuring a large green button with the word "Qt" and a descriptive text block about Qt Quick Controls.

The browser's developer tools Network tab is open, showing the network traffic for the page. The timeline at the top indicates the duration of each request. Below the timeline is a table listing the resources loaded:

Name	Status	Type	Initiator	Size	Time
libQt6QuickControls2Impl.so	200	Fetch	galleryexample.js:9	61.4 kB	45 ms
libQt6QuickControls2Basic.so	200	Fetch	galleryexample.js:9	190 kB	18 ms
libQt6QuickControls2BasicStyleImpl.so	200	Fetch	galleryexample.js:9	10.1 kB	43 ms
libQt6QuickShapes.so	200	Fetch	galleryexample.js:9	94.1 kB	13 ms
libQt6LabsQmlModels.so	200	Fetch	galleryexample.js:9	36.3 kB	3 ms
libQt6QuickControls2Fusion.so	200	Fetch	galleryexample.js:9	166 kB	9 ms
libQt6QuickControls2FusionStyleImpl.so	200	Fetch	galleryexample.js:9	38.2 kB	43 ms
libQt6QuickControls2FluentWinUI3StyleImpl.so	200	Fetch	galleryexample.js:9	33.7 kB	11 ms
libQt6QuickControls2Imagine.so	200	Fetch	galleryexample.js:9	715 kB	15 ms
libQt6QuickControls2ImagineStyleImpl.so	200	Fetch	galleryexample.js:9	7.9 kB	42 ms
libQt6QuickControls2Universal.so	200	Fetch	galleryexample.js:9	160 kB	16 ms
libQt6QuickControls2UniversalStyleImpl.so	200	Fetch	galleryexample.js:9	27.2 kB	43 ms
libQt6QuickEffects.so	200	Fetch	galleryexample.js:9	222 kB	13 ms
libQt6QuickLayouts.so	200	Fetch	galleryexample.js:9	45.2 kB	4 ms
libQt6QmlCore.so	200	Fetch	galleryexample.js:9	26.4 kB	3 ms
data:image/svg+xml;...	200	svg+xml	Other	(memory cache)	0 ms
data:image/svg+xml;...	200	svg+xml	Other	(memory cache)	0 ms
data:image/svg+xml;...	200	svg+xml	Other	(memory cache)	0 ms
data:image/svg+xml;...	200	svg+xml	Other	(memory cache)	0 ms

At the bottom of the Network tab, there is a summary: 105 requests | 14.3 MB transferred | 51.2 MB resources | Finish: 3.31 s | DOMContentLoaded: 148 ms | Load: 149 ms.

The Console tab shows a warning message: `[GroupMarkerNotSet(crbbug.com/242999)!A0F0FB013C340000]Automatic fallback to software WebGL has been deprecated. Please use --enable-unsafe-swiftshader flag to opt in to lower security guarantees for trusted content.`



Name	Status	Type	Initiator	Size	Time
galleryexample.wasm	200	wasm	galleryexample.js:9	666 kB	59 ms
libQt6Core.so	200	fetch	galleryexample.js:9	1.5 MB	55 ms
libQt6QuickControls2.so	200	fetch	galleryexample.js:9	20.3 kB	3 ms
libQt6QuickTemplates2.so	200	fetch	galleryexample.js:9	425 kB	54 ms
libQt6Quick.so	200	fetch	galleryexample.js:9	1.7 MB	81 ms
libQt6QmlMeta.so	200	fetch	galleryexample.js:9	34.8 kB	43 ms
libQt6QmlModels.so	200	fetch	galleryexample.js:9	190 kB	48 ms
libQt6Qml.so	200	fetch	galleryexample.js:9	1.2 MB	80 ms
libQt6Network.so	200	fetch	galleryexample.js:9	216 kB	48 ms
libQt6OpenGL.so	200	fetch	galleryexample.js:9	111 kB	6 ms
libQt6Gui.so	200	fetch	galleryexample.js:9	2.2 MB	104 ms
libqsqlite.so	200	fetch	galleryexample.js:9	588 kB	42 ms
libq wasm.so	200	fetch	galleryexample.js:9	624 kB	39 ms
libqwbmp.so	200	fetch	galleryexample.js:9	4.0 kB	15 ms
libqico.so	200	fetch	galleryexample.js:9	7.3 kB	18 ms
libqjpeg.so	200	fetch	galleryexample.js:9	102 kB	26 ms
libqicns.so	200	fetch	galleryexample.js:9	10.7 kB	17 ms
libqtiff.so	200	Fetch	galleryexample.js:9	103 kB	30 ms

