Torizon™

# CONTAINERIZE YOUR QT EMBEDDED APPLICATION

Torizon™

# DEFINE A CONTAINER

Torizon™

*A container is a colorful metal box.*
*Apart from fancy colors and logos, all containers look the same.*
*They are boring.*

Torizon™


By Maersk Line - Malcolm McLean at railing, Port Newark, 1957, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=27640875

Torizon™

# WHY CONTAINERS?

- Loading cost of a traditional ship 5.96$/ton
- Loading cost of a container ship 0.16$/ton

Torizon™

# CONTAINER ADVANTAGES

- Standard package
- Shared infrastructure
- Can be loaded on ships,trucks and trains

Torizon™

# *Containers changed global trade.*


By Robert Schwemmer for NOAA's National Ocean Service - Flickr: Container Ship, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=19073448

Torizon™

# HOW DOES THIS RELATES TO SOFTWARE?

Torizon™

*Containers are a way to package your software in a standard and easy to deploy way*

Torizon™

# STANDARD PACKAGE

*It works on my machine!*

*Now You can ship your machine.*

Torizon™

# THE DOCKERFILE

```
FROM debian:bullseye-slim

RUN apt-get update && apt-get install -y mydependencies
COPY myapp /usr/bin/myapp
CMD /usr/bin/myapp
```

Torizon™

# COMMON INFRASTRUCTURE

- container build
- container runtime
- container registry

Torizon™

# CAN BE LOADED ON SHIPS,TRAINS AND TRUCKS

- servers
- cloud
- devices

# CONTAINERS ON DEVICES?

Torizon™

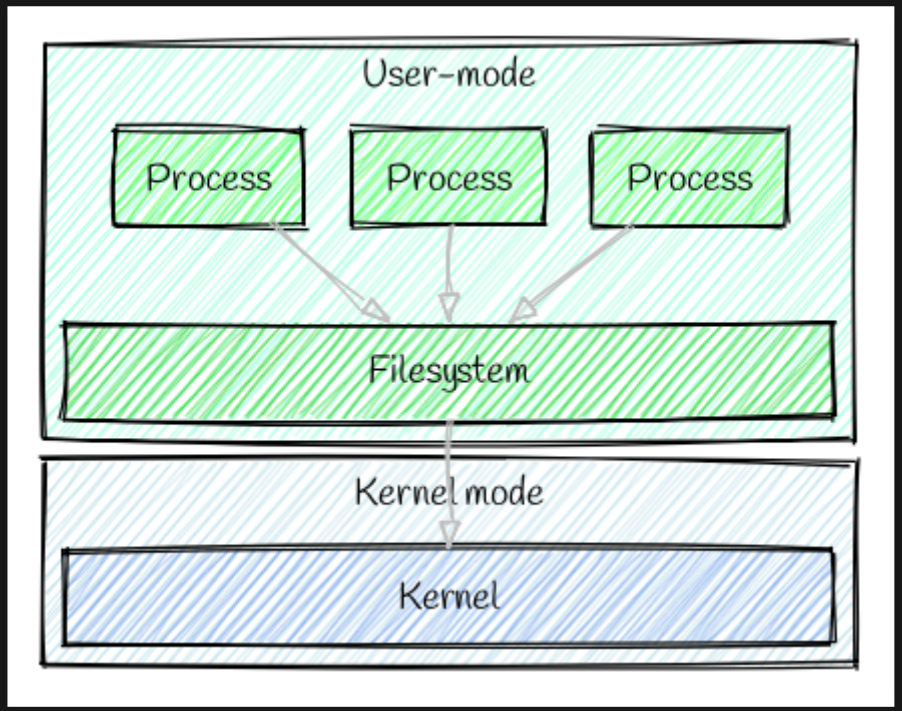
# CONTAINERS ARE DESIGNED TO BE EFFICIENT

