NOKIA

SOSCON
Samsung Open Source Conference

Developer session
At the Open Source Conference 2021
5G, IoT and Edge computing

The O-RAN Alliance and the ORAN-SC Project

thoralf.czichy@nokia.com
PTL, near-RT RIC

Jul-03-2021

The O-RAN alliance was formed to increase competition

RAN openness, hardware vs software separation, programmability

O-RAN ALLIANCE

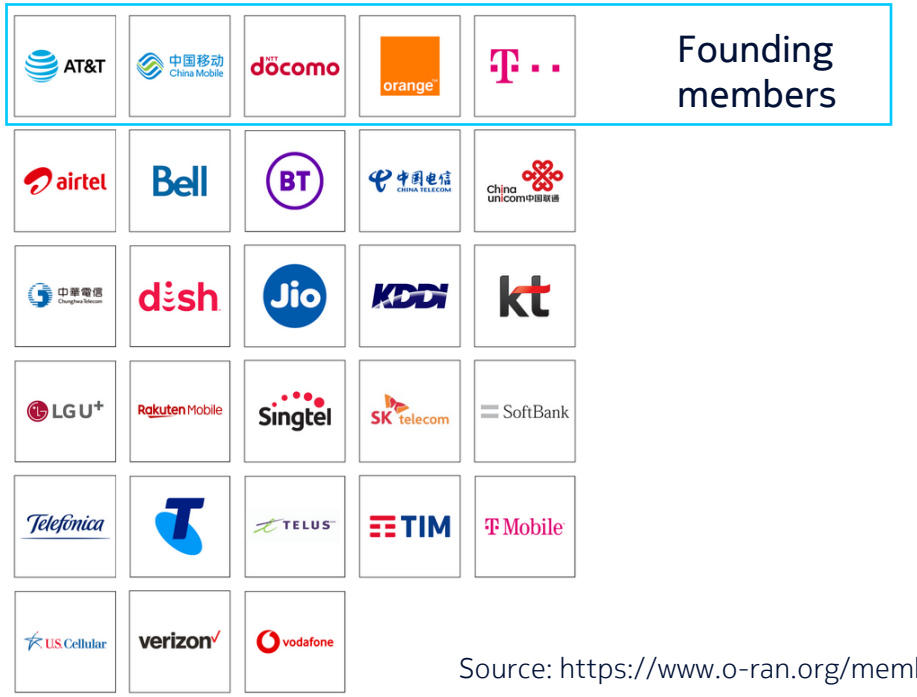
250+

contributors

- Launched June 2018
- Merging of the xRAN Forum with the C-RAN Alliance
- O-RAN Alliance announced collaboration with TIP (Telecom Infrastructure Project) in February 2020
- 9 key working groups led by operators with contributors co-chairing

28

mobile operators



Source: https://www.o-ran.org/membership

Key Objectives:

- Bring Cloud Scale Economies to RAN
- Bring Agility to RAN

Key Principles

- Lead the industry towards open, interoperable interfaces, RAN virtualization, and big data enabled RAN intelligence
- Maximize the use of common-off-the-shelf hardware and minimizing proprietary hardware
- Specify APIs and interfaces, driving standards to adopt them as appropriate, and **exploring open source** where appropriate

NOKIA

The working groups of the O-RAN alliance

#1 Use cases & overall architecture

#2 Non-real-time RIC and A1

#3 Near-real-time RIC and E2AP plus E2SM (service models) for RC, KPM, NI

#4 Open Fronthaul eCPRI 7.2x split

#5 Open F1/W1/E1/X2/Xn (re-use of 3GPP specs)

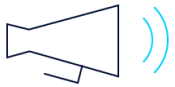
#6 Cloudification and Orchestration (O1 (3GPP reuse), O2)