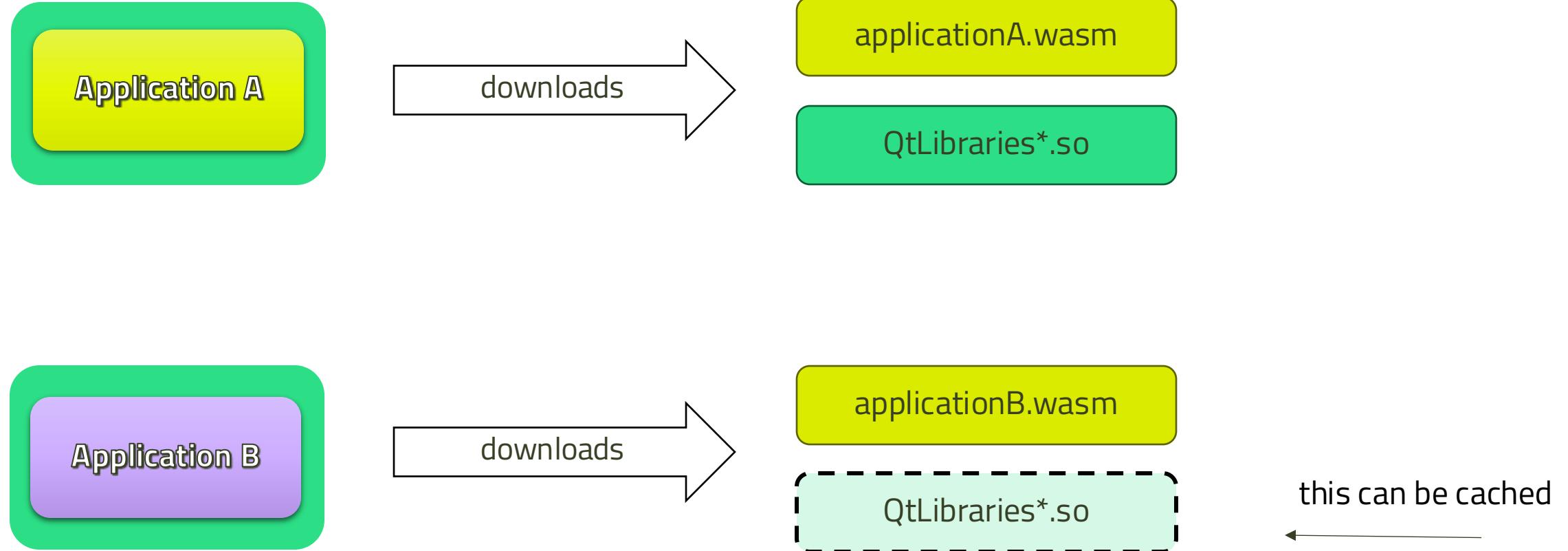
105 requests | 14.3 MB transferred | 51.2 MB resources | Finish: 3.31 s | DOMContentLoaded: 148 ms | Load: 148 ms

Why use dynamic linking?

