



**PROJETO DE ALTO NÍVEL – PAN**

**ARQUITETURA DE SOFTWARE DO  
PLANO DE GERENCIAMENTO DA O-RU**

A2.4 – Desenvolvimento de Software

OpenRAN@Brasil – Fase 2

## Sumário

Sumário.....	2
Lista de Figuras.....	4
Lista de Acrônimos.....	5
1 Introdução.....	7
2 Visão Geral do Modelo C4.....	8
2.1 Abstrações.....	8
2.1.1 Sistema de Software.....	9
2.1.2 Contêiner.....	9
2.1.3 Componente.....	9
2.1.4 Código.....	9
2.2 Diagramas de Arquitetura.....	9
2.2.1 Diagrama de Contexto.....	9
2.2.2 Diagrama de Contêineres.....	10
2.2.3 Diagrama de Componentes.....	10
2.2.4 Diagrama de Código.....	10
3 Premissas de Software.....	11
3.1 Visão Geral do Software de Gerenciamento.....	11
3.1.1 Inicialização do Dispositivo.....	11
3.1.2 Gerenciamento de Software.....	12
3.1.3 Gerenciamento de Configuração.....	12
3.1.4 Gerenciamento de Falhas.....	12
3.2 Open M-Plane.....	12
3.3 Plataforma de Hardware.....	13
3.4 Sistema Operacional.....	13
3.5 Linguagem de Programação.....	13
3.6 Ambiente de Desenvolvimento.....	14
4 Arquitetura de Software.....	15
4.1 Contexto de Sistema.....	15
4.2 Visão de Contêineres.....	16
4.2.1 YANG Handler para Gerenciamento de Usuários.....	17
4.2.2 YANG Handler para Gerenciamento de Software.....	18
4.2.3 YANG Handler para Gerenciamento do Plano de Sincronismo.....	19
4.2.4 YANG Handler para Gerenciamento do Plano de Usuário.....	20
4.2.5 YANG Handler para Gerenciamento de Falhas.....	21

4.2.6	YANG Handler para Gerenciamento de Desempenho.....	23
5	Referências.....	25
6	Histórico de Alterações do Documento Consolidado.....	26
7	Execução e Aprovação.....	27

---

## Lista de Figuras

<a href="#"><u>Figura 1 - Níveis de diagramas do modelo C4 para visualização da arquitetura de software.</u></a>	<a href="#"><u>8</u></a>
<a href="#"><u>Figura 2 - Abstrações utilizadas para descrever um sistema de software.</u></a>	<a href="#"><u>9</u></a>
<a href="#"><u>Figura 3 - Diagrama de blocos do projeto de referência destacando o sistema de processamento Zynq PS.</u></a>	<a href="#"><u>13</u></a>
<a href="#"><u>Figura 4 - Diagrama de contexto ilustrando a interação do M-Plane com elementos externos.</u></a>	<a href="#"><u>15</u></a>
<a href="#"><u>Figura 5 - Diagrama de contêineres do sistema de software M-Plane.</u></a>	<a href="#"><u>16</u></a>
<a href="#"><u>Figura 6 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de usuários.</u></a>	<a href="#"><u>17</u></a>
<a href="#"><u>Figura 7 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de software.</u></a>	<a href="#"><u>18</u></a>
<a href="#"><u>Figura 8 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de sincronismo.</u></a>	<a href="#"><u>19</u></a>
<a href="#"><u>Figura 9 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento do plano de usuário.</u></a>	<a href="#"><u>20</u></a>
<a href="#"><u>Figura 10 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de falhas.</u></a>	<a href="#"><u>21</u></a>
<a href="#"><u>Figura 11 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de desempenho.</u></a>	<a href="#"><u>23</u></a>

### Lista de Acrônimos

APU	Application Processing Unit
BER	Bit Error Rate
C-Plane	Control Plane
CU	Centralized Unit
DFE	Digital Front-End
DHCP	Dynamic Host Configuration Protocol
DU	Distributed Unit
EVK	Evaluation Kit
FPGA	Field Programmable Gate Array
IDE	Integrated Development Environment
IP	Intellectual Property
MCS	Modulation Code Scheme
M-Plane	Management Plane
NETCONF	Network Configuration
NMS	Network Management System
O-RAN	Open Radio Access Network
O-RU	Open Radio Unit
PC	Personal Computer
PTP	Precision Time Protocol
QEMU	Quick Emulation
RAN	Radio Access Network
RFSoc	Radio Frequency System-on-Chip
RIC	Radio Intelligent Controllers
RF	Radio Frequency
RU	Radio Unit
RX	Receiver
SLAAC	Stateless Address Autoconfiguration
SNR	Signal-to-Noise Ratio
SoC	System-on-Chip