#7 White-Box hardware

#8 Stack Reference Design

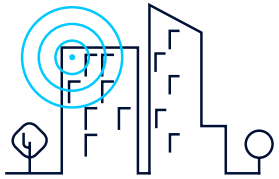
#9 Open X-haul Transport

O-RAN-SC – The O-RAN software community



Est. April 2019

- Project by the O-RAN Alliance and the Linux Foundation (LF)
- Open-source software aligned with the architecture specified by the O-RAN alliance
- Re-using Series of LF Projects, LLC (common “master LLC”, but separate divisions with limited liabilities).
- LF also provides project infrastructure



>20 companies

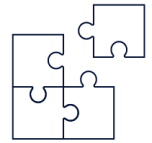
- Source code contributions from >20 companies
- 92% of the commits by the top 10 contributing companies
- License: Apache 2.0

4 releases

- A release every 6 months (in July and December):
- Amber (Nov-2019)
- Bronze (Jun-2020)
- Cherry (Dec-2020)
- Dawn (Jul-2021)
- E …
- F …
- https://wiki.o-ran-sc.org/display/REL/Releases

Tue + Wed

- Day-to-day management via the TOC (Technical oversight committee)
- Weekly meetings on Wednesdays, 5:30pm IST
- https://wiki.o-ran-sc.org/display/TOC
- Subprojects have own meeting practice
- For example, near-RT RIC has fortnightly meetings on Tuesdays, 6:30pm IST
- https://wiki.o-ran-sc.org/display/RICP/Project+meetings



SCP

- The O-RAN Specification Code Project has separate charter, but delegates administration to the O-RAN-SC TOC
- Shares infrastructure and meetings with O-RAN-SC,.
- License: O-RAN software license

O-RAN-SC subprojects

Project	PTL	Description
RICAPP	Matti Hiltunen AT&T	Near-RT RIC XApps
RIC	Thoralf Czichy Nokia	Near-RT RIC platform
OCU	Suzy Gu CMCC	UP implementations for SDAP and PDCP, eGTP-U as binary
ODUHIGH	Sachin Srivastava Radisys	Implementing L2, F1, MAC scheduler, RLC
ODULOW	Zhimin Yuan Intel	Implementing L1, Intel FlexRAN binary used
ORU	TBD NA	O-RAN Radio Unit No contributions

Project	PTL	Description
OAM	Martin Skorupski HST	yang models, RIC dashboard, O1 reference impl. + O1 client
SIM	Alex Stancu HST	Simulators for testing O-RAN, e.g, E2AP
INF	Xiahua Zhang Windriver	Cloud infra for O-RAN (WR)
INT	<open> <open>	Integration of O-RAN SC release
DOC	Weichen Ni CMCC	Documentation to readthedocs.io
NONRTRIC	John Keeney Ericsson	A1 policy mgr (used by rApps), r-app host, Enrichment data, (ML mgmt)
SMO	Mahesh Jethanandani Juniper	Integrated SMO & deployment scenarios with ONAP

Licenses of O-RAN-SC and the Specification code project

- The Specification code project shares infrastructure and meetings with O-RAN-SC
- We give repositories within subprojects the choice between contributions under **Apache 2 license** or under **ORAN Software license**, also referred to as SCP (Specification code project)
- Documentation to be contributed under **Creative Commons Attribution** 4.0 (CC BY 4.0)
- The Apache license is a very **liberal license** in terms of being able to use the source code, e.g., it includes an explicit patent license
  - For example, the Near-RT RIC subproject uses this license.
- ORAN Software license, also referred to as SCP (Specification code project) license is used in the specification code project
  - Details of the license: https://www.o-ran.org/software
  - Used for some xApps and OCU repo. Repos always in folder ”scp” in gerrit
- No contributions without an online-signed **contributor license agreement** (CLA)
  - https://wiki.o-ran-sc.org/display/ORAN/Signing+Contributor+License+Agreement
- Minor related contribution to **asn1c**
  - asn1c fork : https://github.com/nokia/asn1c (BSD2/3) - minor adaptations to make it work with the O-RAN ASN.1 specifications.