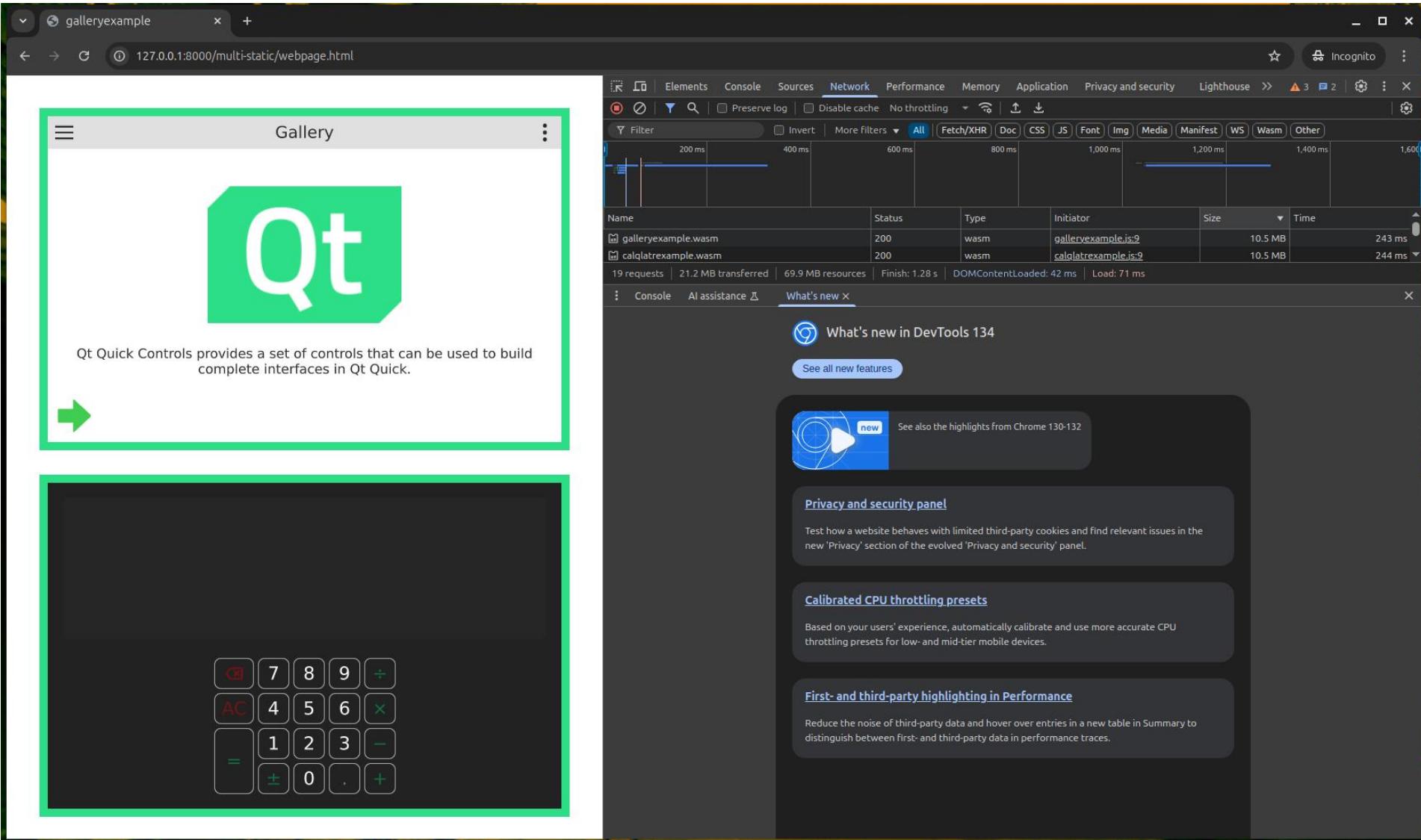
- Smaller downloads size
- Loading modules at runtime
- LGPL licensing for third-party libraries
- Faster development times



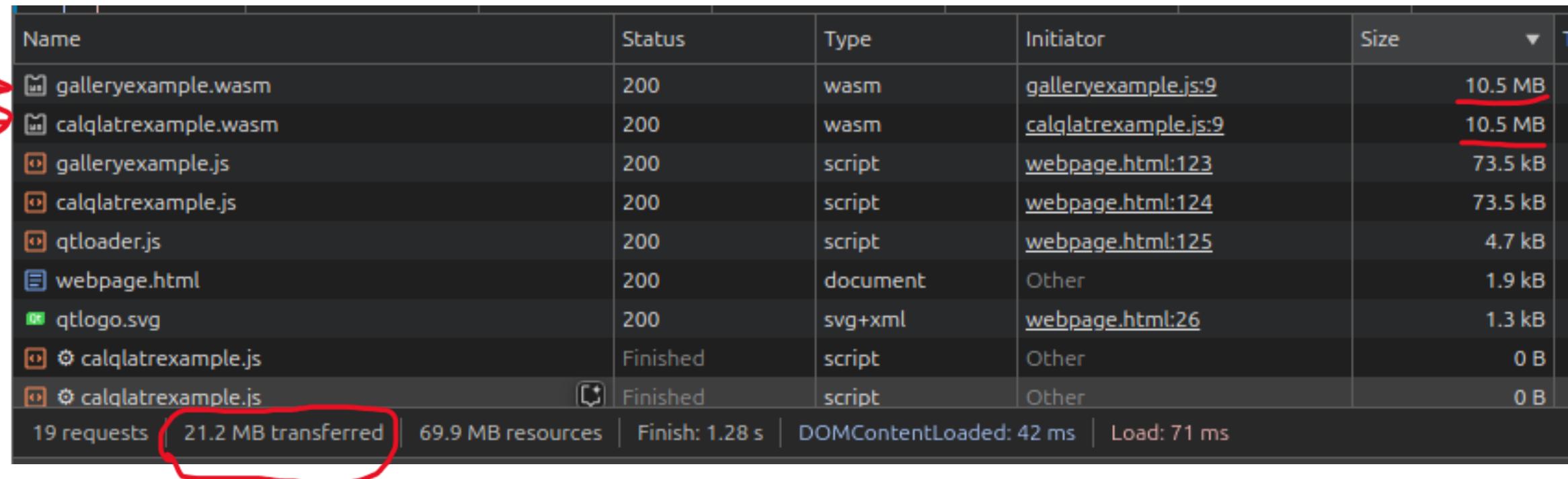
Smaller downloads size with multiple applications



Two applications with static linking:



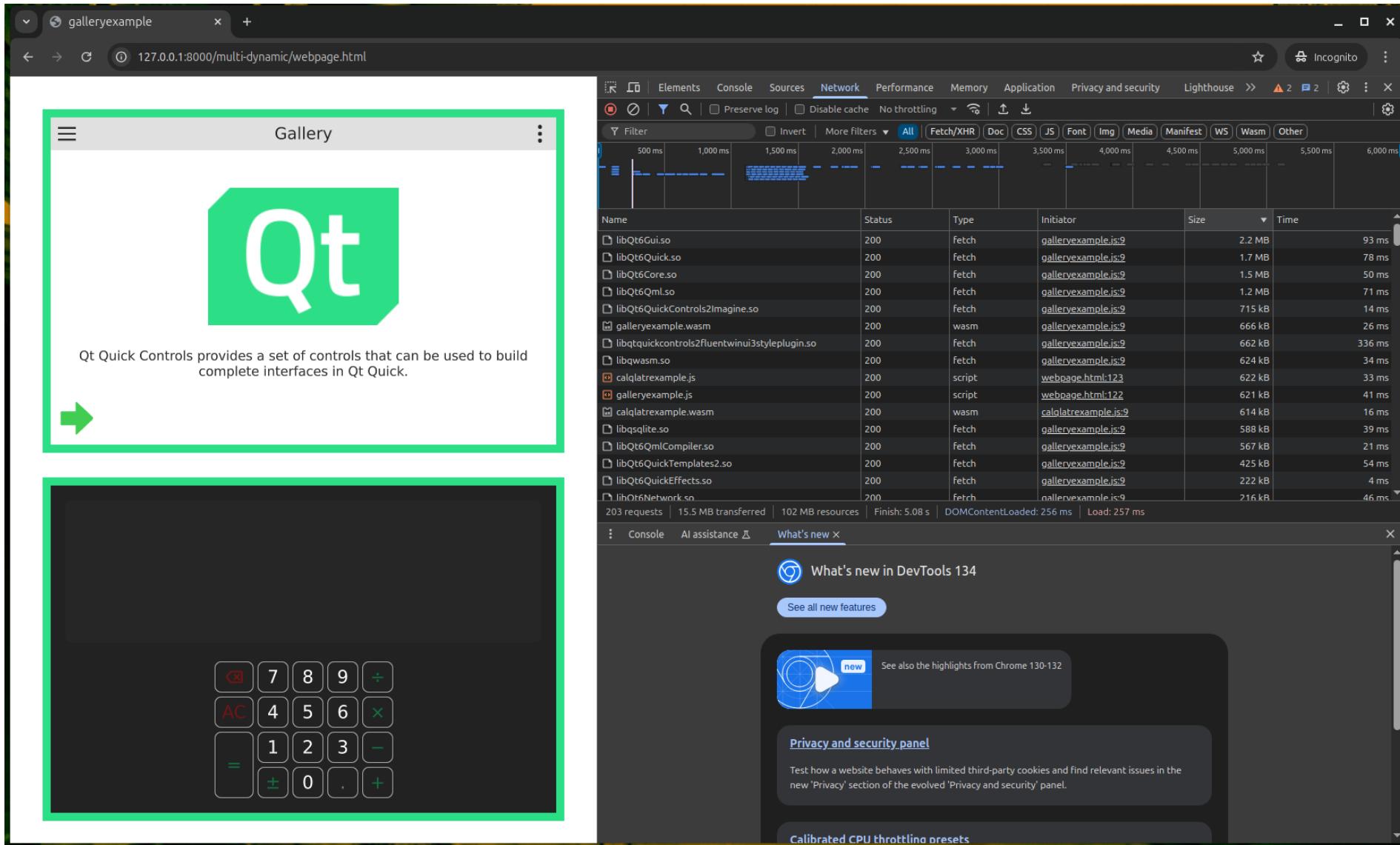
Two applications with static linking:



Name	Status	Type	Initiator	Size
galleryexample.wasm	200	wasm	galleryexample.js:9	10.5 MB
calqlatrexample.wasm	200	wasm	calqlatrexample.js:9	10.5 MB
galleryexample.js	200	script	webpage.html:123	73.5 kB
calqlatrexample.js	200	script	webpage.html:124	73.5 kB
qtloader.js	200	script	webpage.html:125	4.7 kB
webpage.html	200	document	Other	1.9 kB
qtlogo.svg	200	svg+xml	webpage.html:26	1.3 kB
calqlatrexample.js	Finished	script	Other	0 B
calqlatrexample.js	Finished	script	Other	0 B

19 requests | 21.2 MB transferred | 69.9 MB resources | Finish: 1.28 s | DOMContentLoaded: 42 ms | Load: 71 ms

Two applications with dynamic linking:



Two applications with dynamic linking:

The screenshot shows a browser developer tools Network tab with a list of resources. A red arrow points to the entry for 'galleryexample.wasm' (size 666 kB), and another red arrow points to the entry for 'calqlatrexample.wasm' (size 614 kB). Both entries are highlighted with a red underline. At the bottom of the table, a summary row indicates 203 requests, 15.5 MB transferred, 102 MB resources, a finish time of 5.08 s, a DOMContentLoaded time of 256 ms, and a load time of 257 ms.