TX	Transmitter
U-Plane	User Plane
XML	Extensible Markup Language
YANG	Yet Another Next Generation

## 1 Introdução

Este documento apresenta um guia abrangente dos aspectos arquitetônicos do software de gerenciamento da unidade de rádio de interface aberta (*open radio unit* – O-RU) em desenvolvimento no programa OpenRAN@Brasil Fase 2 através de esforços conjuntos de diferentes instituições de ciência e tecnologia (CPQD, Inatel, Eldorado e RNP), oferecendo uma representação estruturada do seu projeto, componentes, interações e implementação.

No contexto da arquitetura da rede aberta de acesso por rádio (*open radio access network* – Open RAN) há um elemento vital denominado Plano de Gerenciamento (*management plane* – M-Plane), atuando como o centro de controle para elementos da rede de acesso por rádio (*radio access network* – RAN). Sua função principal é supervisionar e otimizar o desempenho e a manutenção das unidades de rádio, permitindo que os operadores de rede gerenciem eficientemente sua infraestrutura RAN. Isso envolve a configuração das O-RUs, monitoramento de desempenho e implementação de políticas de tráfego e acesso, além de promover a interoperabilidade e reduzir as restrições normalmente encontradas em produtos de fornecedores de redes 5G tradicionais. Ao adotar componentes desagregados e padrões abertos, o Open RAN se afasta das estruturas de rede tradicionais e fechadas. O M-Plane do Open RAN está estrategicamente posicionado para revolucionar os sistemas de telecomunicações móveis, tornando-os mais adaptáveis, econômicos e abertos. Isso, por sua vez, fomenta a inovação e a concorrência na indústria.

Os principais objetivos deste documento são:

- Fornecer uma visão geral de alto nível da arquitetura de software para desenvolvedores, arquitetos, gerentes de projeto e outras partes interessadas;
- Facilitar a comunicação eficaz e a compreensão da arquitetura do sistema, auxiliando em discussões, tomada de decisões e manutenção contínua;
- Servir como um recurso fundamental para alinhar os esforços da equipe de desenvolvimento com os objetivos e restrições arquitetônicas definidas;
- Consolidar um ponto de referência para futuras decisões e modificações relacionadas à arquitetura.

A arquitetura do software é documentada e representada utilizando-se o modelo C4 [1], tornando-a abrangente e de fácil compreensão. Essa abordagem permite comunicar eficazmente a arquitetura a um público diversificado, alinhar os esforços de desenvolvimento com os objetivos arquitetônicos definidos e manter um registro atualizado do projeto do sistema através de representações visuais, descrições e detalhes relevantes para garantir uma compreensão clara do software de gerenciamento da O-RU.

## 2 Visão Geral do Modelo C4

O modelo C4 constitui uma ferramenta para auxiliar equipes de desenvolvimento a descrever e comunicar a arquitetura de software, tanto durante sessões iniciais de projeto quanto ao documentar retrospectivamente uma base de código já existente, oferecendo uma abordagem para criar mapas do código em vários níveis de detalhes, de maneira semelhante a um mapa geográfico que pode ser ampliado ou reduzido para visualizar áreas territoriais de interesse.

O modelo C4 adota uma abordagem de “abstração em primeiro lugar” para a criação de diagramas de arquitetura de software, fundamentada em abstrações que refletem a forma como arquitetos e desenvolvedores concebem e constroem sistemas. Sua quantidade reduzida de abstrações e tipos de diagramas o torna acessível e fácil de ser utilizado.

É importante destacar que não é obrigatório empregar todos os quatro níveis de diagrama mostrados na Figura 1, mas apenas aqueles que agregam valor – os diagramas de Contexto do Sistema e de Contêiner, por exemplo, são suficientes para muitas equipes de desenvolvimento de software.