Committers

Review

- Committers are the primary contact for a component, they review code contributions

INFO.yaml

- Each repo has a set of committers. Check from the repo's INFO.yaml file (base directory) of the repo.
- Example: https://gerrit.o-ran-sc.org/r/gitweb?p=ric-plt/e2.git;a=blob;f=INFO.yaml

Merge

- Only committers can merge source code to master or maintenance branches
- Only committers can release

New committer?

- Committers can change by majority vote (>=50%) of existing committers.
- Change is automatically notified post-fact to the O-RAN SC TOC (Technical oversight committee)

Tools re-use from Linux Foundation infrastructure

gerrit: All source code + review tool

- https://gerrit.o-ran-sc.org/r/#/admin/projects/

nexus3 as image repository. Release (port 10002) and Staging registry (10004)

- Images in staging registry automatically deleted. Prefer use of release registry instead
- Also incluldes container base images: https://wiki.o-ran-sc.org/display/ORAN/ORAN+Base+Docker+Images+for+CI+Builds

packagecloud.io for binary artifacts, such as RPM and debian packages.

- Master and staging used during development
- Packaging https://wiki.o-ran-sc.org/display/ORAN/Packaging+Libraries+for+Linux+Distributions+with+CMake
- Publishing: https://wiki.o-ran-sc.org/display/ORAN/Binary+Repositories+at+PackageCloud.io

NexusIQ (hosted by LF) for license checks

- https://nexus-iq.wl.linuxfoundation.org (access limited) // Right now only A1 mediator. Checks done at end of release manually.

sonarcloud.io: static code checking and code coverage (via tests)

- All repos: https://sonarcloud.io/organizations/o-ran-sc/projects
- Example repo (rmr library): https://sonarcloud.io/dashboard?id=o-ran-sc_ric-plt-lib-rmr
- We aim for 80% code coverage

readthedocs.io: automatically generated documentation

- Results: https://docs.o-ran-sc.org/en/latest/projects.html#near-realtime-ran-intelligent-controller-ric
- Instructions: https://wiki.o-ran-sc.org/display/ORAN/Configure+Repo+for+Documentation