Torizon™

*Containers are not Virtual Machines*

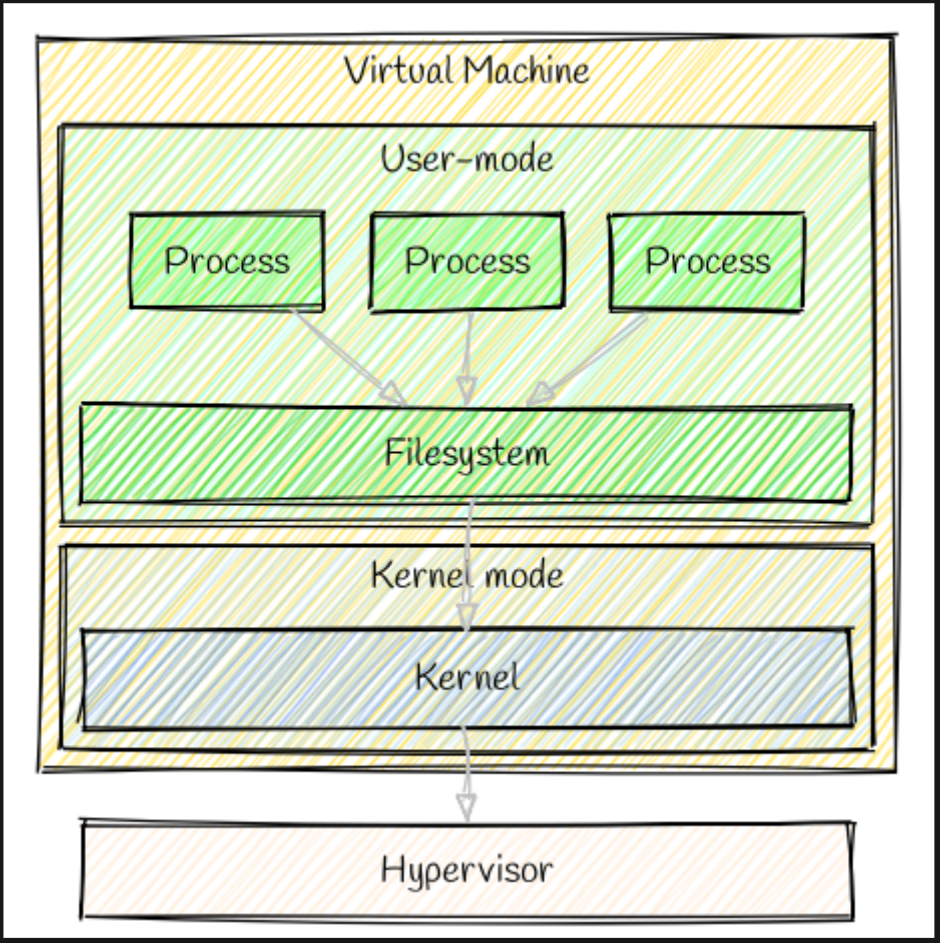
Torizon™

# LINUX



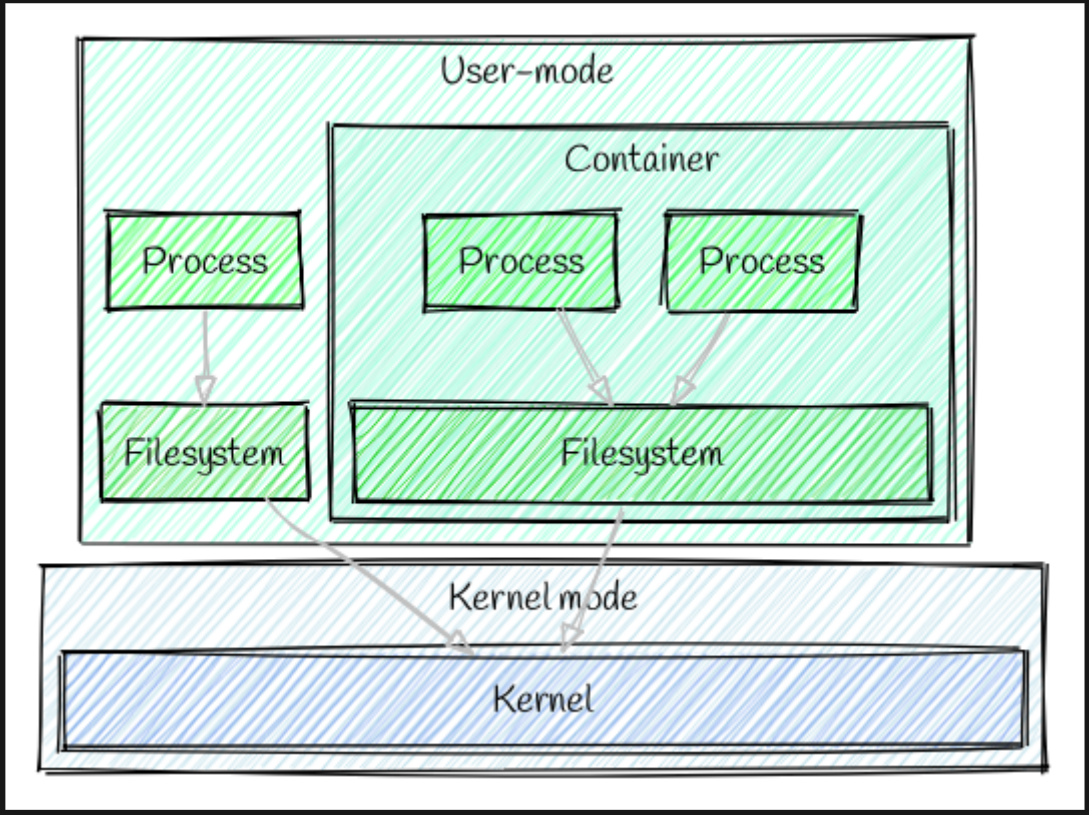
Torizon™

# VIRTUAL MACHINE



Torizon™

# CONTAINERS



Torizon™

*A Container is a “sandbox” where an application/service can run in a well defined environment.*

Torizon™

# DISADVANTAGES

- Footprint of the runtime
- Increased storage/RAM footprint

Torizon™

# LAYERS

- A container is not an archive
- Shared layers reduce overhead

Torizon™

# APPLICATIONS AS CONTAINERS

- Pack application and all its user-mode dependencies
- Limit inter-dependencies
- Leverage modern tools and technologies
- Invest time and effort in your application, not in the OS

Torizon™

# WHY NOT JUST USING A DISTRIBUTION?

- Distributions are not made for embedded
- Secure unattended updates
- Resource optimization

Torizon™

# TORIZON CORE

- Based on Yocto/OpenEmbedded
- Open source
- Runs containers
- Provide support for OTA updates

Torizon™

# CONTAINERS IN EMBEDDED

- Hardware access
- Startup
- Access to host OS

Torizon™

# STARTUP

- boot time
- multiple services

Torizon™

# HARDWARE ACCESS

- Container = Sandbox
- Everything is a file
- Holes in the sandbox!

Torizon™

# GRAPHICS

- Wayland vs X11
- client-server

Torizon™

# DOCKER-COMPOSE

```
version: "2.4"
services:
  weston:
    image: torizon/weston-vivante:2
    environment:
      - ACCEPT_FSL_EULA=1
    network_mode: host
    volumes:
      - type: bind
        source: /tmp
        target: /tmp
      - type: bind
        source: /dev
        target: /dev