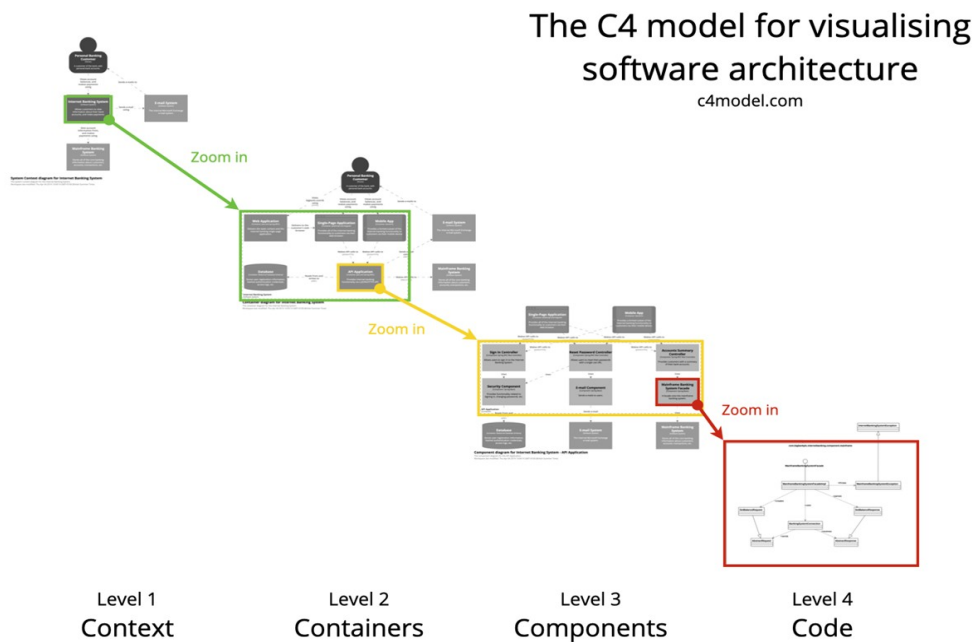


Figura 1 - Níveis de diagramas do modelo C4 para visualização da arquitetura de software.

### 2.1 Abstrações

Conforme ilustrado na Figura 2, o modelo C4 é construído a partir de um conjunto comum de abstrações que são usadas para descrever a estrutura estática de um sistema de software.

Um Sistema de Software é composto por um ou mais Contêineres (aplicações e armazenamentos de dados), cada um contendo um ou mais Componentes, que por sua vez são implementados por um ou mais elementos de Código (classes, interfaces, objetos, funções, etc.).



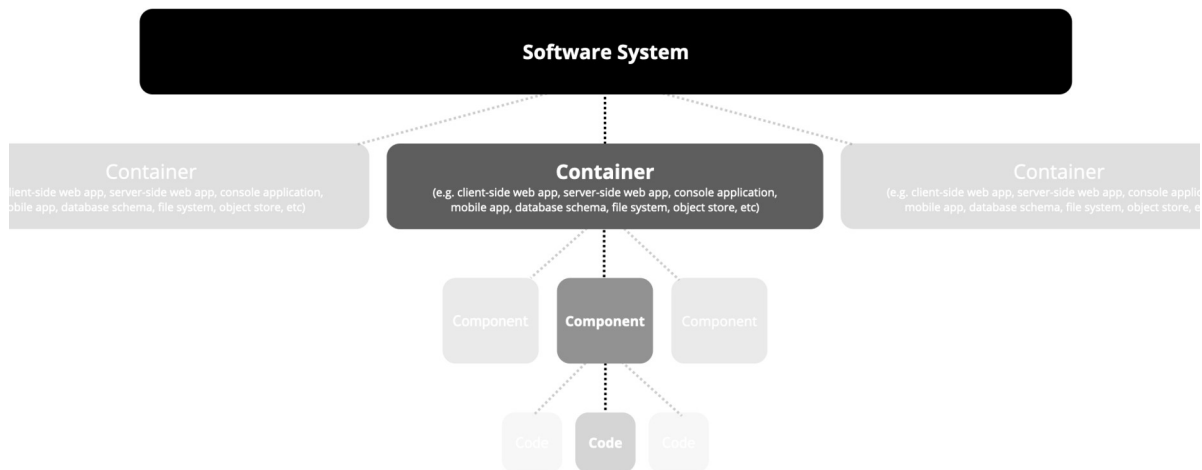


Figura 2 - Abstrações utilizadas para descrever um sistema de software.

### 2.1.1 Sistema de Software

O Sistema de Software representa o nível mais alto de abstração e descreve algo que proporciona valor aos usuários, sejam eles humanos ou não. Isso engloba tanto o sistema de software que está sendo modelado quanto outros com os quais haja alguma dependência.

