



**Embedded Linux
Conference**
Europe

Graphical User Interface Using Flutter in Embedded Systems

Hidenori Matsubayashi
@ Sony Corporation

#felc



SONY

Graphical User Interface Using Flutter in Embedded Systems

Embedded Linux Conference Europe
October 27, 2020

Hidenori Matsubayashi
@ Sony Corporation

Sec.2 Tokyo Laboratory 29 R&D Center

Hidenori Matsubayashi

Embedded Software Engineer @ Sony

- Living in Tokyo, Japan
- Specialties
 - Embedded Linux System Development
 - Board bring-up and software integration for:
 - Qualcomm
 - NXP
 - NVIDIA
 - Middleware
 - Graphics
 - Audio/Video
 - Firmware/Low level layer
 - RTOS
 - Bootloader
 - Device driver
 - FPGA
 - Programming Languages: C/C++, Rust, Dart



Hidenori.Matsubayashi@sony.com



[HidenoriMatsubayashi](https://github.com/HidenoriMatsubayashi)

Agenda

- Background
- Overview of the new approach
- Demo Video
- Details on our approach
- Summary

Agenda

- Background
- Overview of the new approach
- Demo Video
- Details on our approach
- Summary

Background

- We were always searching suitable GUI toolkits for embedded systems
 - OSS ?
 - Commercial software ?
 - Toolkits which is provided by SoC vendor ?
- There are a lot of GUI toolkits in OSS or commercial licenses.
- However, there aren't a lot of GUI toolkits available for especially **consumer embedded products** in OSS.

Why few toolkits for consumer embedded products in OSS?

There are two main reasons

1. Our requirements
2. Typical challenges on Embedded platforms

Reason 1: Main Requirements for GUI toolkits

- **High designability**
 - Need beautiful UI and smooth animation like smartphone and web (not like desktop apps)
- **Ease of development**
 - Sufficient development environment
 - There are more information about them on the internet
- **Good Performance / Footprint**
 - Low CPU usage
 - Low memory usage
- **Software Portability**
 - Support display servers (X11 / Wayland)
 - Support cross-platforms and link an embedded products and smartphone apps
- **Software License**
 - Sometimes, need proprietary software

Examples of GUI toolkits (Exclude toolkits that only focused Android, iOS and desktops)

Category	Name	Software License	Main Maintainer
Web-based	Electron	MIT License	GitHub
	NW.js	MIT License	Intel
	Chromium	BSD 3-Clause	Google
	WebKit	LGPL, BSD	Apple
	Gecko	Mozilla Public License 2.0	Mozilla
Desktop-based	GTK	LGPL v2.1+	GNOME
	Qt	Commercial License (or GPL/LGPL v3.0)	Qt Company
	Mono	MIT, BSD, GPL etc.	Microsoft (Xamarin)
	SDL	zlib License	- (OSS Community)
	Kivy	MIT License	- (OSS Community)
	wxWidgets	wxWindows License	- (OSS Community)
	openFrameworks	MIT License	- (OSS Community)
Mobile-based	Flutter	BSD 3-Clause	Google
Game-based	Unreal Engine	Commercial License (depends on sales)	Epic Games
	Unity	Commercial License (depends on sales)	Unity

often used

often used

our new approach

About WebView (WebKit / Chromium)

- HTML5/JavaScript using WebView (embedded browser rendering engine) or browsers is often used in embedded products
- Pros
 - Beautiful UI
 - Easy to develop (include development environments)
 - High portability (HTML/JavaScript resources), etc.
- Cons
 - High introduction cost
 - Huge source code and dependent packages (libraries)
 - Need high performance and footprint
 - Difficulty to access local files or Hardware resources
 - We usually use an internal web server to resolve it

About native toolkits

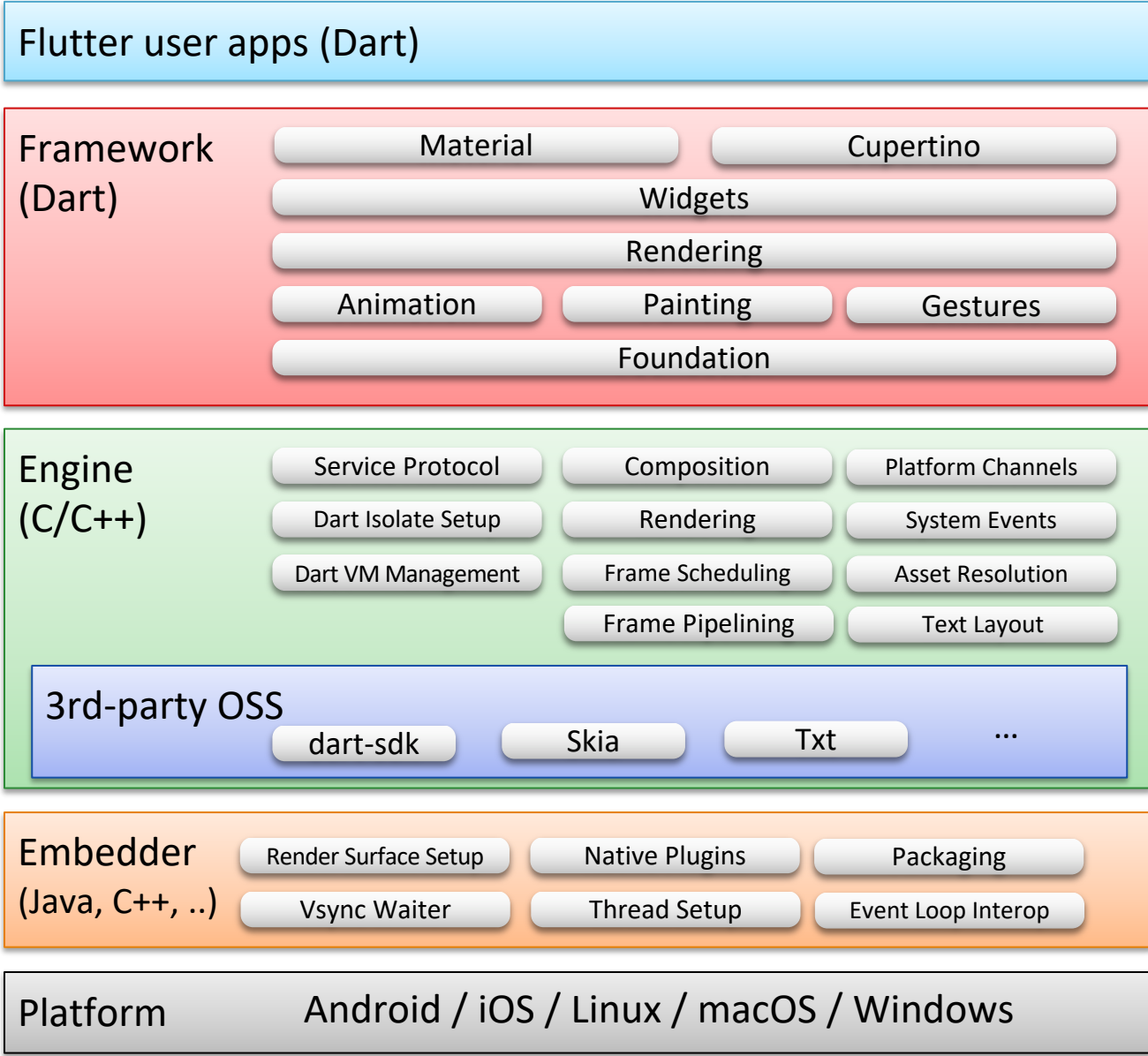
- GTK
 - <https://www.gtk.org/>
 - Open-source cross-platform toolkit
 - Desktop (Linux / macOS / windows)
 - Standard GUI toolkit on Linux
 - Not a modern design just like smartphone apps because it is for desktop apps
- Qt
 - <https://www.qt.io/>
 - Very popular
 - Open-source cross-platform toolkit
 - Desktop (Linux / macOS / windows), mobile, Embedded (Linux / RTOS), RTOS etc.
 - Dual-licensed under commercial and open source licenses (GPL/LGPL v3.0)
- SDL
 - <https://www.libsdl.org/>
 - Open-source cross-platform development library without widgets
 - Suitable for games or apps that have only simple menus

