The distributed operating system targeting IoT devices big and small
OK, but who is that fella?
OK, but who is that fella?

Andrei Gherzan
Huawei Open Source Technology Center
Principal Open Source Architect
DELIVERING A UNIQUE USER EXPERIENCE ACROSS DIFFERENT CONSUMER DEVICES AND SCENARIOS

In a fully-connected world vision, consumer device fragmentation is impacting on our daily life...

Think about how many devices you use on a regular basis that accompany your day and how many actions you have to do to achieve a good user experience from each of them.

Building together a fully-connected all-scenario intelligent ecosystem
WHY THIS IS HAPPENING?

- Fragmented and isolated consumer devices market
- Technology silos
- Small developers ecosystem
- Lack in content creation portability, stability SDK and non-unified tools
- Communication between devices is enabled by discrete apps that are connected to the Cloud to share and sync data, with lack in performances, speed along with security and privacy issues

Building together a fully-connected all-scenario intelligent ecosystem
IoT NEEDS COMMON STANDARDS TO LIVE UP TO ITS PROMISE OF EMPOWERING CONSUMERS’ DAILY LIFE

- One single Cohesive application
- Interoperability between devices
- One broad range ecosystem
- Write once, deploy on different devices
- In-house secured data sync and sharing

Building together a fully-connected all-scenario intelligent ecosystem
Open source technology baseline, is the only way to achieve real co-operability, interoperability and communication between different devices because they are natively talking the same language.

THE INFLUENCE ON REQUIREMENTS, ROADMAP AND FEATURES IS THE KEY!
THE MISSION IS TO ACHIEVE THE GOAL OF AVOIDING TECHNOLOGY SILOS THROUGH THE PATH OF OPEN SOURCE. THAT MEANS PROVIDING EVERY DEVICE MAKER WITH THE SAME TECHNOLOGY BASELINE, IN AN OPEN, TRUSTED, TRANSPARENT, COLLABORATIVE WAY.
Reuse of existing open source software. Welcome developers across all verticals. The role of foundations like Eclipse Foundation and OpenAtom to grant neutrality and open governance.

**Community**

End to end implementation thanks to a dynamic open ecosystem.

**Implementation**

A distributed virtual device concept to provide all the device makers with the same technology baseline.

**Technology**

Distributed app and data management
Diffused app ecosystem
Multidevice app development

**Applications**

Technology freedom and inclusivity. One OS for three different size and category devices.

**Hardware**

Building together a fully-connected all-scenario intelligent ecosystem.
ONIRO HAS A LAYERED ARCHITECTURE BUILT AROUND THE YOCTO PROJECT AND BITBAKE BUILD SYSTEM.

THE YOCTO PROJECT IS VERY POPULAR IN THE EMBEDDED LINUX COMMUNITY AND PROVIDES AN EXCELLENT PLATFORM FOR DEVELOPING A HIGHLY-CUSTOMIZABLE, CROSS-KERNEL OPERATING SYSTEM.

FROM BOTTOM TO TOP, ONIRO CONSISTS OF THE KERNEL LAYER, SYSTEM SERVICES LAYER, FRAMEWORK LAYER, AND APPLICATION LAYER. IN MULTI-DEVICE DEVELOPMENT, YOCTO PROVIDES THE CAPABILITIES TO TWEAK LAYERS AND RECIPES TO REMOVE UNNECESSARY SUBSYSTEMS, FUNCTIONS, OR MODULES AS REQUIRED.
Building together a fully-connected all-scenario intelligent ecosystem

**THE ARCHITECTURE LAYERS**

**Application Layer**
When completed, it will host the system and third-party applications. Oniro applications will be able to use APIs to expose business logic as abilities that may be utilized inside other applications.

**System Service Layer**
It will contain the bulk of the differentiating features of Oniro. It will provide a complete set of capabilities essential for Oniro to offer services for applications through the Framework Layer.

**Framework Layer**
It will provide an SDK to develop Oniro applications in multiple languages such as Java, C, C++ and JS depending on the target device class and its HW constraints.