### 2.1.2 Contêiner

No modelo C4, um contêiner representa uma aplicação ou um repositório de dados. É algo que precisa estar em execução para que o sistema de software geral funcione. Em termos práticos, um contêiner pode ser algo como uma aplicação web (no lado do servidor ou do cliente), uma aplicação de computador ou celular, um banco de dados, um sistema de arquivos, etc.

### 2.1.3 Componente

Neste contexto, um componente é um agrupamento de funcionalidades relacionadas encapsuladas por uma interface bem definida. É importante observar que todos os componentes dentro de um contêiner geralmente são executados no mesmo espaço de processo. No modelo C4, os componentes não são unidades separadamente implantáveis.

### 2.1.4 Código

Constitui a implementação propriamente dita dos códigos de software (classes, interfaces, objetos, funções, etc.).

## 2.2 Diagramas de Arquitetura

Como mostrado na Figura 1, o modelo C4 dispõe de quatro diagramas de representação da arquitetura de software: de contexto, de contêineres, de componentes e de código.

### 2.2.1 Diagrama de Contexto

O Diagrama de Contexto serve como um ponto inicial para a representação e documentação de um sistema de software, mostrando uma visão ampla e geral

(panorâmica) do cenário. A ênfase recai sobre as pessoas (atores, papéis, etc.) e os sistemas de software, em detrimento de tecnologias, protocolos e outros pormenores de baixo nível, sendo assim compreensível por pessoas não familiarizadas com aspectos técnicos.

### **2.2.2 Diagrama de Contêineres**

Permite uma aproximação da visão para a fronteira do sistema. Fundamentalmente, um contêiner é uma unidade independente e executável/implantável (por exemplo, um espaço de processo separado) que executa código ou armazena dados.

O Diagrama de Contêineres expõe a configuração geral da arquitetura de software e a distribuição de responsabilidades entre seus contêineres, destacando as principais escolhas tecnológicas e a intercomunicação entre eles. É um diagrama simplificado e de alto nível, concentrando-se nos aspectos tecnológicos.

### **2.2.3 Diagrama de Componentes**

O Diagrama de Componentes ilustra como um contêiner é composto por vários componentes, apresentando esses blocos estruturais, suas interações, responsabilidades e detalhes tecnológicos e de implementação.

### **2.2.4 Diagrama de Código**

Finalmente, é possível detalhar cada componente para demonstrar como ele é implementado em código. Porém, esse é um nível de detalhe opcional e tipicamente gerado automaticamente, conforme necessário, em ferramentas ou ambientes de desenvolvimento integrados (*integrated development environment* – IDE). Não é um diagrama recomendado, a menos que seja para os componentes mais importantes ou de maior complexidade.

### 3 Premissas de Software

Esta seção apresenta os pressupostos sobre os quais se baseia o desenvolvimento do software de gerenciamento da O-RU, fundamentais para garantir a funcionalidade adequada e o desempenho da unidade de rádio no contexto de um sistema Open RAN [2], conforme definido pela *Open Radio Access Network (O-RAN) Alliance*.

O M-Plane será implementado em conformidade com as especificações técnicas fornecidas no documento “Especificação do Plano de Gerenciamento do Grupo de Trabalho 4 da O-RAN (Interfaces de Fronthaul Aberto WG)”.

#### 3.1 Visão Geral do Software de Gerenciamento

No âmbito do Open RAN 5G, o software de gerenciamento (constituindo o chamado “plano de gerenciamento”) desempenha um papel crucial na orquestração eficiente e na administração abrangente dos diversos elementos de rede, incluindo as unidades de rádio (*radio unit – RU*), as unidades distribuídas (*distributed unit – DU*), unidades centralizadas (*centralized unit – CU*) e controladores inteligentes de rádio (*radio intelligent controller – RIC*). Assim, o plano de gerenciamento pode ser entendido como um conjunto estruturado de diretrizes e protocolos que coordenam a configuração e operação de cada componente de rede, promovendo a interoperabilidade, eficiência operacional e flexibilidade na implementação e gestão de soluções 5G.

Em uma rede 5G Open RAN, cada elemento é dotado de um plano de gerenciamento específico — uma abordagem granular que permite a customização e otimização direcionada para as particularidades inerentes a cada componente, maximizando a eficiência operacional do sistema como um todo. Além disso, cada plano de gerenciamento lida com aspectos específicos da configuração e operação de seus respectivos elementos. Por exemplo, o plano de gerenciamento da RU pode abranger parâmetros relacionados à interface de rádio, enquanto o da CU pode se concentrar em aspectos mais centralizados, como a alocação de recursos e a coordenação geral da rede.