What is Flutter?



- <https://flutter.dev/>
- Flutter is Google's GUI toolkit for building beautiful, natively compiled applications
- Released in 2017
- Supported platforms from a single source code
 - Mobile (Android/iOS)
 - Web: β version
 - Desktop (Linux/Windows/macOS): α version
- Programming language: Dart

Flutter architecture overview



- Written in Dart language
- Create UI by using widgets
- Works on Dart-VM

- Graphics Engine by being written in C/C++
- Provide graphics shell
- Manage Dart-VM
- Drawing by using Skia etc.

Porting layer for specific platforms

Why few toolkits for consumer embedded products in OSS?

There are two main reasons

1. Our requirements

2. Typical challenges on Embedded platforms

Reason 2: Typical challenges on Embedded platforms

- Development is usually done on different architectures
 - Need to support multi architecture and SoCs
- Limited CPU performance / memory
 - Lower cpu-usage is better
- Display server types (X11 or Wayland) that are supported by BSP
 - Especially, Wayland is mainly supported by SoC vendors



We need GUI toolkits that are as lightweight as possible and independent of hardware and architectures

BSP trends of supporting display server

- ... Official Support
- △ ... Unofficial (3rd party)
- × ... Not supported

Vendor	SoC/Board/ Platform	Display server		Notes
		X11	Wayland	
NXP	i.MX 8M	×	○	X11 isn't supported after i.MX 6
Xilinx	Zynq	○	○	Peta Linux supports Wayland from 2019.2
NVIDIA	Jetson	○	○	-
Qualcomm	RB5	○	○	SoC: SD865
Raspberry Pi4	Broadcom BCM2711	○	△	-

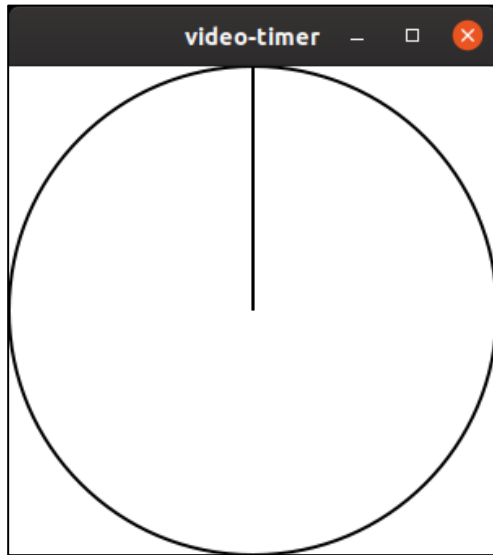
Gradually shifting?

Wayland is suitable for embedded systems because it is lightweight and less dependency packages.

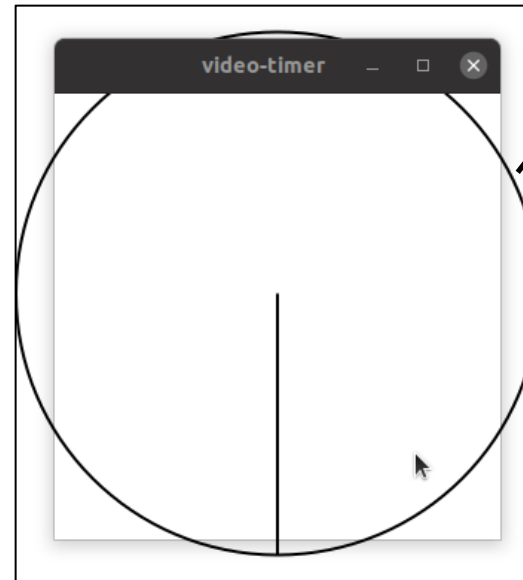
Challenges by using Wayland

- Unfortunately, a lot of GUI toolkits aren't enough support for Wayland
 - Because they are based on X11 and still developing now
 - e.g. Some GTK APIs aren't still supported on Wayland

GTK example on X11



GTK example on Wayland



Drawing size
isn't incorrect

Wrapping up of our purposes

We were looking for a toolkit that can solve the issues mentioned so far.

- A new GUI toolkit has high functionality and designability like WebView
- Suitable for Wayland and Embedded systems
- Lightweight, good performance
- Low costs
- Support cross-platform
- etc.