**Kernel Layer**
Oniro will support a multi-kernel design design out of the box so that appropriate OS kernels can be selected for devices with different resource limitations.
Building together a fully-connected all-scenario intelligent ecosystem

- meta-oniro: root layer
- meta-openembedded
- meta-clang
- meta-zephyr, meta-freertos, meta-liteos
- meta-riscv
- meta-openharmony: openharmony components
- meta-seco, meta-st, meta-av96, meta-intel, ...
ONIRO 22.12 “GOOFY” ROADMAP

**Alpha (June 2022)**
- LTS Toolchain
- Yocto upstream sync
- OpenHarmony upstream sync
- System OTA Linux
- Backend runtime debuggers (gdb, gdbserver, kgdb...)

**Beta (Sept 2022) Feature Freeze**
- LiteOS Kernel Linux 5.1x LTS
- Zephyr LTS
- Security: runtime and configuration hardening Linux
- System OTA Zephyr
- System OTA end-to-end including cloud backend

**GA (December 2022)**
- Zephyr LTS
- LTS Toolchain
- Yocto upstream sync
- OpenHarmony upstream sync
- Security: runtime and configuration hardening Zephyr
- System OTA Zephyr
- System OTA Linux
- Backend runtime debuggers (gdb, gdbserver, kgdb...)

Building together a fully-connected all-scenario intelligent ecosystem
ONIRO 22.12 “GOOFY” ROADMAP

**Reference Hardware**

- **SECO SBC C61** – NXP i.MX8M Mini – Linux and Zephyr
- **SECO SBC B68** – Intel Atom – Linux and Zephyr
- **SECO C23** – Rockchip – Linux and Zephyr
- **Arduino Nano BLE** – Zephyr
- **Rpi 3/4** – Linux
- **Udoo Key** – RPI RP2040 & ESP32 – Linux and Zephyr

*OpenHarmony supported targets*

* - pending IP clearance, no SLA, official support from OpenAtom

**Testing & Infrastructure**

- Integration of performance and benchmarking
- Integration of standard test toolkits
- Finalize device testing infrastructure and testing lab blueprint instructions
- Bring one additional testing lab online
- ACTS decoupling from currently supported implementations
- Security/release tooling: report pending CVEs in all builds

**Alpha (June 2022)**

**Beta (Sept 2022)** Feature Freeze

- Integration of performance and benchmarking
- Integration of standard test toolkits
- Finalize device testing infrastructure and testing lab blueprint instructions
- Bring second additional testing lab online
- Security/release tooling: vulnerability handling tool

**GA (December 2022)**

- Integration of performance and benchmarking
- Integration of standard test toolkits
- Finalize device testing infrastructure and testing lab blueprint instructions
- Four independent testing labs online
- ACTS decoupling from currently supported implementations
- Security/release tooling:
  - Report pending CVEs in all builds
  - Vulnerability handling tool
  - Report stable version updates (available upstream)
- Oniro repositories restructuring

Building together a fully-connected all-scenario intelligent ecosystem
ONIRO 22.12 “GOOFY” ROADMAP

Alpha (June 2022)
- Gesture and Voice Input Midi Device – V1
- Smart Panel – V2 (CATS)
- Jasmine selected BPs carry fwd – V2

Beta (Sept 2022)
- Smart Home Assistant based on Mycroft – V1
- Transparent Gateway – V2

GA (December 2022)
- Gesture and Voice Input Midi Device – V1
- Smart Home Assistant based on Mycroft – V1
- Transparent Gateway – V2

Reference Blueprints
- Jasmine selected BPs carry fwd – V2

Standards & Compliance
- Lock Yocto Project target release
- Lock OpenChain /REUSE target spec
- Lock OpenHarmony target release
- Lock Matter target release
- Lock LEDGE target release

Autonomous Agency
- Intelligent Agency: Communication Plumbing
- Intelligent Agency: Virtualization Layer
- Intelligent Agency: Application Management Layer
- Intelligent Agency: Simple Cooperative Use Case
- Intelligent Agency: Intelligent Coordination Use Cases

Building together a fully-connected all-scenario intelligent ecosystem
LIFECYCLE AND RELEASE SCHEDULE

1.0 JASMINE RELEASE
   (2021)
   - KERNEL UPGRADE
   - TOOLCHAIN UPGRADE
   - USER SPEC UPGRADE
   - BSPS UPGRADE/OBSOLENCE

2.0 GOOFY RELEASE
   (2022)

3.0 DAISY RELEASE
   (2023)

Development

LTS y.1

LTS y.2

LTS y.3

BUILDING TOGETHER A FULLY-CONNECTED ALL-SCENARIO INTELLIGENT ECOSYSTEM
CONTINUOUS INTEGRATION AND DEPLOYMENT

- gitlab runners for builds, with git-repo cache, bitbake sstate and download cache
- strategic placement of jobs across repositories to ease maintenance
- lava for smoke testing on hardware and in virtual environments
- Scancode, Fossology, REUSE, Debian matcher for license compliance and SPDX SBOM