No contexto do M-Plane da O-RU, o software de gerenciamento possui funcionalidades essenciais para a configuração, monitoração e otimização de desempenho da unidade. Como exemplo, uma de suas principais funções reside na administração dos parâmetros da interface de rádio, permitindo ajustes dinâmicos na potência de transmissão, modulação e configurações de *beamforming* para otimizar a qualidade e a eficiência da comunicação sem fio.

O software de gerenciamento implementado na O-RU integra um servidor NETCONF (*NETwork CONFiguration*) — um protocolo de gerenciamento largamente utilizado pela indústria de equipamentos de redes de comunicação. O gerenciamento de configurações da unidade é baseado em modelos YANG (*Yet Another Next Generation*) — uma linguagem de modelagem de dados que representa informações estruturadas em uma árvore XML (*Extensible Markup Language*).

Suas principais funções podem ser categorizadas em quatro grupos: Inicialização do Dispositivo, Gerenciamento de Software, Gerenciamento de Configuração e Gerenciamento de Falhas.

### 3.1.1 Inicialização do Dispositivo

Nesta fase é estabelecida a comunicação entre a O-RU e o sistema de gerenciamento de rede (*network management system* – NMS). Endereços da camada de transporte são configurados manualmente ou automaticamente, via *Dynamic Host Configuration Protocol* (DHCP) ou *Stateless Address Autoconfiguration* (SLAAC).

Uma função denominada “Call Home” corresponde a um processo em que a O-RU se anuncia na rede depois do seu ciclo de inicialização (carga do sistema operacional e configurações iniciais) que ocorre logo após ser energizada, estabelecendo comunicação com o NMS (através da conexão com uma DU) e respondendo a requisições NETCONF que solicitam informações acerca de suas capacidades e componentes disponíveis para gerenciamento e monitoração.

### 3.1.2 Gerenciamento de Software

Essa função permite uma administração eficiente e segura das atualizações de software, garantindo a integridade e a operação otimizada da O-RU através do controle de inventário de software e dos processos de *download*, instalação e ativação.

O controle de inventário permite ao NMS obter informações relativas às versões de software disponíveis na O-RU em diferentes espaços (*slots*) de armazenamento. O processo de *download* estabelece uma conexão com um servidor FTP (*File Transfer Protocol*) a partir da qual o pacote de atualização é baixado na unidade de rádio. No processo de instalação, o pacote baixado é descompactado e seus diferentes componentes (aplicação, firmware, etc.) são armazenados nas posições de memória correspondentes. Finalmente, no processo de ativação o sistema de inicialização (*boot*) é configurado para carregar todo o software da O-RU a partir de um determinado *slot* selecionado.

### 3.1.3 Gerenciamento de Configuração

Nesse contexto encontra-se o processo de configuração dos diversos parâmetros necessários para a operação da O-RU, bem como a monitoração (coleta) de dados operacionais que fornecem informações cruciais para análises de desempenho, manutenção preditiva e adaptação dinâmica às demandas do tráfego de rede.

### 3.1.4 Gerenciamento de Falhas

Por fim, essa função do software de gerenciamento da O-RU está relacionado ao processo de monitoração proativa de erros de comunicação, falhas na rede ou no hardware, gerando notificações para o NMS com informações fundamentais como a identificação da falha, localização, impacto nas áreas afetadas, severidade, entre outras, essenciais para uma resposta rápida e eficaz e contribuindo para a confiabilidade e estabilidade do sistema como um todo.

## 3.2 Open M-Plane

Como parte do programa *Linux Foundation Connectivity*, foi disponibilizado o software Open M-Plane [3] — uma implementação de código aberto do plano de gerenciamento Open RAN da O-RU Evenstar desenvolvida pela Meta. Esse código fornece insumos relevantes para iniciar o desenvolvimento de uma solução completa de um software de gerenciamento da unidade de rádio.

A premissa adotada no contexto do projeto OpenRAN@Brasil foi aproveitar ao máximo a implementação do Open M-Plane, determinando alguns direcionamentos de arquitetura

neste documento. O código está implementado principalmente na linguagem C++, com algumas implementações complementares em Shell Script e Lua.