Agenda

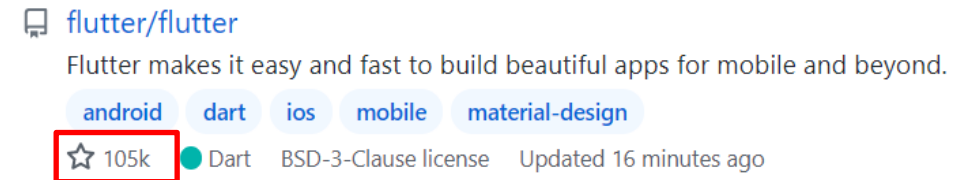
- Background
- Overview of the new approach
- Demo Video
- Details on our approach
- Summary

New approach to GUI in embedded systems

- Flutter + Wayland is the best practice in embedded systems using Linux
- Why Flutter?
- Why is the combination of Flutter and Wayland good?

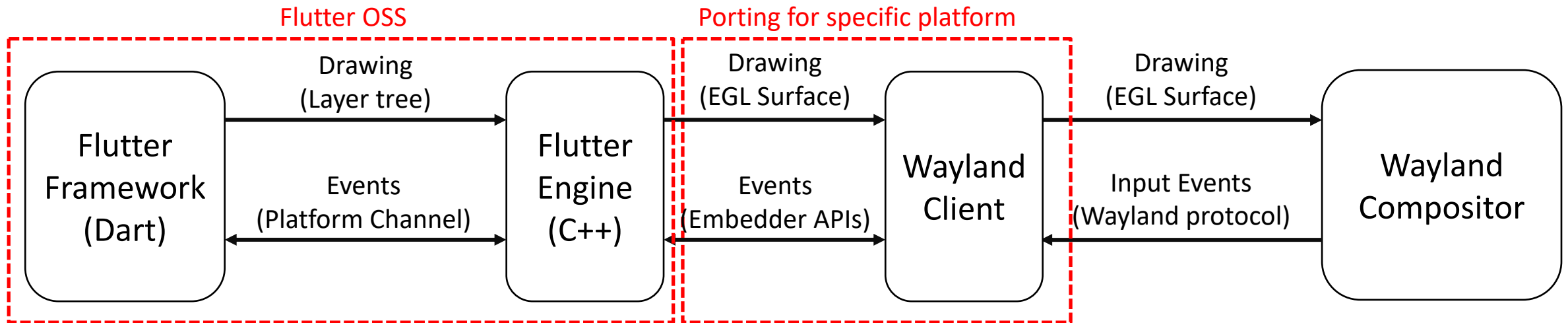
Why we choose Flutter?

- You can easily create a modern UI like a mobile app
- Supporting cross-platform (Desktop, Mobile, Web)
- Flutter is popular OSS and there are a lot of information
- Flutter is natively compiled applications (Fast!)
- Flutter provides the custom embedder API-layer for specific platforms
 - <https://github.com/flutter/flutter/wiki/Custom-Flutter-Engine-Embedders>
- Fewer library dependencies (Flutter Engine)
 - Basically your platform needs only OpenGL/EGL library
- Software license is BSD 3-Clause



Why is the combination of Flutter and Wayland good?

- Flutter draws graphics by using OpenGL or Vulkan
 - Flutter draws directly UI to EGL Surface
- Wayland (Client)
 - Provides EGL Surface to Flutter
 - Sends Keyboard / Mouse input event to Flutter
 - Communication with Flutter (Dart)
 - ...



Flutter embedder APIs example

➤ embedder header file

- <https://github.com/flutter/engine/blob/master/shell/platform/embedder/embedder.h>

```
FLUTTER_EXPORT
FlutterEngineResult FlutterEngineRun(size_t version,
                                     const FlutterRendererConfig* config,
                                     const FlutterProjectArgs* args,
                                     void* user_data,
                                     FLUTTER_API_SYMBOL(FlutterEngine) *
                                     engine_out);
```

Flutter run

```
FLUTTER_EXPORT
FlutterEngineResult FlutterEngineSendWindowMetricsEvent(
    FLUTTER_API_SYMBOL(FlutterEngine) engine,
    const FlutterWindowMetricsEvent* event);
```

Set Window Size

```
FLUTTER_EXPORT
FlutterEngineResult FlutterEngineSendPointerEvent(
    FLUTTER_API_SYMBOL(FlutterEngine) engine,
    const FlutterPointerEvent* events,
    size_t events_count);
```

Send Input Events

Agenda

- Background
- Overview of the new approach
- **Demo Video**
- Details on our approach
- Summary

About demo video

- We prepared the demo app for ELC Europe 2020
- The System UI prototype for demo