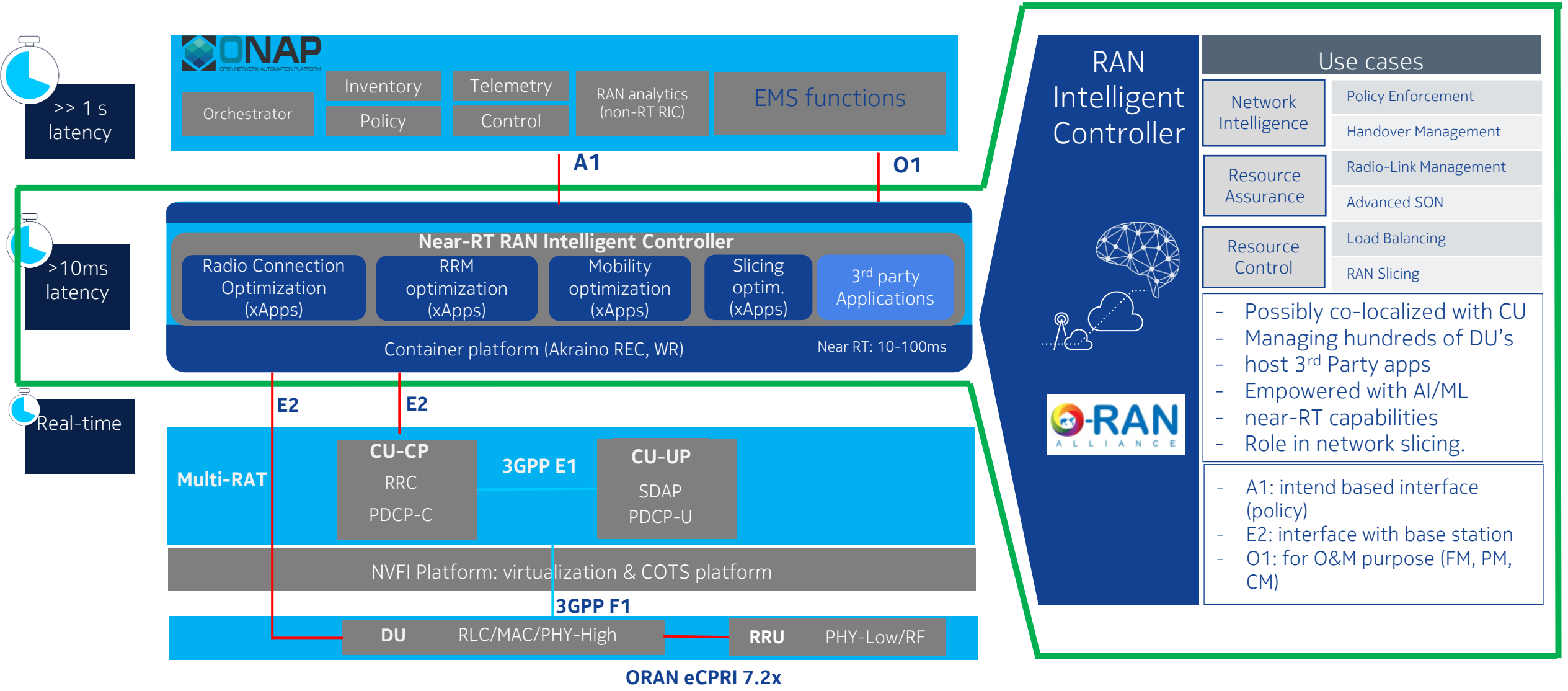
Testing

- Robot framework used in test cases of E2 manager, routing manager and integration test
- Unit tests: cgreen, gtest, …

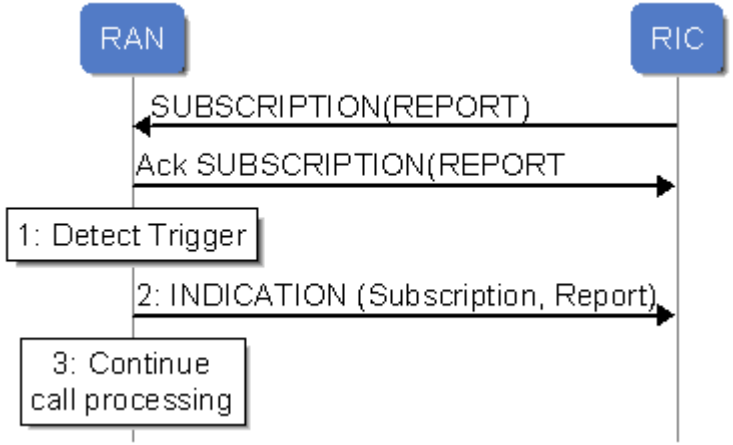
More details

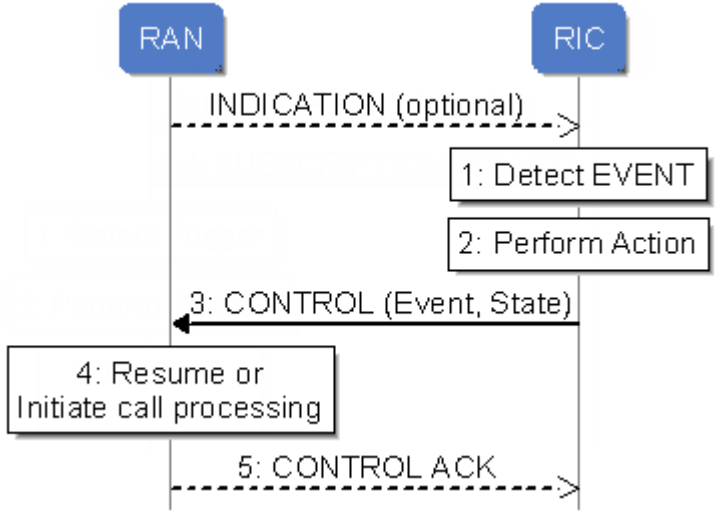
https://wiki.o-ran-sc.org/display/ORAN/O-RAN+Developer%27s+Guide+to+CI+Resources+and+Processes+at+the+LF