Name	Status	Type	Initiator	Size
libQt6Qml.so	200	fetch	galleryexample.js:9	1.2 MB
libQt6QuickControls2Imagine.so	200	fetch	galleryexample.js:9	715 kB
galleryexample.wasm	200	wasm	galleryexample.js:9	666 kB
libqtquickcontrols2fluentwinui3styleplugin.so	200	fetch	galleryexample.js:9	662 kB
libq wasm.so	200	fetch	galleryexample.js:9	624 kB
calqlatrexample.js	200	script	webpage.html:123	622 kB
galleryexample.js	200	script	webpage.html:122	621 kB
calqlatrexample.wasm	200	wasm	calqlatrexample.js:9	614 kB
libsqlite.so	200	fetch	galleryexample.js:9	588 kB
libQt6QmlCompiler.so	200	fetch	galleryexample.js:9	567 kB
libQt6QuickTemplates2.so	200	fetch	galleryexample.js:9	425 kB
libQt6Core.so	200	fetch	galleryexample.js:9	222 kB
203 requests	15.5 MB transferred	102 MB resources	Finish: 5.08 s	DOMContentLoaded: 256 ms
				Load: 257 ms

21.2MB -> 15.5MB (-27%)

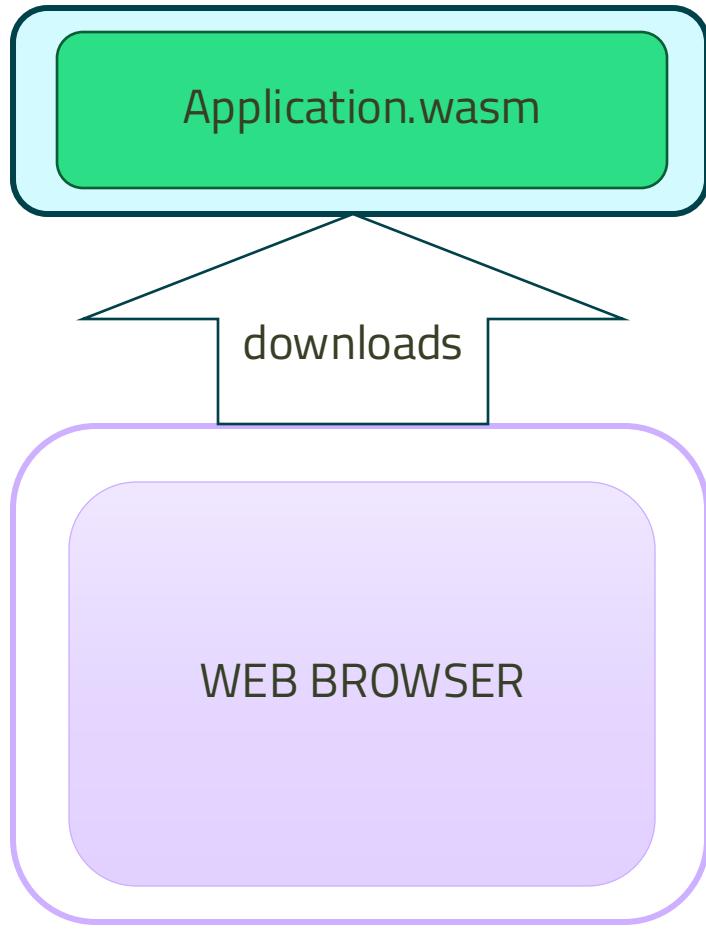
Two applications with dynamic linking:

libQt6QuickControls2Fusion.so	200	fetch	galleryexample.js:9	166 kB
libQt6QuickControls2Fusion.so	200	fetch	calqlatrexample.js:9	(disk cache)
libQt6QuickControls2FusionStyleImpl.so	200	fetch	galleryexample.js:9	38.3 kB
libQt6QuickControls2FusionStyleImpl.so	200	fetch	calqlatrexample.js:9	(disk cache)