Watch the demo video



[References] Performance / Footprint

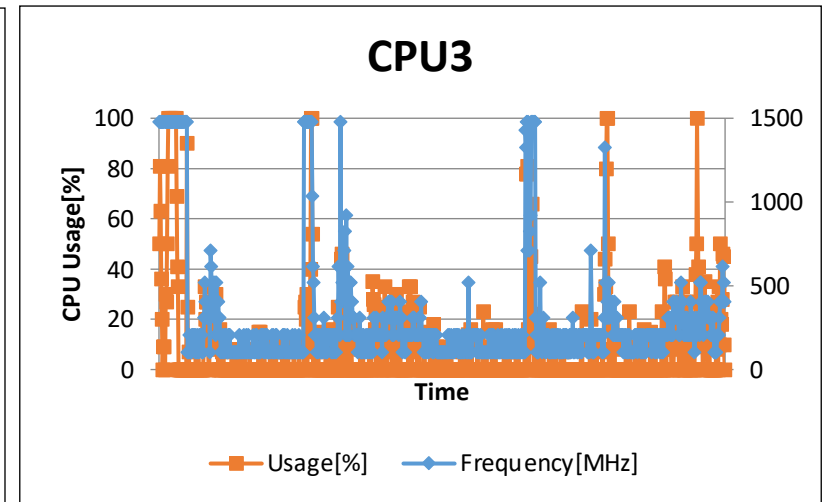
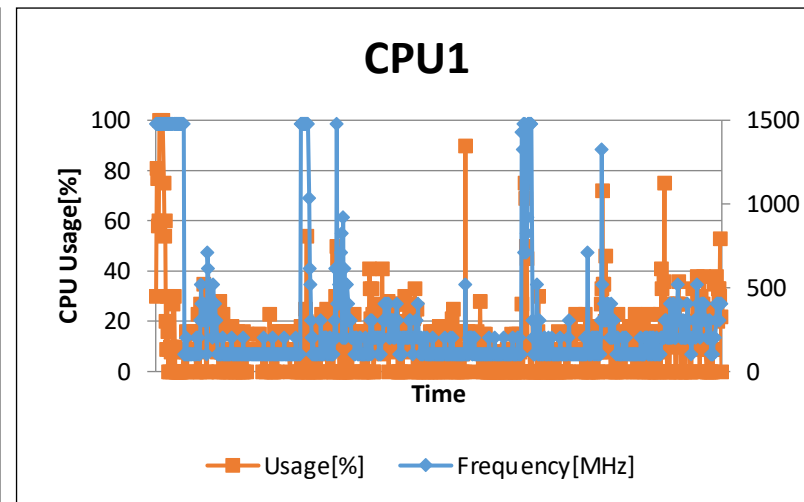
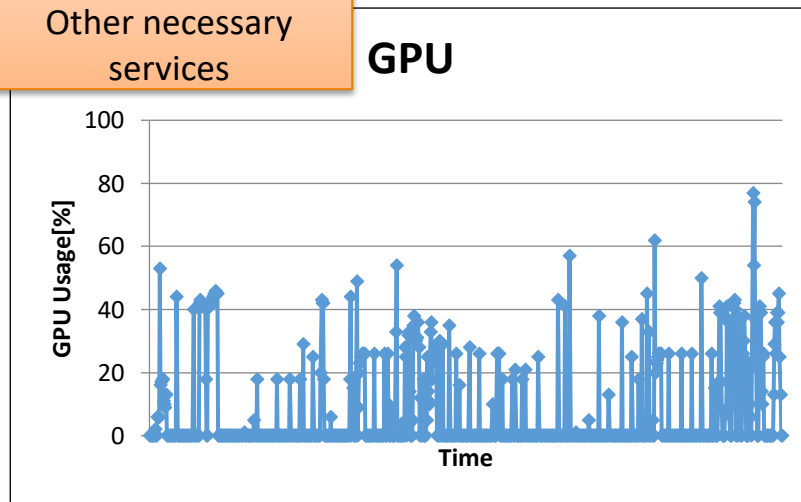
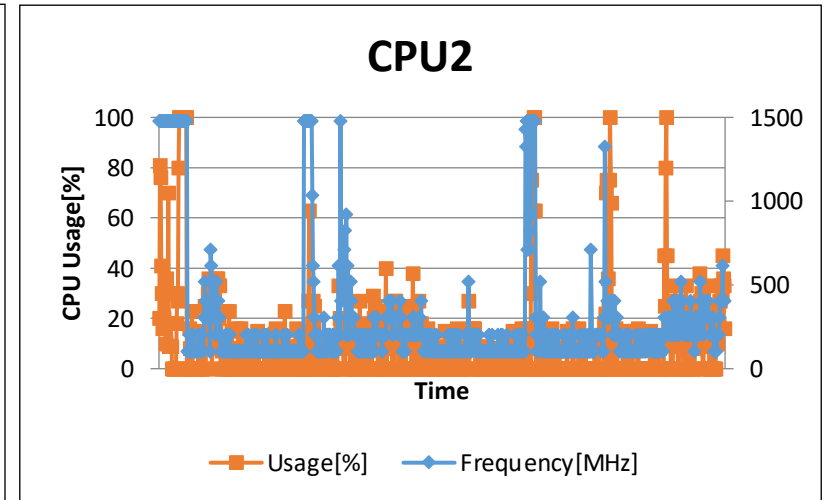
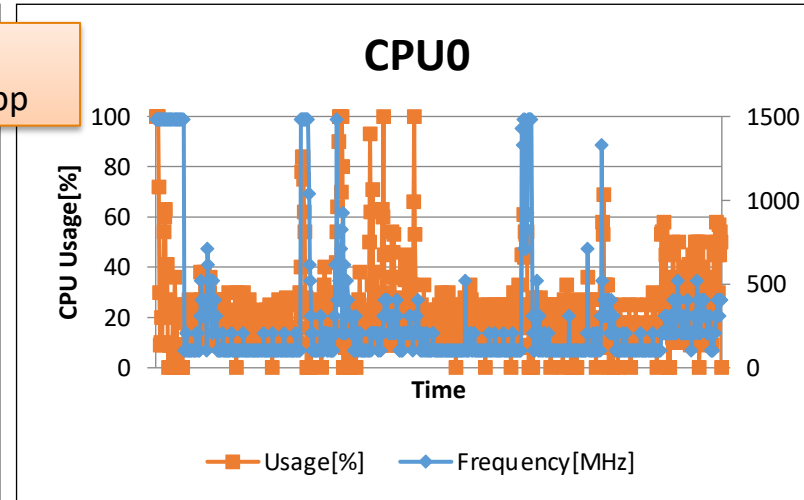
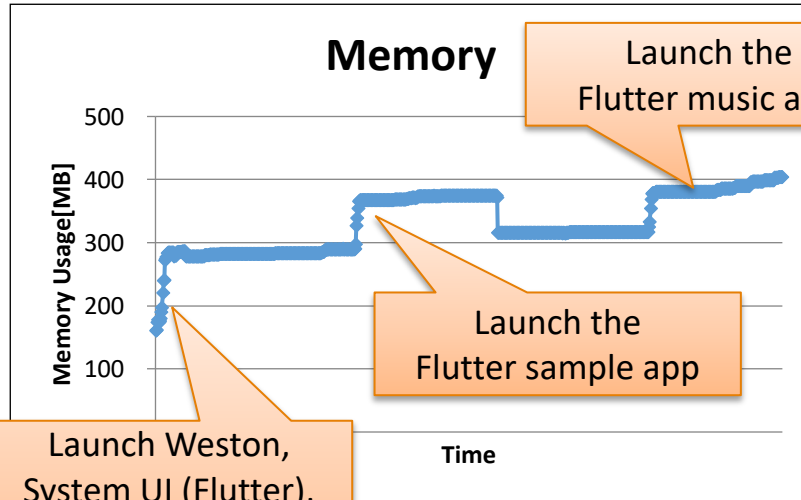


<https://developer.nvidia.com/embedded/jetson-nano-developer-kit>

Charts show performances when running the demo app shown in the video from booting on Jetson Nano.