      - type: bind
```

Torizon™

# DEMO

Torizon™

# ACCESS TO HOST OS

| Requirement | Solution |
|---|---|
| data storage | shared folders / volumes |
| configuration | dbus |
| run containers | docker socket |
| reboot/updates | dbus, custom |

Torizon™

# DEVELOPMENT TOOLS

- Visual Studio Code
- Qt Creator (not officially supported)

Torizon™

# CONTAINERIZE QT

- build
- runtime

Torizon™

# BUILD

- create build container (with cross-compilation)
- install/build Qt
- build your app

Torizon™

# BUILD CONTAINER (DEBIAN)

```
FROM debian:bullseye

ARG ARCH_ARG=armhf
ARG CROSS_COMPILE_ARG="arm-linux-gnueabihf"

ENV CROSS_COMPILE=${CROSS_COMPILE_ARG}-
ENV PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/lib/${CROSS_COMPIL

RUN dpkg --add-architecture ${ARCH_ARG}
RUN apt-get -q -y update \
    && apt-get -q -y install crossbuild-essential-${ARCH_ARG}
    qt5-qmake:armhf qtbase5-dev:armhf
```

Torizon™

# RUNTIME

- create container to host application
- run required services
- share devices
- design deployment method

# RUNTIME CONTAINER (DEBIAN)

```
FROM debian:bullseye-slim

RUN apt-get -y update && \
        apt-get install -y \
        libqt5quick5-gles \
        libqt5quickparticles5-gles \
        libqt5concurrent5 \
        libqt5dbus5 \
        libqt5network5 \
        libqt5printsupport5 \
        libqt5sql5 \
        libqt5test5 \
        libqt5widgets5 \
        libqt5xml5 \
        libqt5qml5 \
```

Torizon™

# BUILD RUNTIME

- use binformat/qemu (not needed on Windows/MacOS)
- use buildx and --platform

Torizon™

# Q&A