Embedded linux is a glorious *lie*

Aleix Pol i Gonzàlez <aleixpol@kde.org>

- 🐧 Aleix
- 🐧 From Barcelona, living in Berlin
- 🐧 Been doing KDE things for ages
- 🐧 Employed by MBition GmbH, putting KDE and Wayland in Mercedes cars
- 🐧 KDE e.V. President, at your service

What do we mean by Embedded linux?

> Embedded Linux is built on **the same Linux kernel**, available from kernel.org, as all Linux systems. But embedded systems have tight constraints that enterprise systems simply don't have, ranging from higher **reliability** and **security** requirements to **tighter resource availability** and the need for engineering **support** that often lasts 10 years or more.

Source: windriver.com

Does KDE not care about…

…security?

…resources?

…reliability

…availability?

Why do I call it a *lie*?

How does it differ from any other linux?

A useful *lie*…

… to create a system that *just* works

… to allow the licence to fork and adapt software

Record scratch

Where do we come from in KDE and FOSS?

Are we complicating things?

Are we shifting the responsibility to the user?

What does KDE know about how our software is to be used?

Breaking this *lie* down:

- 🐧 How can we better serve industry partners?
- 🐧 What can we learn from it?
- 🐧 Which strategies will better serve us?

They'll patch your code until it fulfills their requirements

👎

Create and document strategies to extend without modifying (patching/forking)

👍

Encourage contributing changes upstream

👍

They'll be using old (and eventually buggy) versions of the software

👎

Be mindful about compatibility, document porting tasks when possible

👍

They'll be unfamiliar with our ways

… making something simpler is always a good idea

👍

Allowing cross-process interaction helps to simplify

👍

Organisations often opt for FOSS solutions when they aren’t their main focus or priority

👍👎

Having other organisations base their work on our products helps us stay stable

👍

They will likely be testing their own things, making it easier to test helps make that happen

👍

But hey ❗

We are not here to serve you!!

But we are not that different...

… we too struggle with embracing varied form factors

... we too need to extend our functionalities

… we too need support complex features over time

We do want the world to build upon our standards, tech and solutions

Where does that leave us?

Let's reconsider which levels we are abstracting for

Are we confident about our security model?

- 🐧 What is an app?
- 🐧 Is the OS still the main software provider to our users?
- 🐧 Are our sandboxing models ready to export as a solution to embedded vendors?

Is ABI as important as it used to be?

Take-aways, for others to use our stuff

- 🐧 When our software is easy to consume, we all win
- 🐧 Collaboration makes us all stronger
- 🐧 Abstract only what's useful to abstract

Take-aways, for our products

- 🐧 We should sit on tools that allow us to create the product we want
- 🐧 We probably need to have a plan for when upstream needs addressing
- 🐧 We should sit on tech that allows us to embrace different form factors and architectures

Was then Plasma an embedded product all along?

🌸

?

Aleix Pol i Gonzàlez <aleixpol@kde.org>