Display: WSVGA (1024x600) / Power: AC / Flutter: Release mode

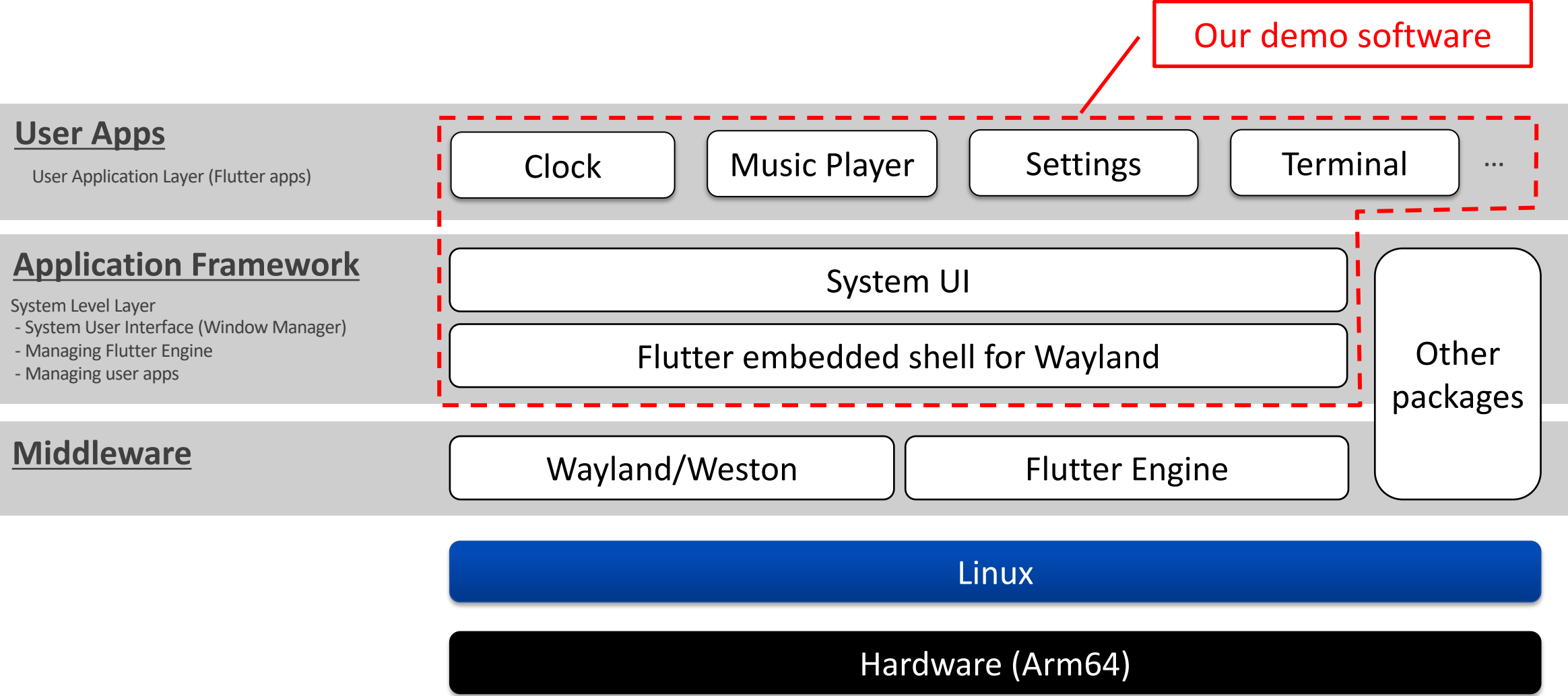
Flutter Engine has four threads



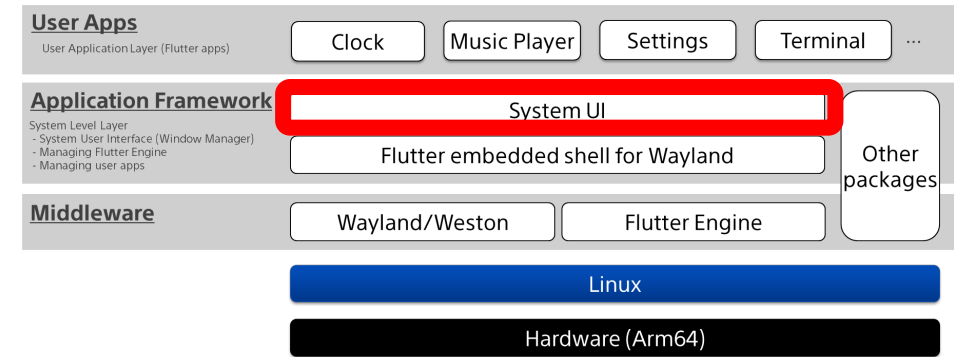
Agenda

- Background
- Overview of the new approach
- Demo Video
- Details on our approach
- Summary

Software Architecture of Demo App



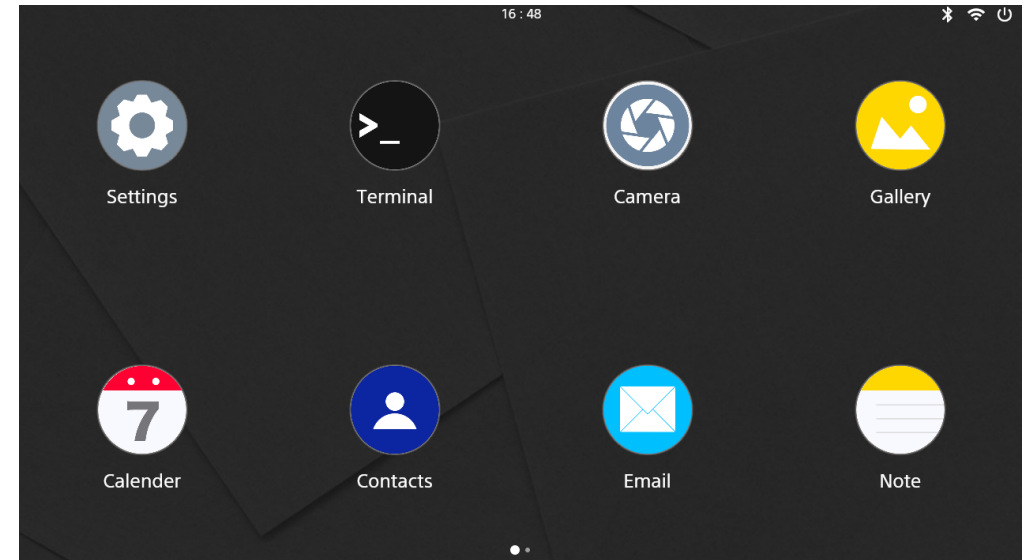
System UI (prototype for demo)