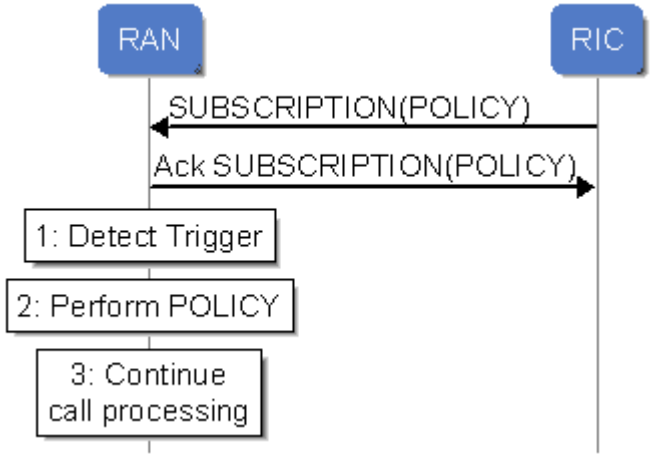
What is the near-RT RIC: architecture and key requirements

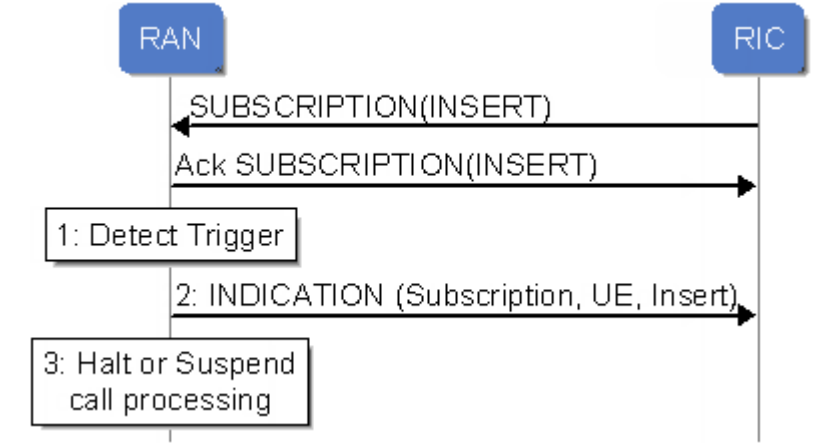


E2AP functional procedures

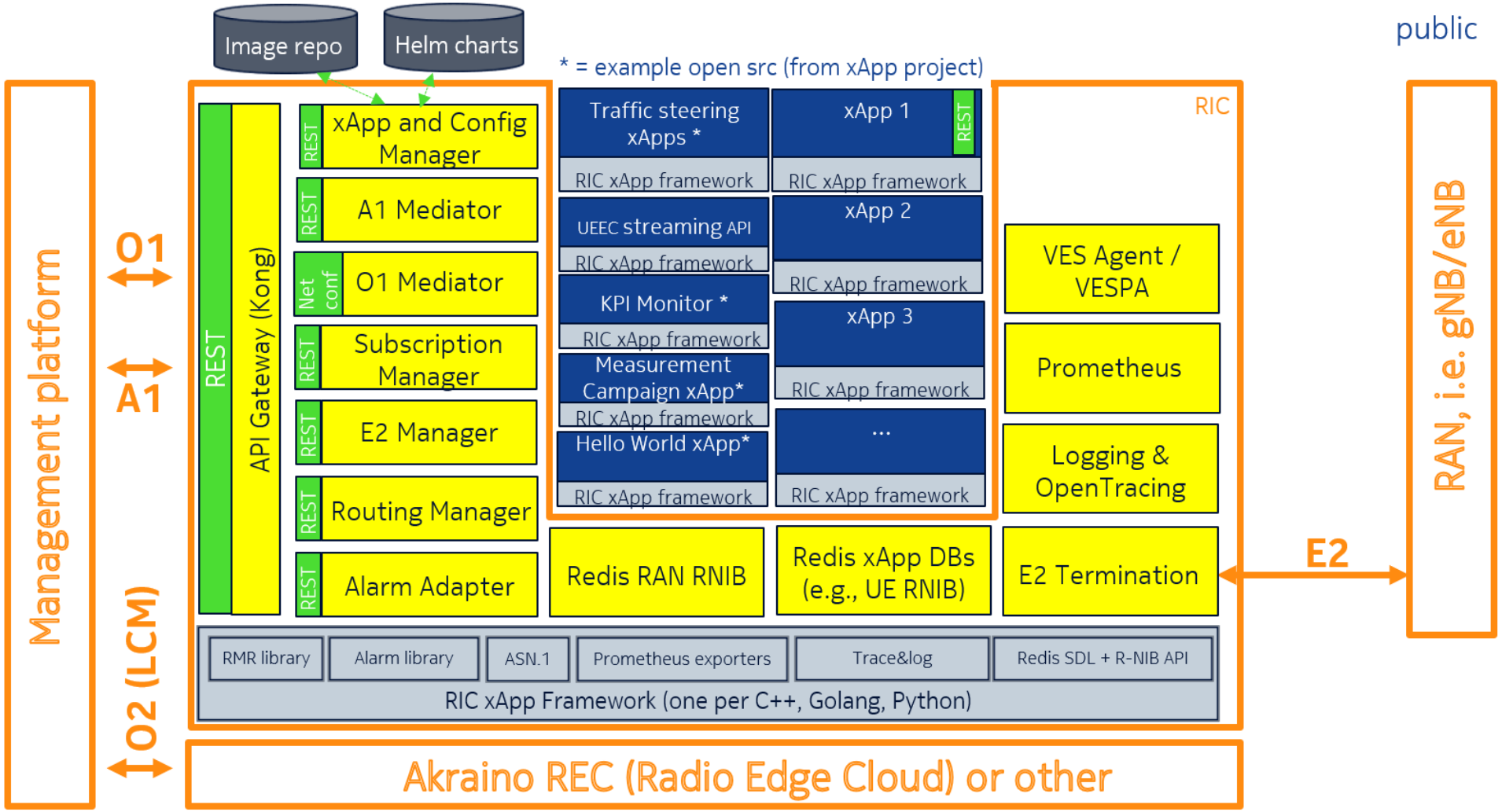








The near-RT RIC platform components



E2 principles

- Amber implemented a pre-spec version of E2AP. Bronze, Cherry and Dawn implement E2AP version v01.00. E2AP version 2.0 is currently in the O-RAN approval cycle.
- The RIC **E2AP** (Application protocol) specification (ORAN-WG3.E2AP-v01.00.00) defines the general protocol by which the near-RT RIC and RAN (gNB, eNB, CU-{CP,UP}, DU) communicate.
- More detailed **E2SM** (Service model) specifications define the function-specific protocol that is implemented on top of the E2AP specification. Typical functions are X2AP, F1AP, E1AP, S1AP, NGAP interfaces and RAN internal functions UE, Cell, Node.
  - Now: Formal O-RAN E2SMs: Network interface: ORAN-WG3.E2SM-NI-v01.00.00 and Key performance monitoring: ORAN-WG3.E2SM-KPM-v01.00.00
  - For example, while the E2 specification defines the concept of event triggers, it is the E2SM for NI that defines the specific triggers in the X2/F1/E1/… function based on matching X2/F1/E1/… AP message type, or X2/F1/E1/… IE.
  - E2SMs are an agreement between xApp and E2SM function on E2 node. To the RIC platform E2SMs are opaque.
  - The implementation of the E2 service model on gNB side requires explicit feature development on O-RAN CU/DU side.