

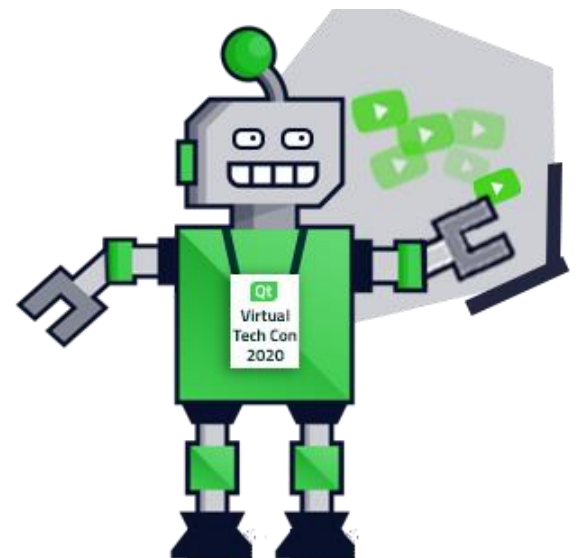
Qt

Virtual Tech Con 2020

Qt Licensing Explained

What are your options and what should you consider?

May 2020





Team

- › **Thilak Ramanna** Territory Sales Director
- › **Somnath Dey** Regional Sales Manager



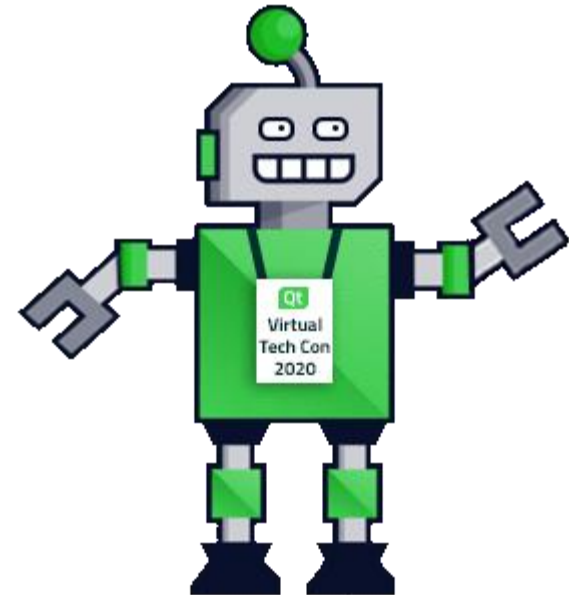
Agenda

Qt Commercial License

Qt Open Source License

Open Source vs.
Commercial

Qt Product Licensing



Qt dual licensing model

Open Source Qt

- Under GPLv3 and LGPLv3 – Limitations and obligations
- Free of charge
- Target desktop and mobile out of box
- Embedded targets need DIY work

Commercial Qt

Qt for Device Creation

- Target all devices, including embedded
- Additional distribution License for each device

Qt for Application Development

- Feature wise same as OSS Qt

Other commercial Qt products

- UI designer offering (Qt Design Studio, Qt 3D Studio)
- Qt for MCU
- Qt Safe Renderer
- Add-ons – Qt M2M Protocols, Qt Automotive Suite
- Professional Services



Qt Commercial License

Accelerate and Protect your Investment

No need to comply with (L)GPL restrictions

- Build Closed/locked down devices
- Software patents, DRM or other technical reasons
- Contamination

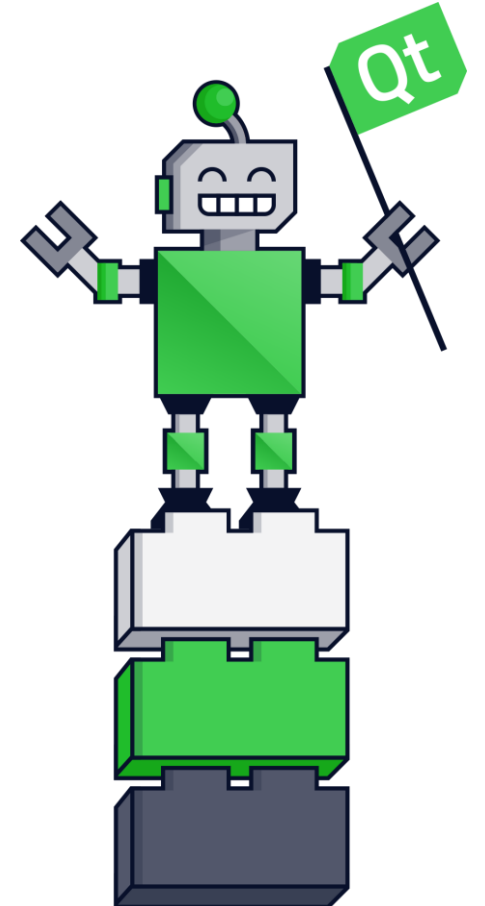
Freedom to modify and compile source codes

- Static linking
- Make libraries compact (Memory efficiency)
- Build competitive advantage without sharing changes

Keep Qt usage confidential

- LGPL usage of Qt needs to be public knowledge

Commercial only offerings and value-added functionality available



Patents and IPR



Intellectual property is increasingly important

Past: Products of the factory

Future: Products of the mind



Patents protect inventions; copyrights protect expression

Copyrights will only protect the way software is written

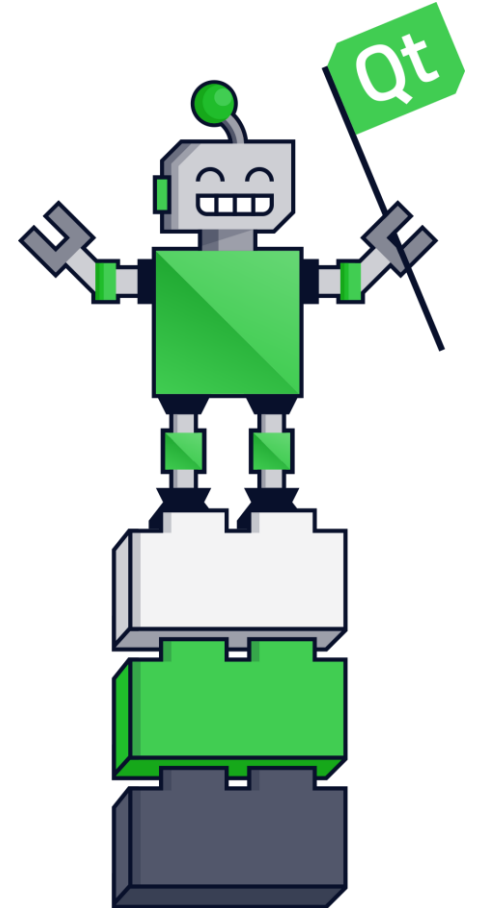
Patents are the only way to secure software inventions



Software patents will protect the R&D investments put down in a product



A commercial license for Qt will not interfere with your IPR and patents



Locked Devices



Locking down devices (a.k.a. *Tivoization*)