- System-UI prototype was developed as the Flutter app written in Dart and C++

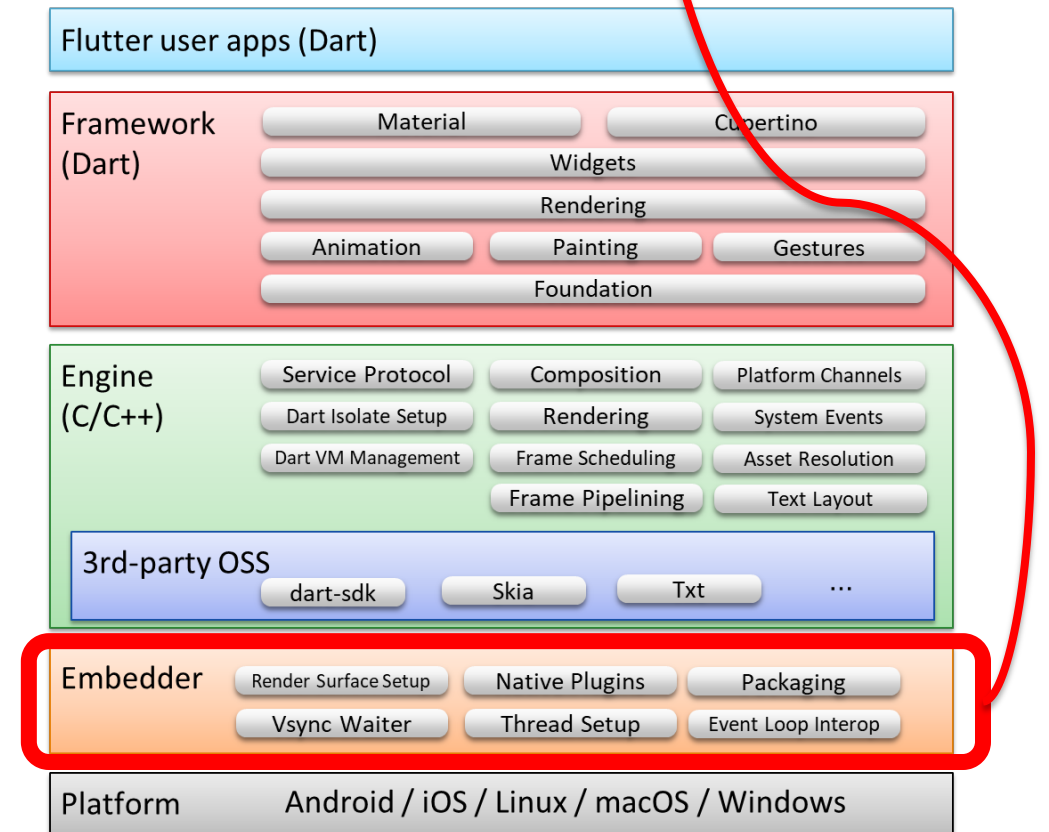
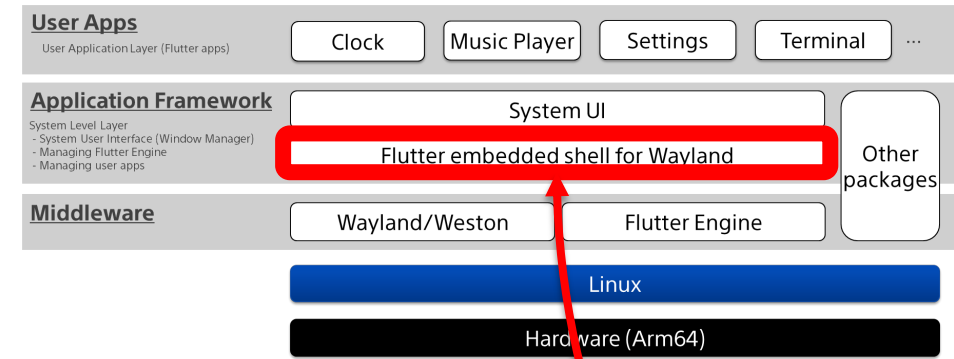
➤ Features

- App launcher
- Installing / Uninstalling apps
- Clock



Flutter embedded shell for Wayland

- Management of the Flutter Engine
- Simple window manager
 - Small Wayland shell instead of Weston-desktop-shell
 - Creating EGL surface
 - User input (Mouse, touch, keyboard) support
- Others
 - Communication between Dart (Flutter Apps) and platform native programs
 - System locale
 - etc.
- For the platform-independent part, we have used the Flutter official source code

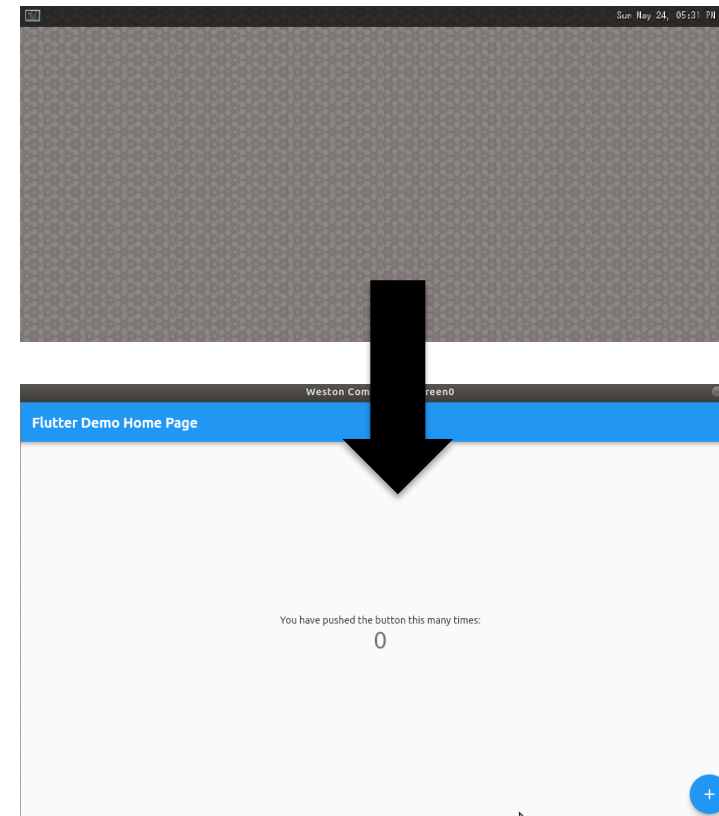


Wayland client

- We use Weston (Wayland compositor)
- Run the graphics shell for Flutter as a desktop client on Weston by using client-shell in *weston.ini*

```
1 [core]
2 idle-time=0
3
4 [shell]
5 client=/usr/bin/flutter_linux_shell
6 panel-position=none
7 animation=zoom
8
9 [flutter_linux_shell]
10 show-cursor=true
11 flutter-project-path=/etc/flutter/sample
```

weston.ini (sample)