### 3.3 Plataforma de Hardware

A plataforma de hardware adotada para o desenvolvimento da O-RU é baseada em um sistema em chip de radiofrequência (*radio frequency system-on-chip* – RFSoc) XCZU67DR-2FSVE1156I [4] da AMD (Xilinx). Este RFSoc apresenta um processador de quatro núcleos ARM Cortex A-53 de 64 bits projetado para funcionar como uma Unidade de Processamento de Aplicações (*application processing unit* – APU), onde o software de gerenciamento da O-RU será executado. A Figura 3 ilustra o relacionamento entre a APU e os subsistemas que implementam o projeto de referência na cadeia de processamento do dispositivo.

Dentro de cada subsistema, a maioria dos blocos é controlada por software. O software do sistema é organizado em camadas, com *drivers* de interface e acesso à memória no nível mais baixo e *drivers* dos blocos lógicos de propriedade intelectual (*IP cores*) nas camadas intermediárias. A camada superior exibe várias aplicações de sistema.

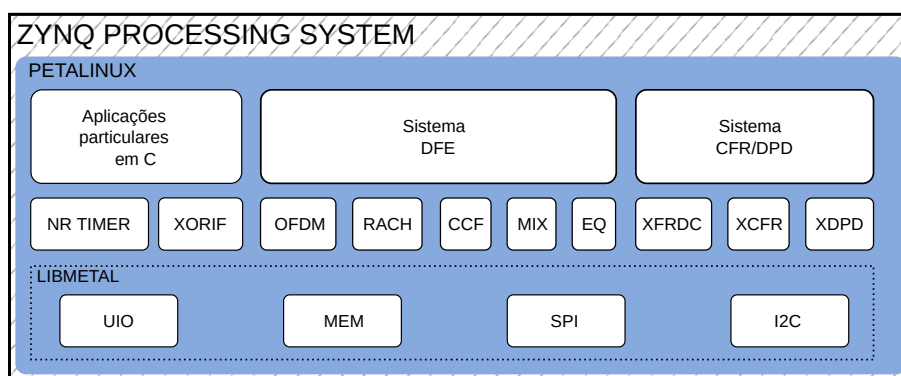


Figura 3 - Diagrama de blocos do projeto de referência destacando o sistema de processamento Zynq PS.

### 3.4 Sistema Operacional

O sistema operacional disponibilizado pela AMD para o RFSoc escolhido é o Petalinux [5]. Trata-se de uma distribuição Linux projetada especificamente para aplicações embarcadas em plataformas de hardware desse fabricante, oferecendo uma integração sólida com as ferramentas de desenvolvimento correspondentes, em particular o Vivado [6].

O Petalinux é construído com base na estrutura do Projeto Yocto para a geração de distribuições. Conseqüentemente, o processo de integração de novos pacotes de software segue as diretrizes de integração estabelecidas pelo Yocto.

### 3.5 Linguagem de Programação

O software de gerenciamento da O-RU não possui requisitos específicos de operação em tempo real ou de alto desempenho. Por outro lado, por se tratar de um sistema embarcado será preciso garantir que ele não exceda a capacidade de memória, armazenamento e processamento do hardware final da unidade. Também será necessária a interação com módulos de software de mais baixo nível, como os *device drivers*. Dado essas premissas, a linguagem de programação predominante adotada será C/C++, com abertura para a utilização de Shell Scripts em operações de software mais simples, como inicializações de

serviços. Conforme a conveniência pode ser avaliado o uso de outras linguagens de Script de forma complementar, como o Python.

### 3.6 Ambiente de Desenvolvimento

De modo geral, o ambiente de desenvolvimento deve ser baseado no conjunto de ferramentas (*toolchain*) GNU para desenvolvimento de software em C/C++, sendo complementado pelo sistema de *build cmake*. O software gerado deverá ser testado e depurado em três ambientes distintos, conforme o estágio do desenvolvimento.

O primeiro será baseado em arquitetura x86, de forma a executar a aplicação em computador pessoal (*personal computer* – PC), preferencialmente utilizando contêineres da plataforma *docker*. Nesse ambiente intermediário, o sistema operacional não será o Petalinux, mas uma distribuição padrão de Linux x86. A opção preferencial do sistema operacional em x86 seria o Ubuntu, pelo padrão utilizado no CPQD, mas a utilização da base de software do Open M-Plane levou a adoção do CentOS.

O segundo ambiente será baseado em máquina virtual QEMU (*Quick Emulation*) rodando em um PC ou servidor (x86), onde será possível emular a plataforma de processamento ARM e executar o software em um ambiente mais próximo do hardware final, já utilizando o Petalinux.