End users not allowed to inspect or change the software

Locking of devices is prohibited by LGPLv3



Crucial for some applications

E.g. safety critical devices, medical equipment, military equipment

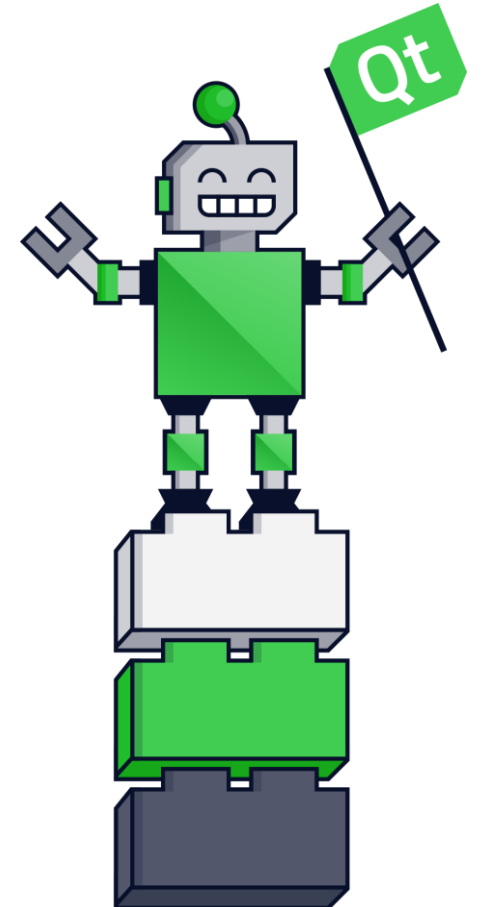


Other benefits

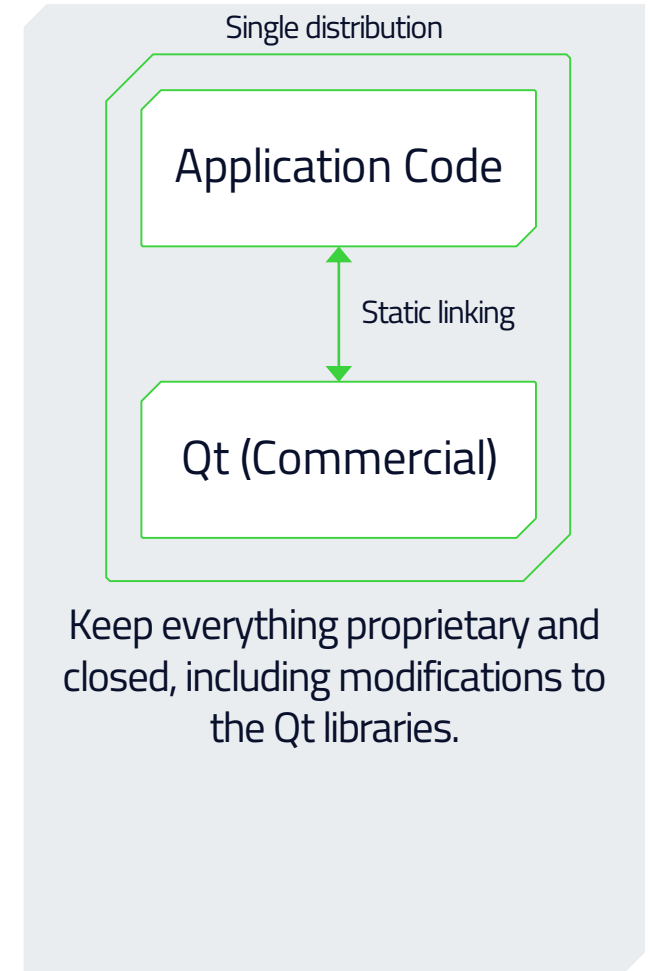
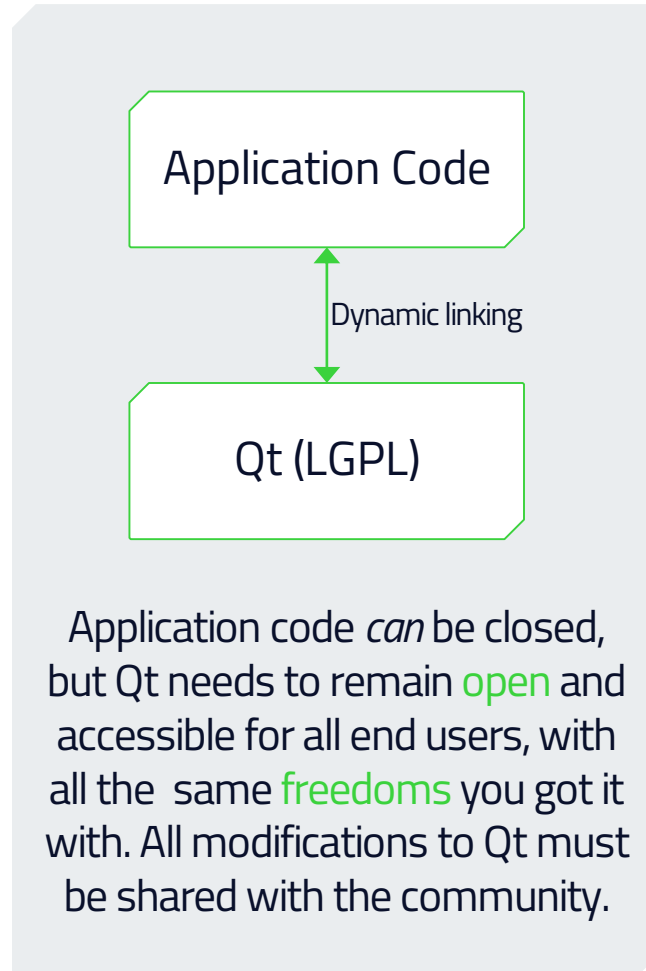
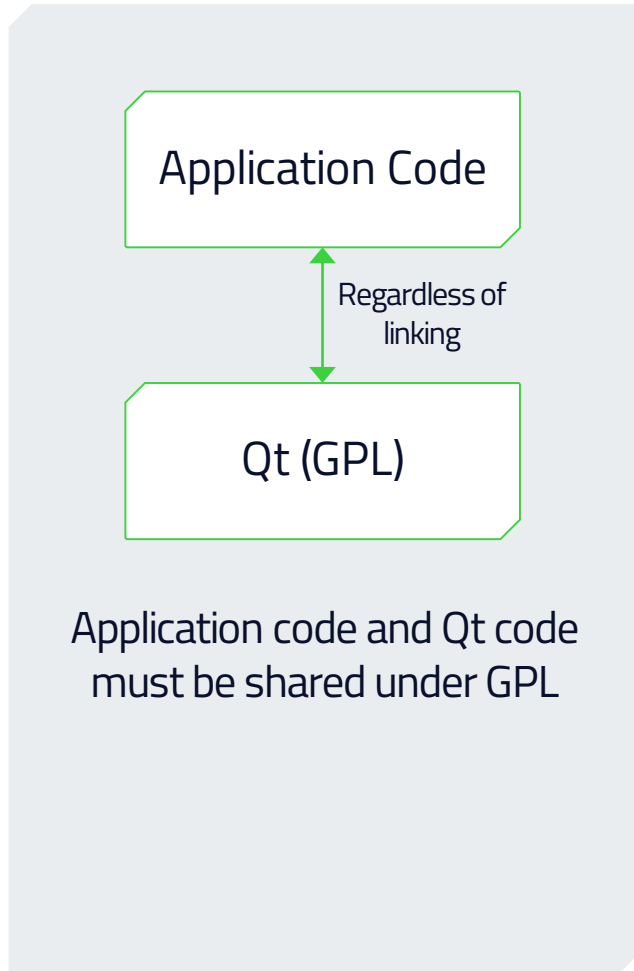
Protect IPR - keep the advantage on the competition

Warranty of the device

Etc.