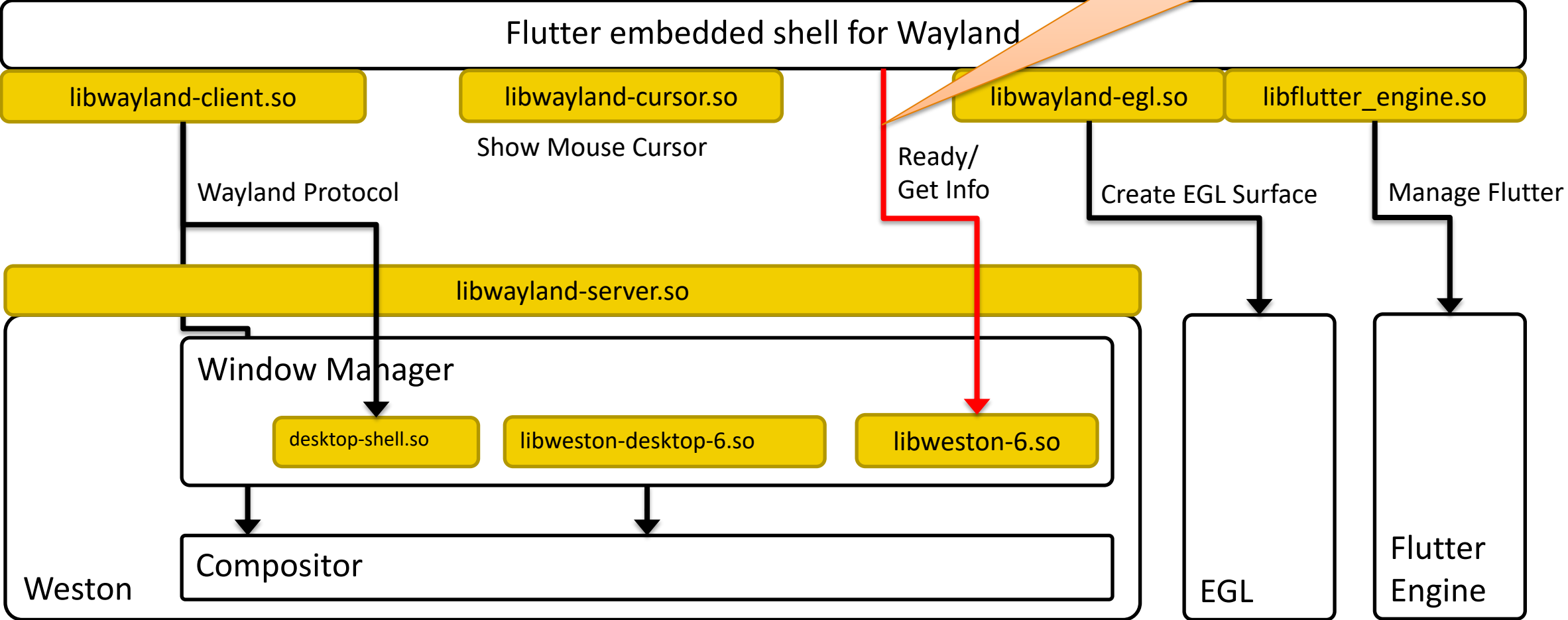


weston-
desktop-shell

Graphics shell for
Flutter
on Wayland

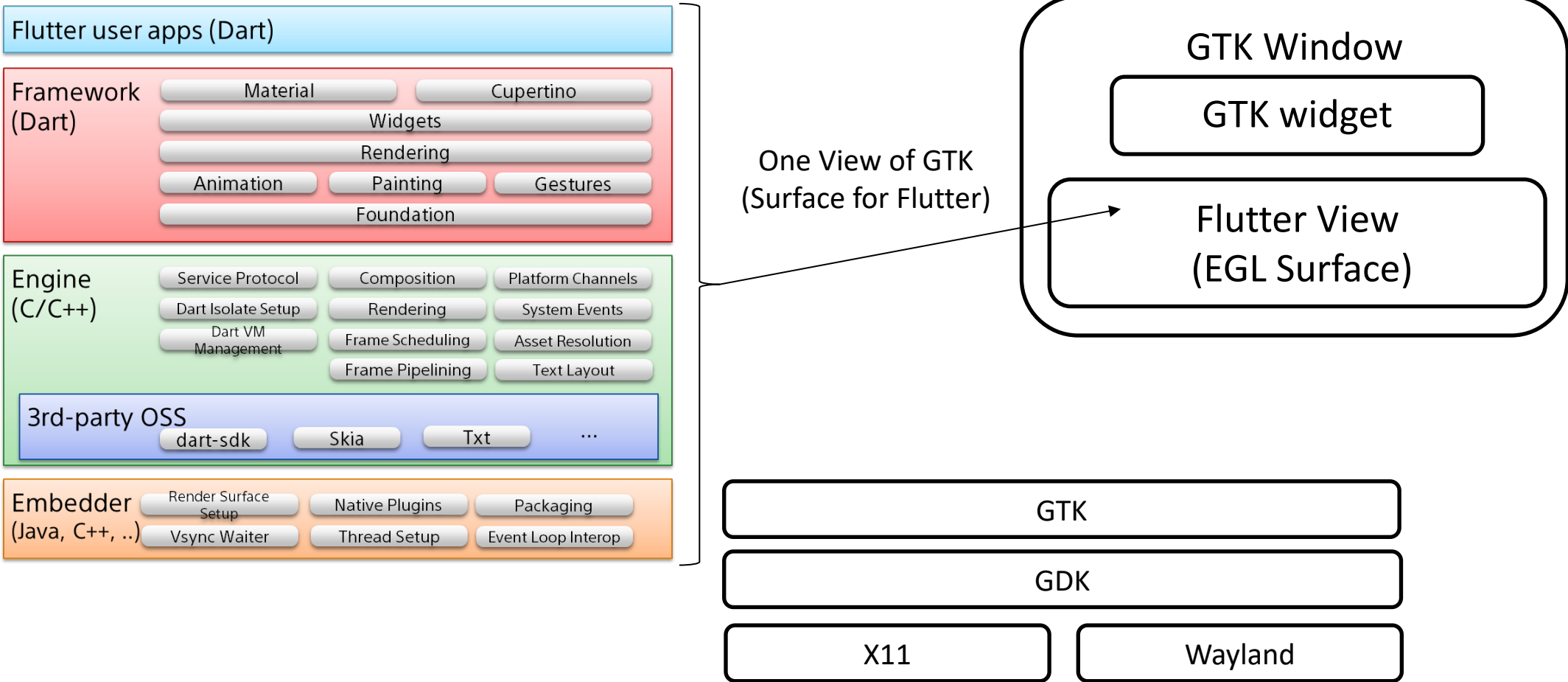
Weston API

Weston APIs
weston_desktop_shell_desktop_ready
weston_config_get_name_from_env
...