O terceiro será o próprio hardware da O-RU, sendo inicialmente implementado em um kit de avaliação (*evaluation kit* – EVK) ZCU 670 da AMD contendo o RFSoc adotado e, posteriormente, em uma amostra do hardware final da O-RU.

Conforme já mencionado, o uso do Petalinux implica a utilização do ambiente de desenvolvimento correspondente, baseado no sistema Yocto com ferramentas GNU de desenvolvimento. O sistema Yocto, por sua vez, emprega um ambiente de *build* particular, baseado no *bitbake*. A geração da *build* em Petalinux também dependerá de informações sobre o hardware alvo, geradas pela ferramenta Vivado.

## 4 Arquitetura de Software

A arquitetura do software de gerenciamento da O-RU é apresentada nas subseções seguintes a partir dos diagramas do modelo C4, proporcionando assim uma visão desde um nível mais alto até o detalhamento necessário nesse momento.

### 4.1 Contexto de Sistema

O sistema de software em desenvolvimento constitui o plano de gerenciamento da unidade de rádio (O-RU M-Plane), em conformidade com as especificações Open RAN [2] para garantir a interoperabilidade entre equipamentos de diversos fornecedores e a integração perfeita com sistemas de gerenciamento de rede. O O-RU M-Plane é um componente crítico, responsável por gerenciar eficientemente vários aspectos da operação de uma unidade de rádio. Suas funcionalidades principais incluem gerenciamento de falhas, gerenciamento de configuração, contabilidade, gerenciamento de desempenho e segurança. A Figura 4 apresenta o diagrama de contexto que ilustra a interação do sistema M-Plane com entidades externas ao software de gerenciamento.



Figura 4 - Diagrama de contexto ilustrando a interação do M-Plane com elementos externos.

No contexto da operação do O-RU M-Plane a interoperabilidade e a gestão da unidade de rádio são possibilitados por diversos sistemas de software externos. O M-Plane Controller, por exemplo, atua como um cliente do servidor NETCONF implementado na O-RU, constituindo o NMS que desempenha um papel crucial ao interagir com a unidade de rádio para realizar configurações e colocá-la em estado operacional. Além disso, o Sistema Operacional embarcado na O-RU é responsável por fornecer funcionalidades fundamentais, permitindo o acesso às interfaces (*application program interface* – API) dos blocos lógicos funcionais de hardware para a configuração dos sistemas de processamento digital de sinais e controle da parte analógica de radiofrequência (*radio frequency* – RF), além da orquestração geral de temporização.

A infraestrutura de rede é suportada por componentes adicionais como o Servidor NETCONF, composto por uma solução de código aberto formada pelos softwares *Netopeer2* e *Sysrepo*. Esses elementos são essenciais para armazenar dados de configuração e validar mensagens XML de configurações provenientes do M-Plane Controller ou de outros componentes da rede.

Por fim, os Serviços de Rede desempenham um papel crucial na oferta de recursos de conectividade, incluindo atribuição de endereços IP, resolução de nomes de domínio e controle de acesso.

As subseções seguintes apresentam uma visão detalhada os elementos internos do sistema de software M-Plane, através dos diagramas de contêineres que destacam também suas inter-relações com os componentes externos mencionados.



## 4.2 Visão de Contêineres

A visão da arquitetura do sistema de software a partir dos diagramas de contêineres do modelo C4 fornece uma compreensão clara do seu funcionamento interno. A Figura 5 ilustra um diagrama de contêineres mostrando as relações entre os principais elementos do sistema de software M-Plane.

No cerne da funcionalidade do plano de gerenciamento está uma solução de software aberto formada pelos contêineres *Netopeer2* e *Sysrepo*. O primeiro atua como um receptor de mensagens XML de algum cliente na rede (i.e. Controlador M-Plane ou NMS), realizando uma validação inicial dos dados antes de passá-los para o *Sysrepo* para uma validação mais ampla.