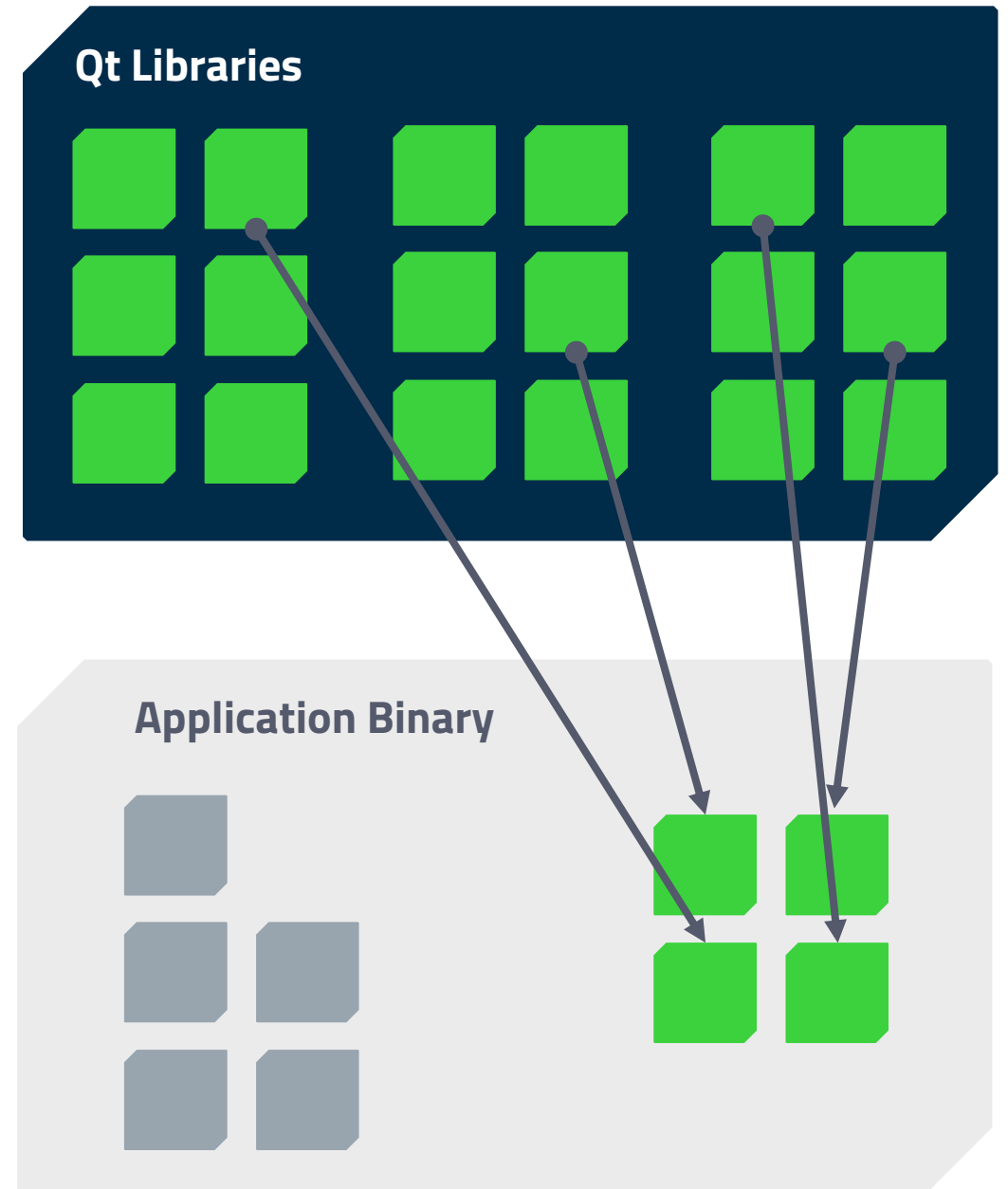
Linking libraries under different use-cases



Performance benefits

Static Linking

- › **Faster application loading time**
 - › Static linking reduces application loading time.
 - › Guarantees consistent loading time.
- › **Flexibility to design/customize HMI & Qt library**
 - › Best optimized memory foot-print
 - › Make Qt libraries compact on a function level.
- › **Performance**
 - › Compiles ONLY needed part of a library per application level.
 - › Single process architecture provides the most performance benefit.
- › **Security**
 - › Shared Lib can be hacked/replaced. Static Linking prevents this by including libraries into the binary file.
- › **Compatibility**
 - › Eases concerns of compatibility issues by pre-installed libraries.
- › **Easy publishing to the Apple App Store**



Technical Support

Working closely with R&D teams, we help developers address complex technical challenges

Standard Support*

- › Support on Qt APIs, functions, methods, and programming techniques
- › Software updates, patches and fixes for all releases
- › 48 business hours response time
- › Bug fix prioritization

*Support granted to Qt license holders

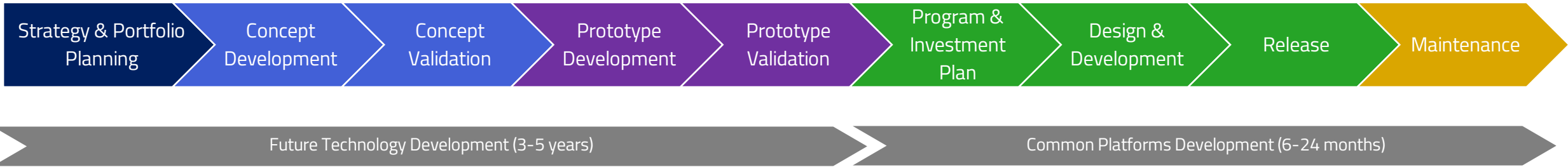
Premium Support

- › Dedicated support engineer
- › 24 business hours response time
- › Remote desktop assistance and troubleshooting
- › Support for your specific software solution
- › Direct online access

Extended Support

- › Enable long term maintenance and Qt upgrade opt-outs
- › Longer term support for Qt 5.x releases
- › Qt 4.x releases supported directly by the Services team

Professional Services



Strategic Decisions

- Needs:
- Strategy consulting

First technology decisions

- Needs:
- Technology Evaluation
 - Concept development services

Most important technology decisions

- Needs:
- Proof of Concept
 - Technology Migration

Productization and Optimization

- Needs:
- Architecture consulting
 - Design and Implementation
 - Platform standardization and optimization

Qt Commercial Offering



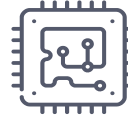
Qt for Application Development - QtAD

- › Licensing model: Individual developer licenses
- › Comparable to open source Qt
- › Windows, Linux, macOS, Android, iOS, Windows 10 (WinRT)



Other Products

- › **Qt for MCUs:** Qt Quick based offering for developing UI applications on Microcontrollers supporting FreeRTOS and BareMetal
- › **Qt Safe Renderer:** Safety critical GUI on separate hypervisor / virtual machine for *QtDC* or *Qt Automotive Suite*.
- › **Qt Automotive Suite:** Services, consulting and automotive-specific hardware support for *QtDC*
- › **Qt M2M Protocols:** M2M protocols for *QtAD* and *QtDC*
- › **Qt Design Studio:** Add on offering with QtAD and QtDC



Qt for Device Creation - QtDC

- › Licensing model: Individual developer licenses + distribution licenses
- › Commercial differentiation *on top* of QtAD
- › Improved productivity and performance for embedded development
- › Embedded Linux, Windows Embedded, INTEGRITY, QNX, VxWorks...



Your own SDK Based on Qt

- › Create SDKs based on Qt
- › For *QtAD* and *QtDC*
- › Licensed per agreement with The Qt Company