Flutter Linux desktop (OSS official support version)

➤ Official Flutter for Linux uses GTK for the graphics shell

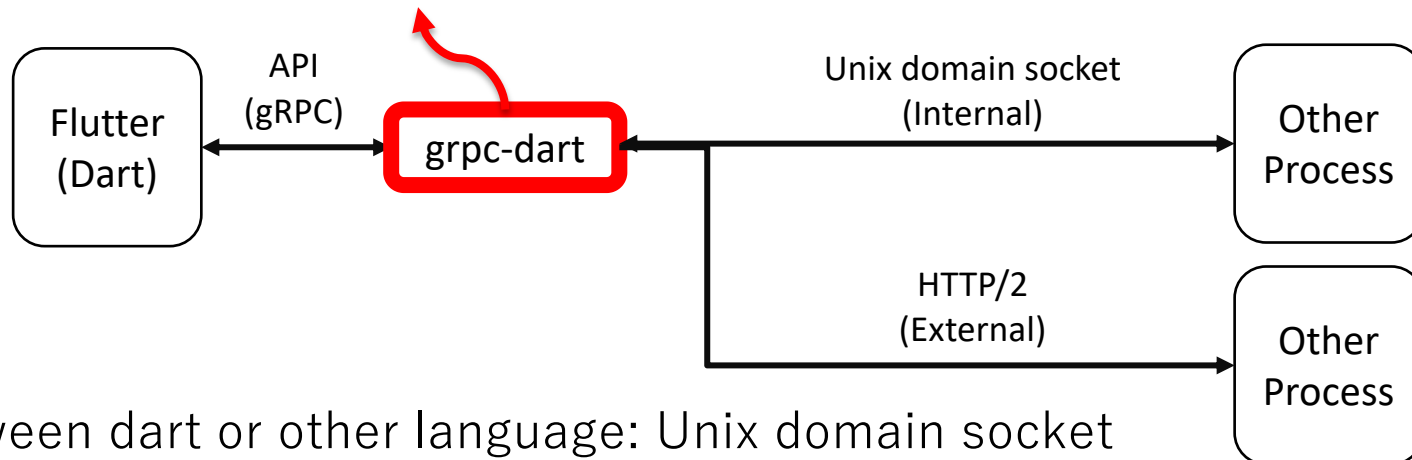


IPC / Binding to native code in Flutter

- Communication API is provided by Flutter
 - Method Channel
 - Event Channel
 - Basic Message Channel
- dart:ffi
 - Foreign Function Interface for interoperability with the C programming language
- Unix domain sockets in dart:io
- 3rd party library
 - A native Dart client implementation of D-Bus
 - <https://github.com/canonical/dbus.dart>
 - grpc-dart
 - <https://github.com/grpc/grpc-dart>
 - ...

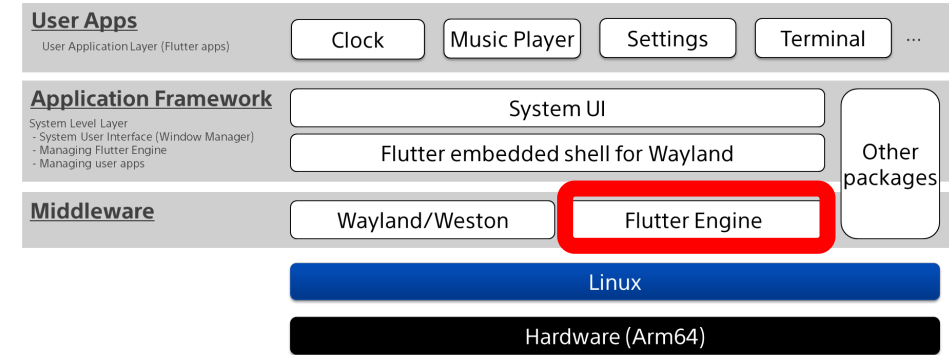
Support Unix domain socket in grpc-dart

- **grpc-dart** is the gRPC library in Dart implementation
 - <https://github.com/grpc/grpc-dart>
- We have been contributing to support Unix domain socket in grpc-dart
 - <https://github.com/grpc/grpc-dart/pull/327>



- Use case
 - IPC between dart or other language: Unix domain socket
 - External communication between the device and other devices: HTTP/2

Flutter Engine for Linux on Arm64 Architecture



- Flutter Engine is the core rendering engine library by using mainly **Skia** and **Dart-VM** (Virtual Machine)
 - <https://github.com/flutter/engine>
- Flutter doesn't currently officially support Linux Arm64 hosts
- We have been contributing to support Linux on Arm64 hosts
 - <https://github.com/flutter/engine/pull/20254>
 - <https://github.com/flutter/buildroot/pull/390>
- You can use Flutter in Arm64 embedded systems now

Development Environment of Flutter

- Flutter provides the Flutter SDK to develop and debug a Flutter app
 - <https://github.com/flutter/flutter>
- Flutter doesn't currently officially support Arm64 hosts such as Linux, macOS, Windows.
- We have been contributing to support multi-architecture hosts (especially Linux)
 - Works on Linux Arm64 Hosts
 - Cross-builds on Linux x64 Hosts for Arm64 targets etc.
 - <https://github.com/flutter/flutter/pull/61221>
 - <https://github.com/flutter/website/pull/4740>

Agenda

- Background
- Overview of the new approach
- Demo Video
- Details on our approach
- Summary

Summary

- Introducing the new GUI approach using Flutter in embedded systems
- Our motivation and purpose
 - Use Flutter in Arm64 and Linux Embedded Systems
- Future works
 - Further contributing to Flutter for Linux
 - Flutter for Linux is *α* version now
 - To be promoted to the official version as soon as possible

Thank you for your time



**Embedded Linux
Conference**
Europe