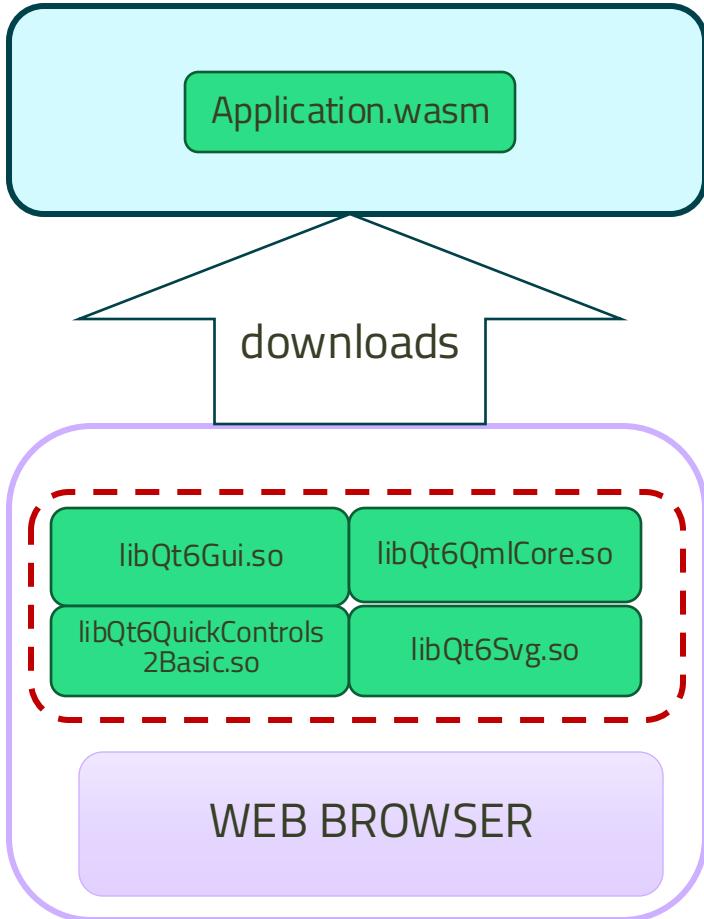
Smaller downloads size with single app

SERVER



LOCAL MACHINE

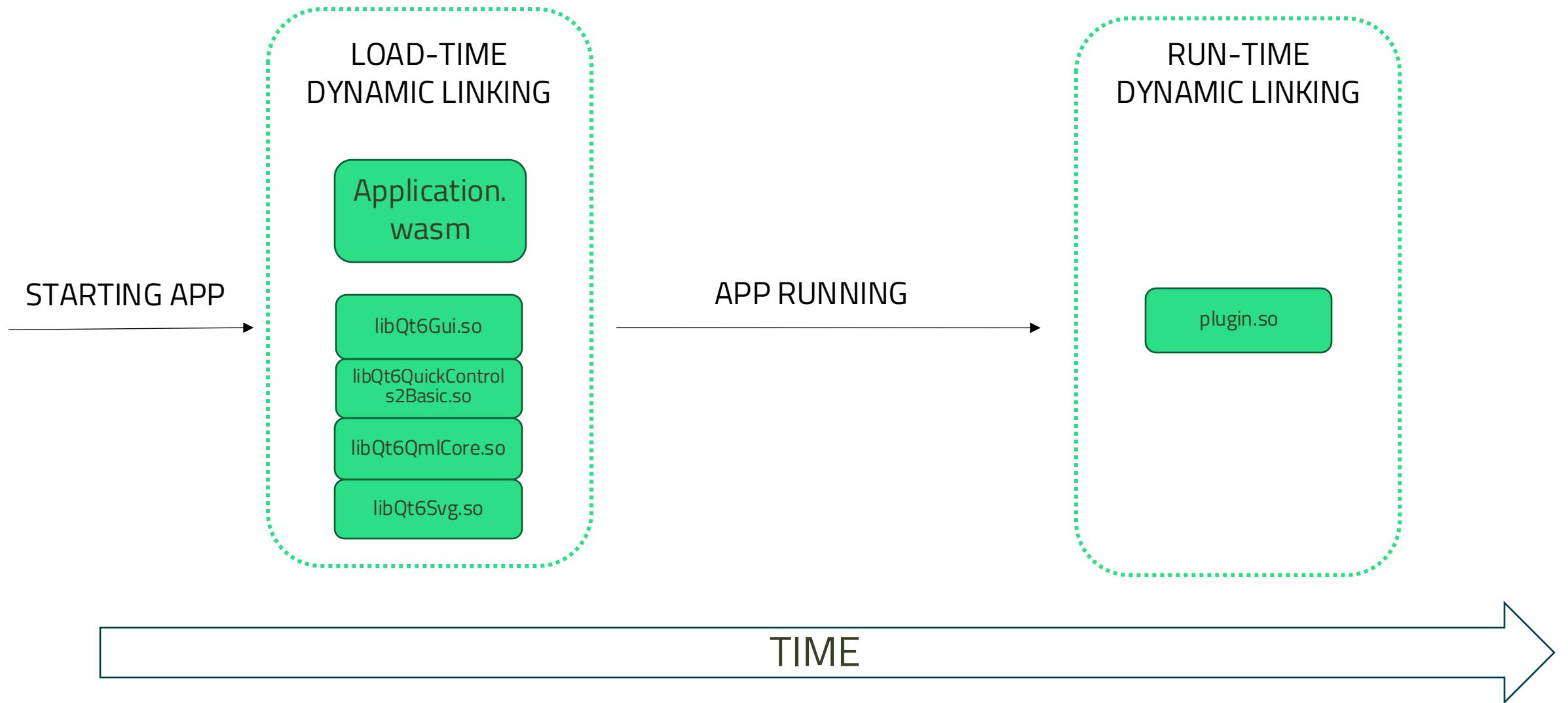
STATIC LINKING



DYNAMIC LINKING

PROXY
SERVER

Loading modules at runtime



Loading modules at runtime

LGPL licensing for third party libraries



Faster development times





Some technicalities...



Filesystem

1. Virtual filesystem
2. Preloading
3. Preload JSON files

App.html

```
const instance = await qtLoad({
    qt: {
        onLoaded: () => showUi1(screen1),
        onExit: exitData =>
        {
            status.innerHTML = 'Application exit';
            status.innerHTML +=
            ||| exitData.code !== undefined ? ` with code ${exitData.code}` : '';
            status.innerHTML +=
            ||| exitData.text !== undefined ? ` (${exitData.text})` : '';
            showUi1(spinner1);
        },
        entryFunction: window.galleryexample_entry,
        containerElements: [screen1],
        preload: ['qt_plugins.json', 'qt_qml_imports.json'],
    }
});
```

qt_plugins.json

```
{  
    "source": "$QTDIR/plugins/imageformats/libqtiff.so",  
    "destination": "/qt/plugins/imageformats/libqtiff.so"  
},  
{  
    "source": "$QTDIR/plugins/imageformats/libqsvg.so",  
    "destination": "/qt/plugins/imageformats/libqsvg.so"  
},  
{  
    "source": "$QTDIR/plugins/imageformats/libqtga.so",  
    "destination": "/qt/plugins/imageformats/libqtga.so"  
},  
{  
    "source": "$QTDIR/plugins/imageformats/libqgif.so",  