Virtual Tech Con 2020

Qt Open Source License

The Four Freedoms of the Free Software Movement

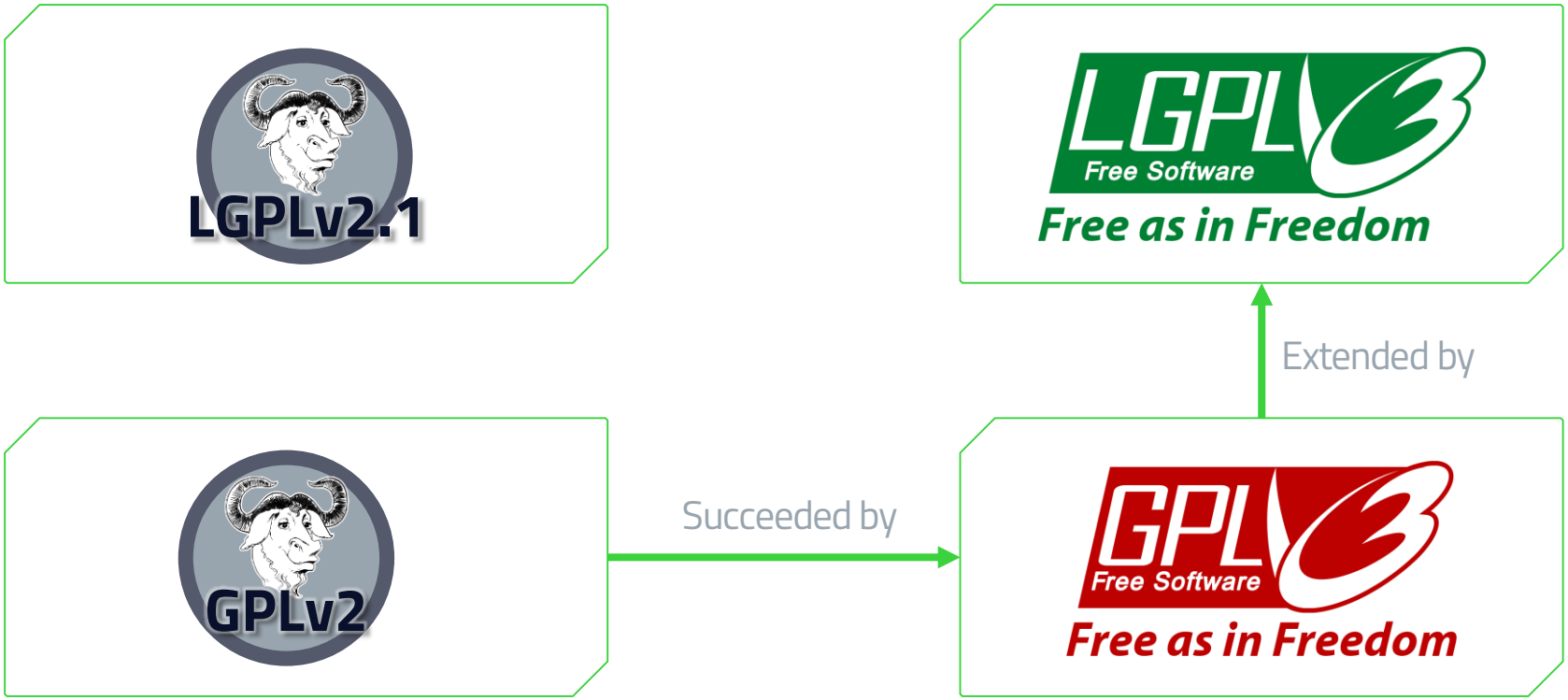
- 1) The freedom to run the program for any purpose
- 2) The freedom to study how the program works and adapt it to specific needs
- 3) The freedom to redistribute copies so you can help your neighbor
- 4) The freedom to improve the program and release your improvements to the public, so that the whole community benefits

> From the (L)GPL Preamble:

- > "To protect **your rights**, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to **certain responsibilities for you** if you distribute copies of the library or if you modify it."
- > "If you distribute copies of the library, whether gratis or for a fee, you must give the recipients **all the rights that we gave you.**"

- > When we speak of free software, we are referring to freedom, not price

GPL and LGPL Licenses



Important aspects of LGPL 1/4

No modifications to the license

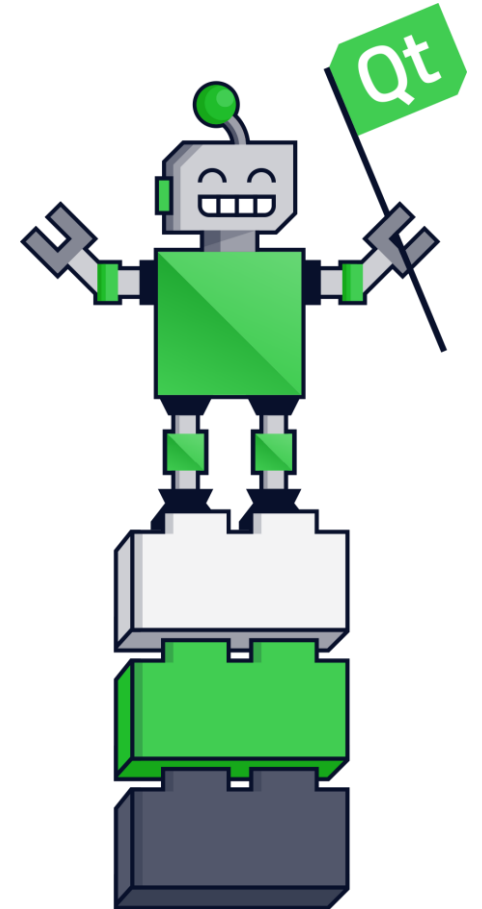
- › LGPL is written and copyrighted by the Free Software Foundation (FSF)
- › Only FSF can change and amend the standard

Housekeeping

- › Provide a notice stating that you are using the LGPL with each copy of the application.
- › Provide a copy of the full LGPL license text, i.e. GPL and LGPL.
- › If you are displaying any copyright notices, you also need to correctly display all copyrights and trademarks etc. associated with the open source software

Share any and all changes

- › You need to make sure that you share all modifications that you make to the LGPL licensed software (Qt)
- › Can this be a competitive advantage or disadvantage?
- › Will this be true for the whole scope of the development project?



Important aspects of LGPL 2/4

End user licensing

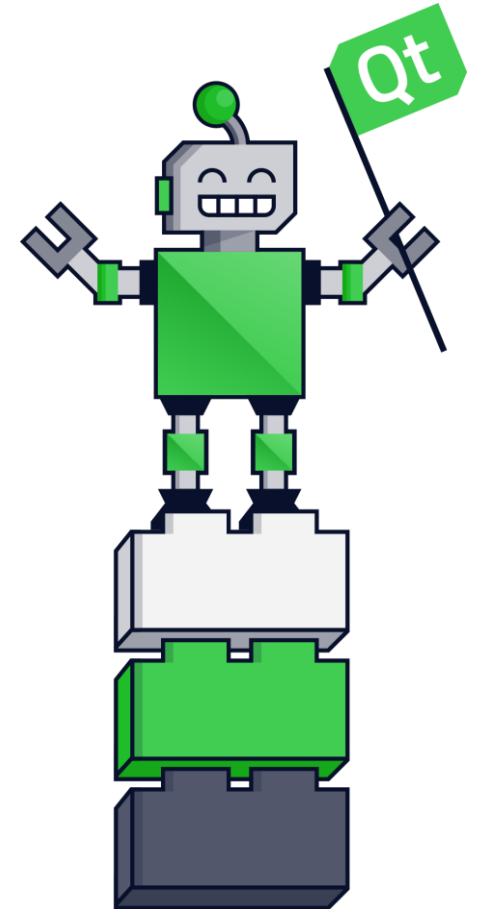
- › LGPL does not only affect you, but follow the product — you make this decision for all the recipients of your product.
- › Is this a decision you are comfortable making on behalf of all your customers and users?
- › In the Medical, Defense and Aerospace, and many other industries, they do not want to, or do not allow using LGPL-based code.

Dynamic vs. Static linking

- › Is there a requirement for static linking? Static linking is not OK under LGPLv3
- › The application needs to work properly even if relinked with a modified version of the library

Appropriate relinking method

- › The recipient of the LGPL license SW should be enabled to rebuild the SW.
- › Dynamically linking to Qt is not always enough.
- › Are you allowed to distribute all other libraries needed to relink the application?



Important aspects of LGPL 3/4

Compliance

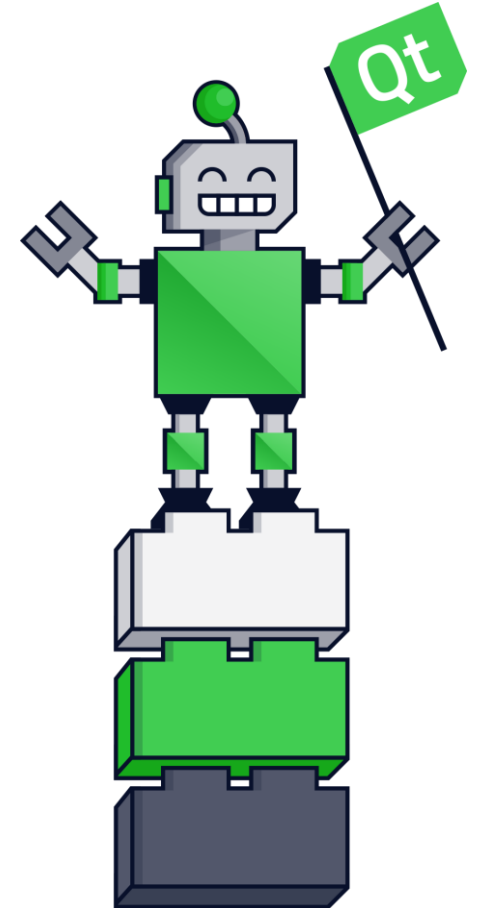
- › Terms in the LGPL license cannot be circumvented
- › All usage of the device must comply with the LGPL license
- › It is not allowed to circumvent the LGPL license by end user terms, or voiding the warranty if certain freedoms of the LGPL license are being executed.