- 14 shared jobs for images officially supported
- workspace --> assembles all repos via git repo
- bitbake-workspace --> initialized bitbake build
- build-linux, build-zephyr, build-freertos, build-liteos
- build-docs
- lava-test, lava-report
- ip-scan
LAVA based, decentralized, distributed device testing

- Each member, contributor,... can add physical devices at different locations

- Device added under testing can be shared via public cloud infrastructure

- Each site can add one to hundreds of devices

- Sites broadcast their availability to a central repository / directory
• Openchain Specification 2.0
• Training, R&R, fundings, activity, IP auditing embedded into R&D
• IP compliance integrated in CI / CD
• Low Resolution SBOM
  Merge --> Scan --> Fossology → Dashboard
• High Resolution
  Fossology --> IP Auditors --> SPDX
• Releases SBOMs for alpha, beta and official yearly release
Building together a fully-connected all-scenario intelligent ecosystem

STANDARDS AND COMPLIANCE

OpenHarmony

yocto PROJECT

COMPATIBLE

OPENCHAIN

SPDX

REUSE SOFTWARE

arm SystemReady

IR
Oniro Project integrates its various components into a representative use-case called a *Blueprint*.

A blueprint shows off the OS capabilities and best practices in building software-based products. To this end, *blueprints* are a way to distill real-world products into a minimum viable product to demonstrate how partners and users may adopt Oniro Project securely in their own products.

**What is a Blueprint?**
1. It shows off a key feature or two of the product it is trying to emulate
2. It focuses on reproducing a representative user interaction for a use case
3. It is implemented on the closest appropriate reference HW platform from Oniro Project
Building together a fully-connected all-scenario intelligent ecosystem

**Gateway Blueprint:**
- WiFi AP functionality
- OpenThread Border Router functionality
- Onboarding of a mesh node in an OpenThread network
- A mesh node to participate in an OpenThread network

**Doorlock Blueprint:**
- Operating several types of locks
- Keypad input to operate the lock locally
- Secure wireless communication to operate the lock locally
- Secure communication with the lock remotely
- Secure OTA

**BLUEPRINTS**

**Arduino Nano 33 BLE board**
The **Vending Machine** Blueprint provides support for building a PoC smart vending machine with components showing the capabilities of the Oniro Project build infrastructure to easily build an end-to-end solution using multiple operating systems cooperating inside a product.

The **Vending Machine** blueprint provides full support for two complete setups. One based on Raspberry Pi 4B, an ARMv8 target, and one based on SECO B68, an X86 board.

[https://www.youtube.com/watch?v=HQ9hD63ypvl&list=Ply7t4z5SYNaQBDReZmeHAKnEchYmu0LLa&index=10](https://www.youtube.com/watch?v=HQ9hD63ypvl&list=Ply7t4z5SYNaQBDReZmeHAKnEchYmu0LLa&index=10)

CHECK THE DOCUMENTATION:
DEMO TIME
WHAT’S IN IT FOR YOU?

CONSUMERS
If you are an end user, Oniro integrates your various smart devices to implement fast connection, capability collaboration, and resource sharing between them. This way your services can be seamlessly transferred to a suitable device that delivers smooth all-scenario experience.

DEVICE MAKERS
OEMs, SILICON VENDORS, CHIPSET MANUFACTURERS
If you are an application developer, with Oniro, you will have the choice to focus on upper-layer service logic and develop applications in a much easier and more efficient way.

CONTENT CREATORS
If you are a device developer, Oniro uses a component-based software design to tailor itself to your particular device forms based on their respective resource capabilities and service characteristics.
HOW TO GET STARTED

Download ➔ Use ➔ Read the documentation ➔ Contribute into the community

Join our Mattermost Town Square chat
VISIT OUR WWW.ONIROPROJECT.ORG WEBPORTAL
Thank you!

Communication channels

- Oniro Working Group mailing list: oniro-wg@eclipse.org
- Oniro Projects (technical) mailing list: oniro-dev@eclipse.org
- Chat channel: #oniroproject at libera.chat
  - Twitter: @oniro_project #Oniro