    "destination": "/qt/plugins/imageformats/libqgif.so"  
},  
{  
    "source": "$QTDIR/plugins/imageformats/libqwebp.so",  
    "destination": "/qt/plugins/imageformats/libqwebp.so"  
},
```

qt_qml_imports.json

```
{  
    "source": "$QTDIR/qml/QtQuick/Controls/Basic/impl/qmldir",  
    "destination": "/qt/qml/QtQuick/Controls/Basic/impl/qmldir"  
},  
{  
    "source": "$QTDIR/qml/QtQuick/Shapes/libqmlshapesplugin.so",  
    "destination": "/qt/qml/QtQuick/Shapes/libqmlshapesplugin.so"  
},  
{  
    "source": "$QTDIR/qml/QtQuick/Shapes/qmldir",  
    "destination": "/qt/qml/QtQuick/Shapes/qmldir"  
},  
{  
    "source": "$QTDIR/qml/Qt/labs/qmlmodels/liblabsmodelsplugin.so",  
    "destination": "/qt/qml/Qt/labs/qmlmodels/liblabsmodelsplugin.so"  
},  
{  
    "source": "$QTDIR/qml/Qt/labs/qmlmodels/qmldir",  
    "destination": "/qt/qml/Qt/labs/qmlmodels/qmldir"  
},  
{  
    "source": "$QTDIR/qml/QtQml/Models/libmodelsplugin.so",  
    "destination": "/qt/qml/QtQml/Models/libmodelsplugin.so"  
},
```

Run-time dynamic linking

1. Not supported by Qt directly
2. Need to use Emscripten API
(dlopen)



Dynamic linking limitations

- Worse dead code elimination
- Multithreading is experimental
- Slightly worse performance
- More complicated setup



Can I use it now?

Qt 6.8

- Technical Preview
- Requires 2 patches to Emscripten
- Generating preload files with two python helper scripts

Qt 6.9

- Technical Preview
- Requires 1 patch to Emscripten
- Generating preload files with one bash script

Qt 6.10 ???