License Inheritance

- › You inherit the LGPL license and all the obligations
- › Your customers will in turn inherit LGPL from you

Immutability

- › If you start out with commercial licenses, you can always go to LGPL later, or even ship certain devices under LGPL.
- › If you actively choose to use LGPL now, you cannot move to commercial later. Are you sure you are OK with that?



Important aspects of LGPL v3 4/4

The three main additions in version 3

“Tivoization”

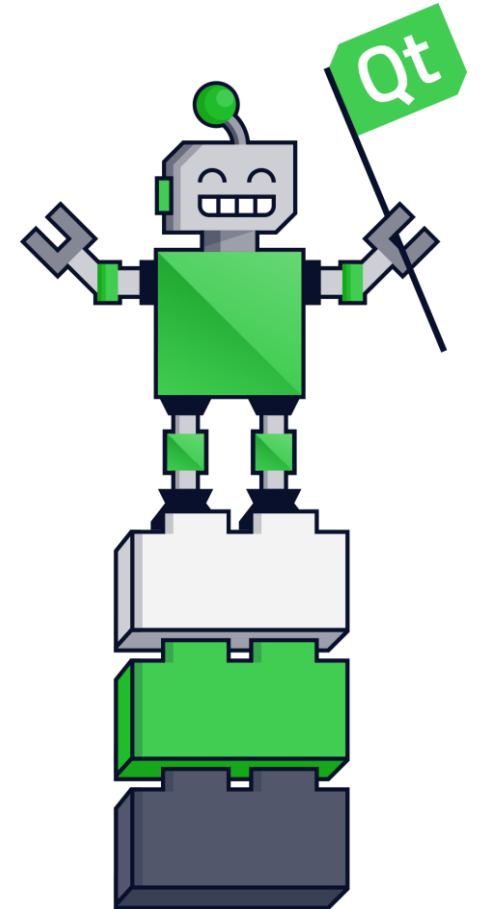
- › Definition = locking down devices as not to allow end users to change the software.
- › TiVo is a Set Top Box/DTV company that did this, and this clause is a result of their attempt to work around the LGPL v2.1 text.
- › LGPL v3 specifically prevents “Tivoization”, limited to User Products
 - › User Product is 1) a Consumer Product 2) anything designed or sold for incorporation into a dwelling
- › Includes providing any encryption keys or other techniques used in the device to limit tinkering and access

No Patent retaliation

- › Not possible to implement or enforce software patents when using an LGPL v3 library.
- › Section 10 prohibits people who uses the software from filing patent suits against other licensees.
- › This is a very important point for many – especially larger – corporations
- › A main reasons that LGPLv3 is a license that several major companies specifically do not allow to use internally.

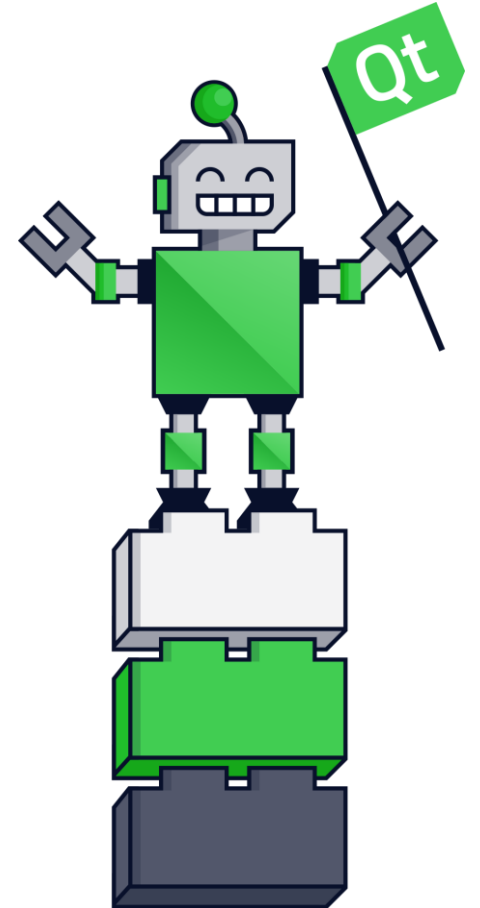
No DRM protection

- › Implementing DRM systems with LGPLv3 will not act as an effective technological "protection" measure.



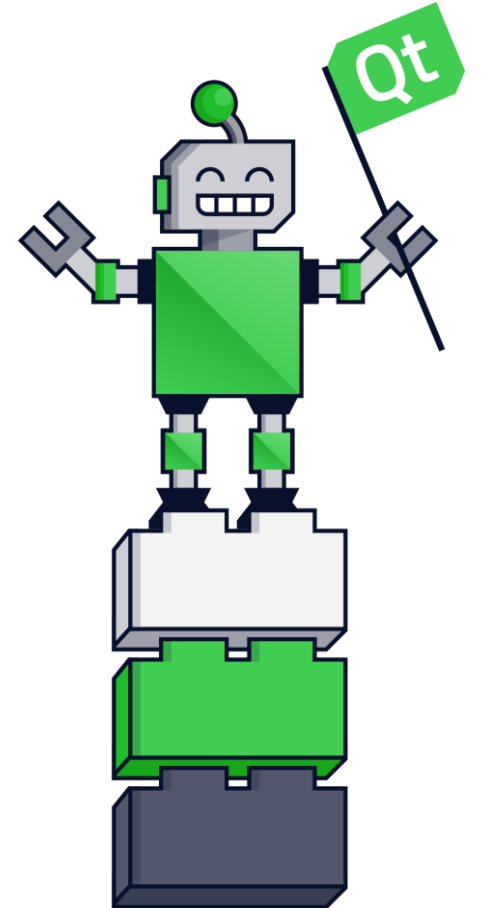
Significance of GPL in tools and applications

- › Tools with GPL license can be used to create applications with different licenses
 - › You can build a commercial application using GPL licenses tools
- › If you change or use the source code of the tool
 - › The resulting application/tool must have GPL license
 - › Source code must be available for all
- › GPL in a software module means
 - › That the resulting code must be under GPL
 - › The resulting source code must be available for all



Open source is not free of cost

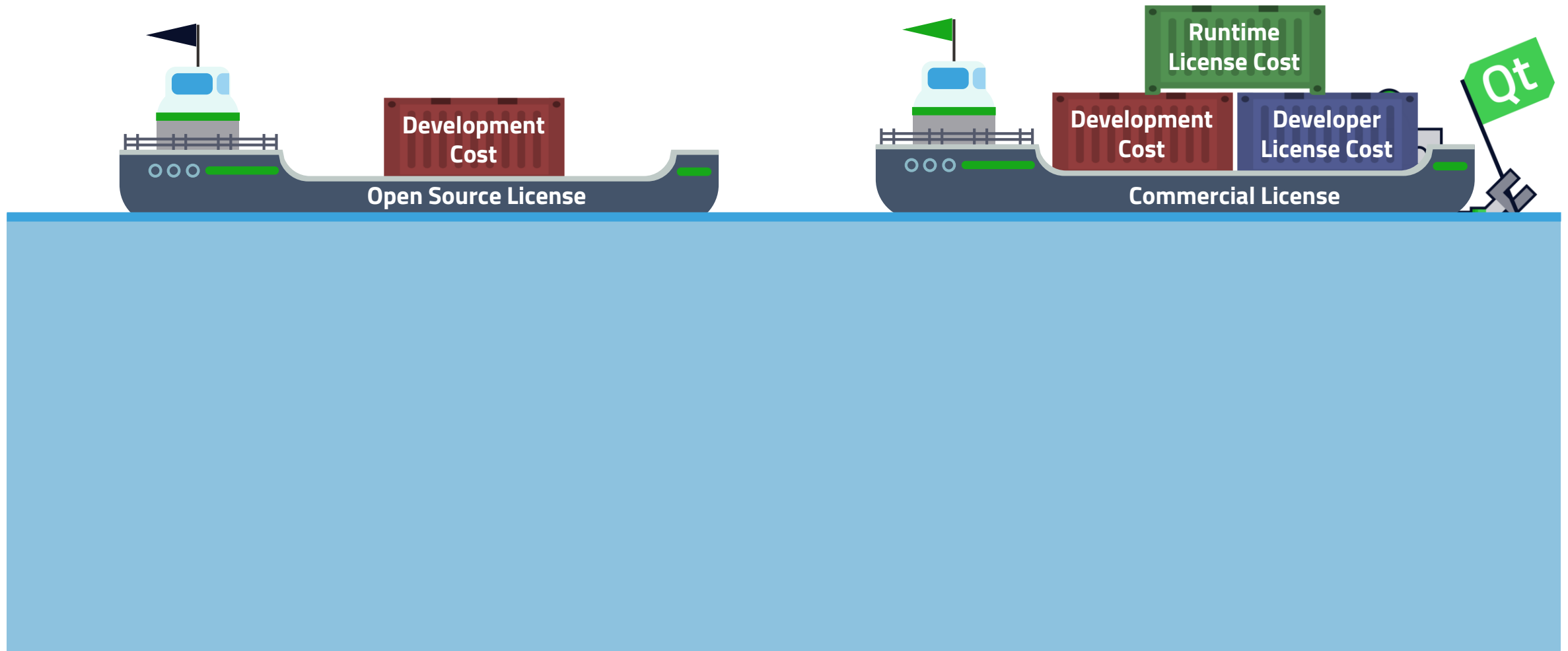
- › Managing and ensuring project compliancy with LGPL and GPL license requirements adds to the project costs:
- › Work on implementation, processes and legal practices in circumventing around license
- › Work on manual porting and building of things that are part of the commercial product; Relatively small cost for doing it first time, big cost on maintaining it over time
- › Possible warranty, patent, IPR and legal costs and risks for your company



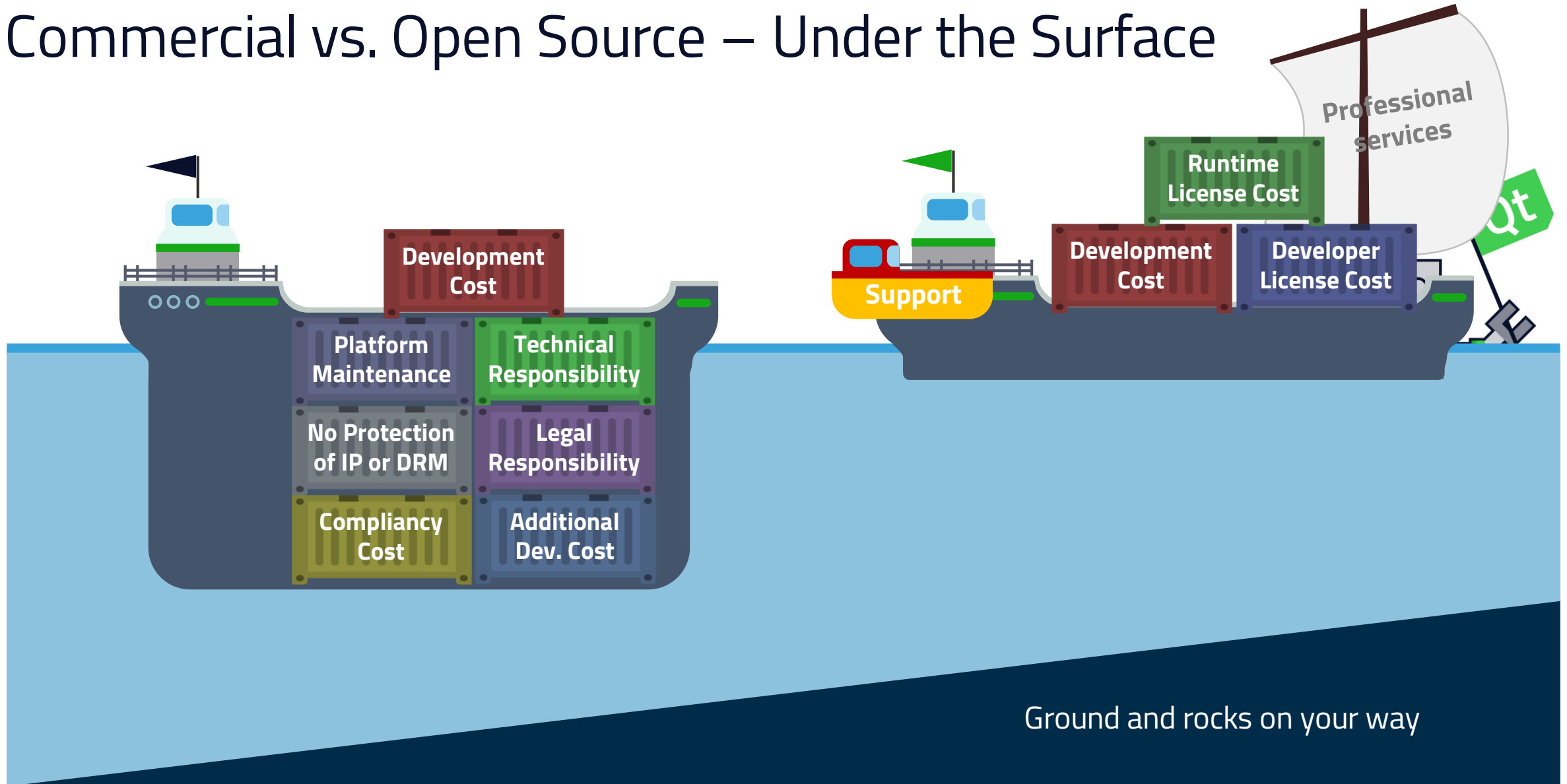


Open Source vs. Commercial

Commercial vs. Open Source – On the Surface



Commercial vs. Open Source – Under the Surface



Commercial vs. Open Source Comparison Table

	Open source offering	Commercial product
Price	Free of charge – but rarely a \$0 cost of ownership	Company contract
Community support	Yes	Yes
Qt support helpdesk	Not available	Yes
Access to commercial only offerings, features, benefits, previews etc.	No	Yes
Possibility to keep application private with static linking	No	Yes
Possibility to keep application private with dynamic linking	LGPLv3 only (Qt 5.7+)	Yes
Enable customers to relink the Qt libraries (also with static linking)	Yes, mandated	Optional, up to you
Provide copy of the license and explicitly acknowledge the use of Qt	Yes, mandated	Optional, up to you
Make a copy of the Qt source code available for your customers	Yes, mandated if requested	Optional, up to you
All rights to make and keep Qt source code modifications proprietary	No	Yes
Usage of DRM	See LGPLv3 limitations	No limitations
Possibility to create or include patented software	Strong limitations on patents	No limitations