- Out of Technical Preview
- Using upstream Emscripten
- Deployment with qwasmdeploy

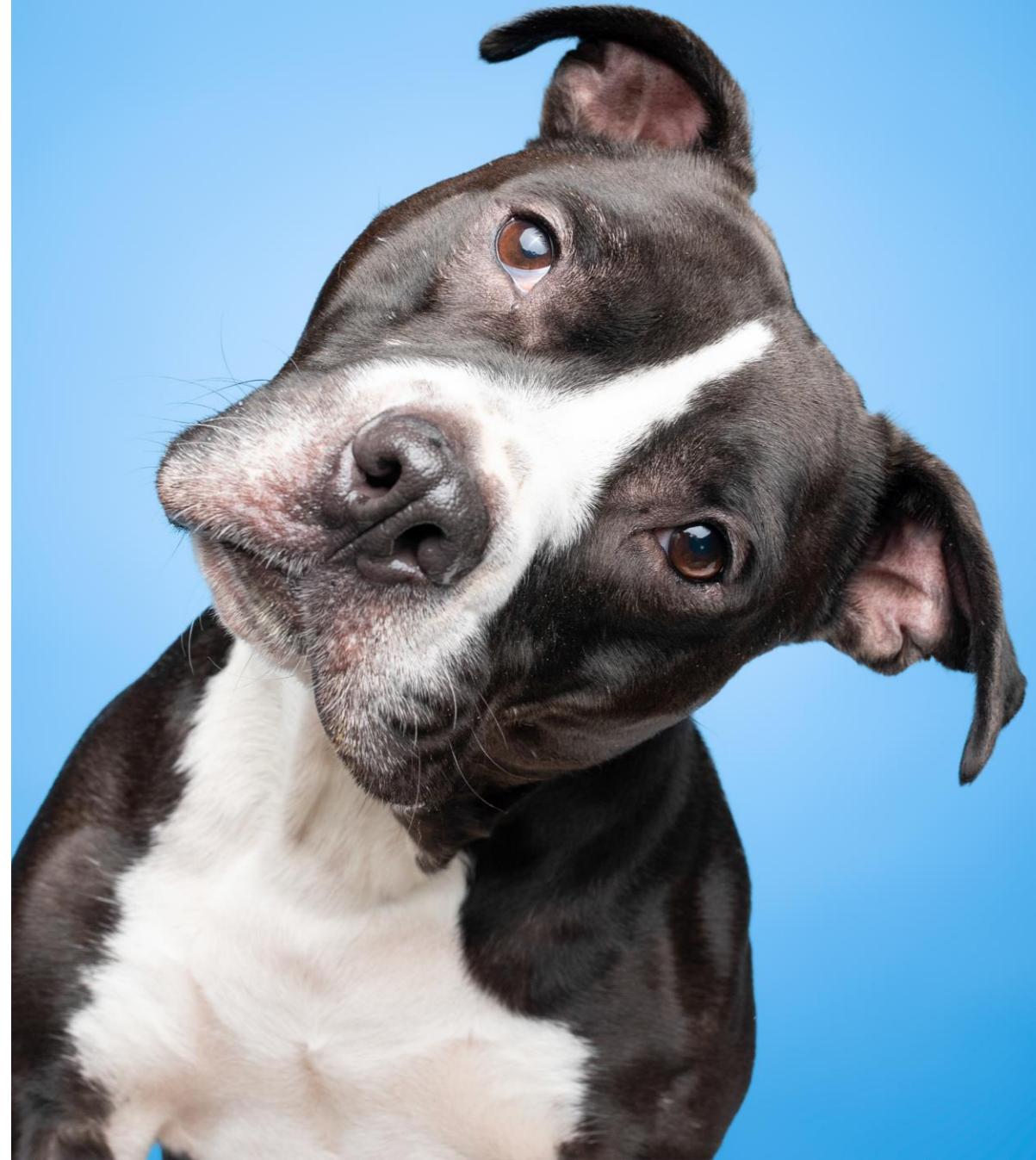
How to do it in Qt 6.9

Process is documented on:

<https://doc.qt.io/qt-6/wasm.html>.

Procedure:

1. Build Emscripten
2. Build Qt as host
3. Build Qt for WebAssembly
4. Build your application
5. Generate preload files
6. Deploy on web server



Example of deployment

```
4.0K May  2 17:44 .
4.0K May  2 17:16 ..
5.2K May  2 17:16 app.html
5.4M May  2 17:16 app.js
1.8M May  2 17:16 app.wasm
671 May  2 17:16 generate_default_preloads_for_galleryexample.sh
  71 May  2 17:16 qt -> /home/plotr-wiercinski/repos/qt6/build/installer
   21 May  2 17:16 qt.conf
  12K May  2 17:16 qtloader.js
  1.7K May  2 17:16 qtlogo.svg
  1.9K May  2 17:16 qt_plugins.json
  7.1K May  2 17:16 qt_qml_imports.json
```

Questions?

Slides

<https://github.com/laminowany/Slides>



Piotr Wierciński
piotr.wiercinski@qt.io

THANK YOU

QtWS²⁵