Qt for Application Development, Commercial

Add-Ons

		Qt Mac Extras	Qt Quick Controls	Qt SVG	Qt X11 Extras
Active Qt	Qt D-Bus	Qt Network Authorization	Qt Quick Extras	Qt UI Tools	Qt XML & XML Patterns
Qt3D	Qt Gamepad	Qt NFC	Qt Quick Widgets	Qt WebChannel	Qt Wayland Compositor
Qt Android Extras	Qt Graphical Effects	Qt Platform Headers	Qt SCXML	Qt WebEngine	Qt Charts
Qt Bluetooth	Qt Help	Qt Positioning	Qt Sensors	Qt WebSockets	Qt Data Visualization
Qt Canvas 3D	Qt Image Formats	Qt Print Support	Qt Serial Bus & Serial Port	Qt WebView	Qt Virtual Keyboard
Qt Concurrent	Qt Location	Qt Purchasing	Qt Speech	Qt Windows Extras	Qt 3D Studio Runtime

Essentials

				Qt Quick Dialogs	Qt SQL
Qt Core	Qt Multimedia	Qt Network	Qt Quick	Qt Quick Layouts	Qt Test
Qt GUI	Qt Multimedia Widgets	Qt QML	Qt Quick Controls 2	Qt Quick Test	Qt Widgets

Desktop & Mobile Platforms

Windows	Mac	Linux Desktop	Android	iOS	WinRT
---------	-----	---------------	---------	-----	-------

Development Tools

Qt Creator Cross-platform IDE	CPU Usage Analyzer
Qt Designer GUI Designer	GPU Profiler
Qt Linguist I18N Toolset	Clang Static Analyzer
Qt Assistant Documentation Tool	Qt Quick Compiler
moc, uic, rcc Build Tools	Qt Quick Profiler
qmake Cross-platform Build Tool	Autotest integration

LGPLv2.1 +
 GPL2

Commercial

Qt for Application Development, Open Source

Add-Ons					
		Qt Mac Extras	Qt Quick Controls	Qt SVG	Qt X11 Extras
Active Qt	Qt D-Bus	Qt Network Authorization	Qt Quick Extras	Qt UI Tools	Qt XML & XML Patterns
Qt3D	Qt Gamepad	Qt NFC	Qt Quick Widgets	Qt WebChannel	Qt Wayland Compositor
Qt Android Extras	Qt Graphical Effects	Qt Platform Headers	Qt SCXML	Qt WebEngine	Qt Charts
Qt Bluetooth	Qt Help	Qt Positioning	Qt Sensors	Qt WebSockets	Qt Data Visualization
Qt Canvas 3D	Qt Image Formats	Qt Print Support	Qt Serial Bus & Serial Port	Qt WebView	Qt Virtual Keyboard
Qt Concurrent	Qt Location	Qt Purchasing	Qt Speech	Qt Windows Extras	Qt 3D Studio Runtime

Essentials					
				Qt Quick Dialogs	Qt SQL
Qt Core	Qt Multimedia	Qt Network	Qt Quick	Qt Quick Layouts	Qt Test
Qt GUI	Qt Multimedia Widgets	Qt QML	Qt Quick Controls 2	Qt Quick Test	Qt Widgets

Desktop & Mobile Platforms					
Windows	Mac	Linux Desktop	Android	iOS	WinRT

Development Tools

Qt Creator Cross-platform IDE	CPU Usage Analyzer
Qt Designer GUI Designer	GPU Profiler
Qt Linguist I18N Toolset	Clang Static Analyzer
Qt Assistant Documentation Tool	Qt Quick Compiler
moc, uic, rcc Build Tools	Qt Quick Profiler
qmake Cross-platform Build Tool	Autotest integration

LGPLv2.1 +
 GPL2

LGPLv3 + GPLv3
 + GPLv2

GPLv3

Commercial
 Only

Qt for Device Creation, Commercial

Embedded solutions

Device Utilities

Qt Quick 2D Renderer

Software Stack (eLinux only)

Yocto based Boot to Qt stack(s)

Build your own stack

Embedded tooling

Cross compilation toolchains

Remote deployment and debugging over USB ...and over network

Remote profiling over USB ...and over network

LGPLv2.1 +
 GPL2

Commercial

GPLvX (varies)

Qt for Application Development

Add-Ons

Qt WebEngine

Essentials

Desktop & Mobile Platforms

Development Tools

Additional target platforms

eLinux

QNX

INTEGRITY

VxWorks

Qt for Device Creation, Open Source

Embedded solutions

Device Utilities

Qt Quick 2D Renderer

Software Stack (eLinux only)

Yocto based Boot to Qt stack(s)

Build your own stack

Embedded tooling

Cross compilation toolchains

Remote deployment and debugging over USB ...and over network

Remote profiling over USB ...and over network

LGPLv2.1 + GPL2

LGPLv3 + GPLv3 + GPLv2

GPLv3

Commercial Only

Qt for Application Development

Add-Ons



Essentials

Desktop & Mobile Platforms

Development Tools



Additional target platforms

eLinux

QNX

INTEGRITY

VxWorks

Other Products, Commercial

Designer Tools

Qt 3D Studio

Qt Design Studio

Modules

Qt Safe Renderer

Qt Application Manager

Industry specific packages

Qt Automotive Suite

Qt M2M Protocols

LGPLv2.1 +
 GPL2

LGPLv3 + GPLv3
 + GPLv2

GPLv3

Commercial
 Only

New Offerings

Qt for MCUs

Other Products, Open Source

Designer Tools

Qt 3D Studio

Qt Design Studio

Modules

Qt Safe Renderer

Qt Application Manager

Industry specific packages

Qt Automotive Suite

Qt M2M Protocols

LGPLv2.1 +
GPL2

LGPLv3 + GPLv3
+ GPLv2

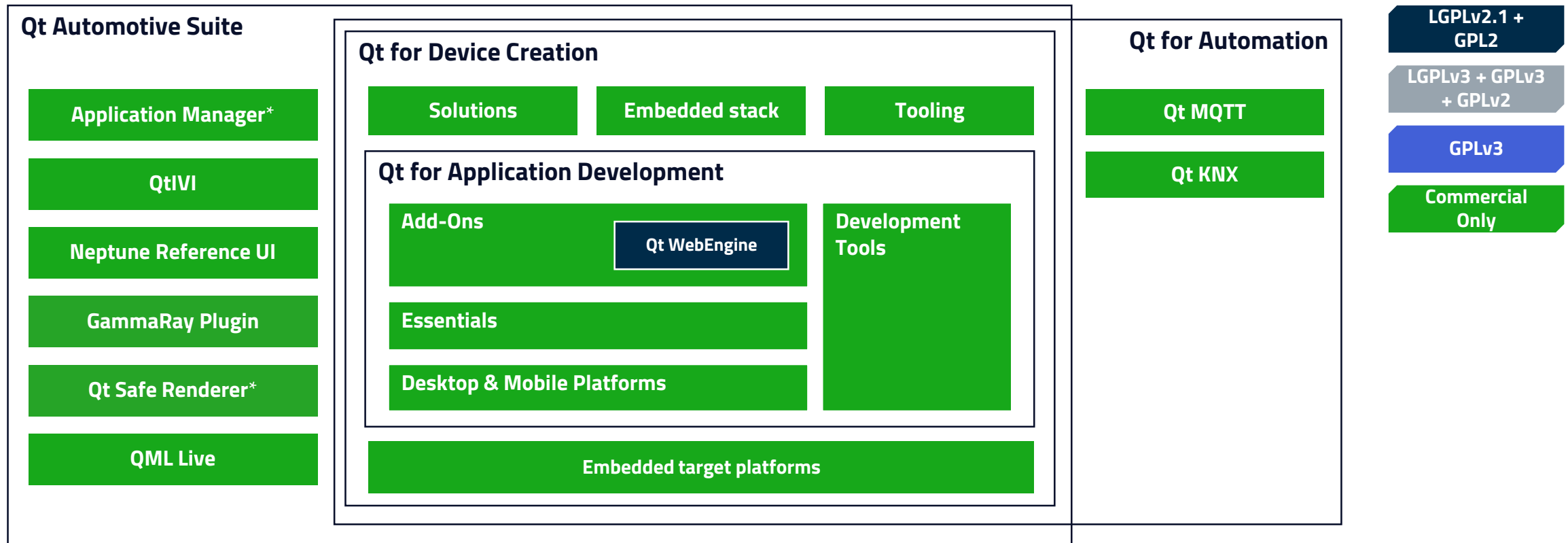
GPLv3

Commercial
Only

New Offerings

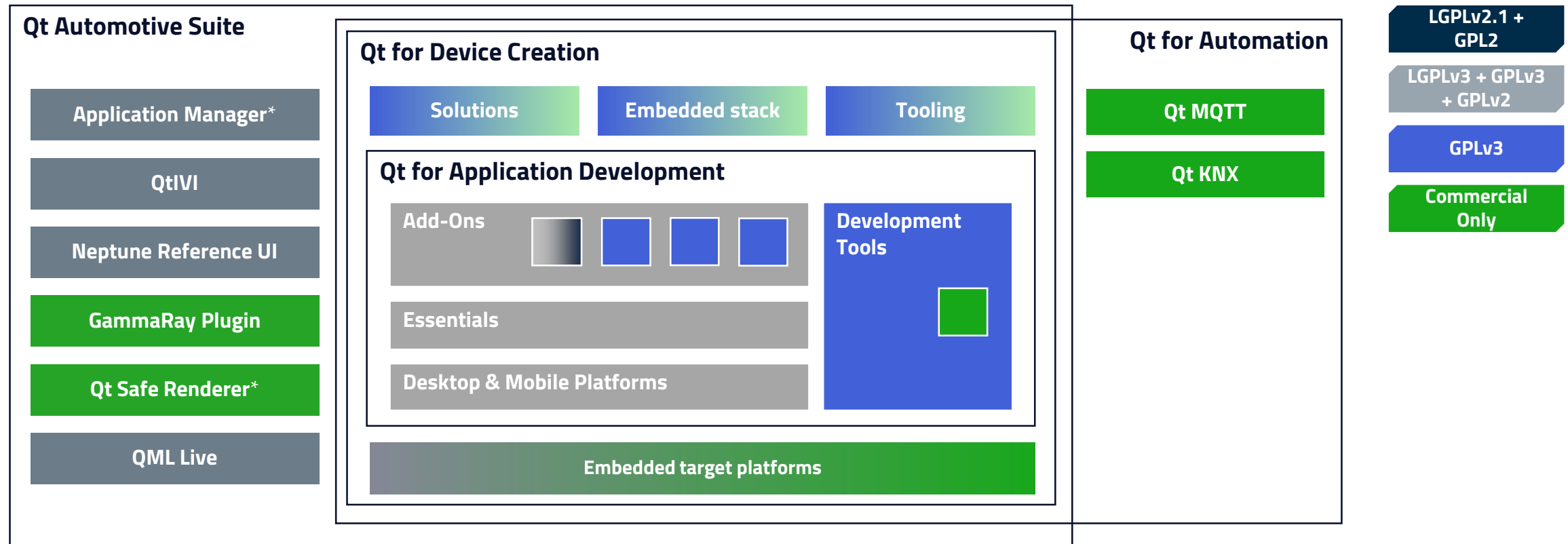
Qt for MCUs

Qt Automotive Suite and Qt M2M Protocols, Commercial



*The Qt Safe Renderer and Qt Application Manager can also be sold separately with Qt for Device Creation

Qt Automotive Suite and Qt M2M Protocols, Open Source



*The Qt Safe Renderer and Qt Application Manager can also be sold separately with Qt for Device Creation



Virtual Tech Con 2020

Qt Product Licensing

Important Aspects of the Qt Commercial license

Developer licensing

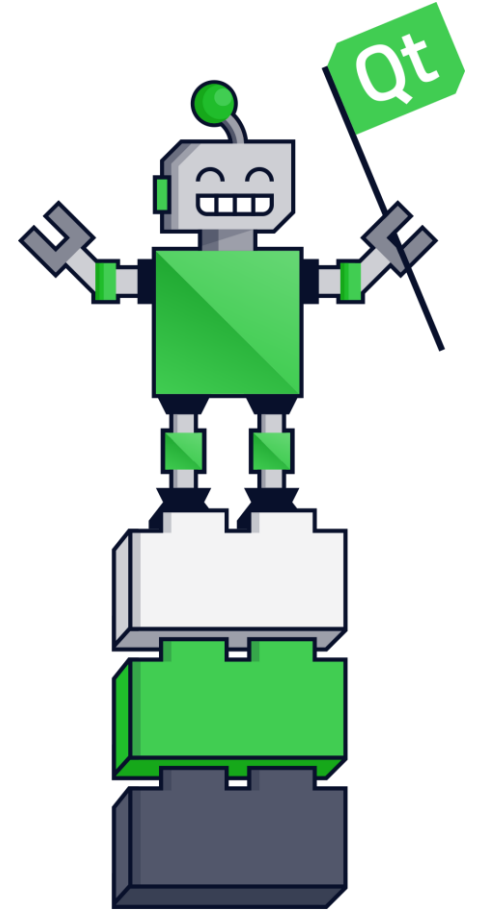
- › No mixing of commercial and (L)GPL Qt usage!
- › Make sure to clarify your licensing choice before you start developing!
- › What are your distribution channels, now and in the future?
- › How critical is your time-to-market?

Distribution licensing

- › For “Joint software and hardware products”
- › Protects the device, the source code and the intellectual property you have developed
- › Indemnifications and liability
- › Compliant with patent clauses and third party licenses

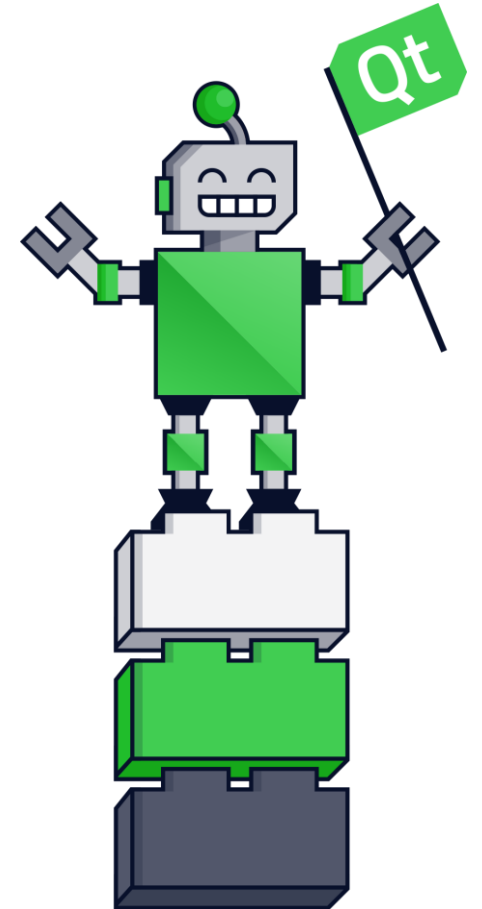
Developer License

- › Access to LTS versions, standard support and product updates during entire term
- › License to use the product for the given term, usually between 1 and 3 years
- › Can't continue use of Qt, nor distribute software, once the term is ended
- › Purchased through The Qt Company Sales
- › Can be bought online from <http://www.qt.io>



Distribution License

- › Small fee paid per device benefitting from Qt usage
 - › Required for distribution of devices (combined hardware and software)
- › Unit price depends on factors such as:
 - › Distribution volume and term length for distribution
 - › Use of additional add-on features (e.g. Qt for Automation)
- › Can't continue use of Qt, nor distribute devices, once the term is ended
 - › Exception: Can keep using purchased distribution licenses for 6 months
- › Purchased through The Qt Company Sales



Thank you!

- › Learn more at www.resources.qt.io
- › Contact us at www.qt.io/contact-us/
- › Try for Free at www.qt.io/download/

Qt World Summit 2020, Palm Springs, October 20-22

www.qtworldsummit.com

Follow us:

Twitter: @Qtproject

Facebook: /qt/

Linkedin: /company/theqtcompany/

Youtube: /Qtstudios/