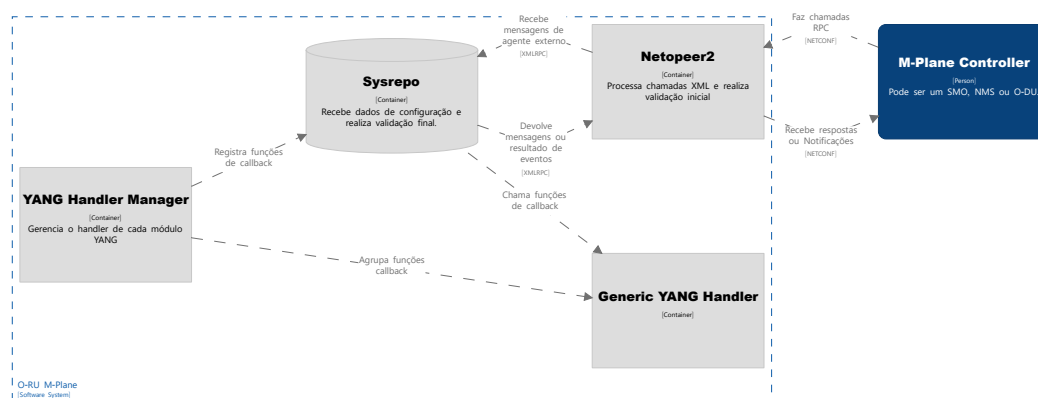


Figura 5 - Diagrama de contêineres do sistema de software M-Plane.

O contêiner *Sysrepo* recebe as informações de configuração pré-validadas do *Netopeer2*, realiza a validação final, armazena os dados de registros de configuração com segurança e dispara eventos quando necessário. No caso de um evento desencadeado dentro do sistema de armazenamento (*datastore*) do *Sysrepo*, uma função de retorno de chamada (*callback*) associada ao contexto do momento é invocada. Ela é organizada e gerenciada pelo YANG Handler Manager — uma classe que agrupa todas as funções de retorno de chamada relacionadas aos Módulos YANG registrados, os quais modelam todos os parâmetros de configuração disponíveis para a O-RU de forma padronizada, garantindo a interoperabilidade com qualquer NMS. Além disso, o YANG Handler Manager assume a responsabilidade de registrar todas as funções de retorno de chamada do YANG Handler na API do *Sysrepo*.

Nas próximas subseções serão apresentadas visões mais específicas dos contêineres para cada tópico de gerenciamento associado aos módulos YANG, os quais são tratados pelos YANG Handlers correspondentes. Adicionalmente, os *handlers* e *callbacks* estarão vinculados a funções de acesso ao sistema operacional (APIs), que por sua vez estabelecerão a conexão entre diferentes serviços responsáveis pelo tratamento pertinente a cada conjunto de módulos YANG.

### 4.2.1 YANG Handler para Gerenciamento de Usuários

A gestão de usuários da O-RU envolve o gerenciamento da autenticação, autorização e controle de acesso dos usuários que interagem com a rede. Os tratamentos específicos relacionados ao gerenciamento de usuário serão realizados pelo conjunto de elementos apresentados na Figura 6.

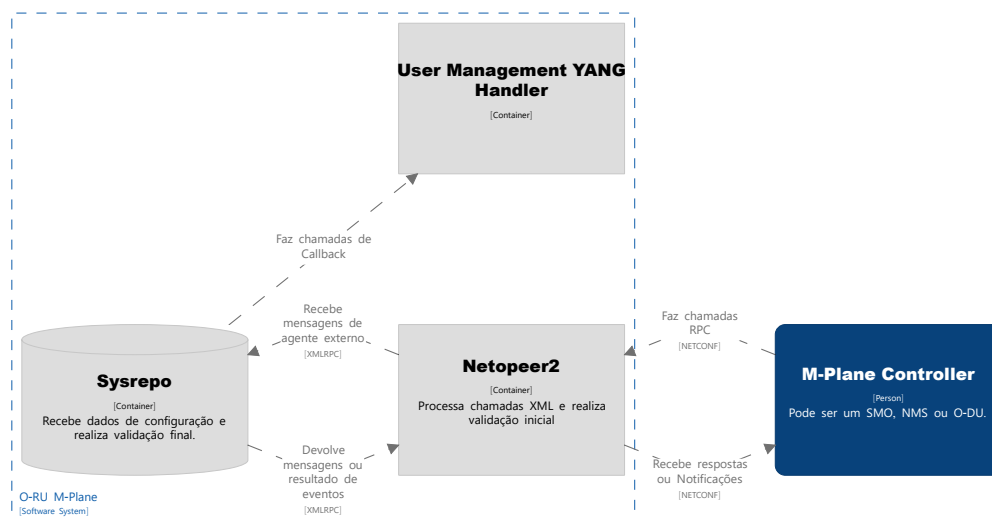


Figura 6 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de usuários.

O User Management YANG Handler é um contêiner dedicado à gestão dos agentes que utilizarão a O-RU. Este elemento é derivado das funcionalidades do módulo YANG *o-ran-usrmgmt* que define um modelo de dados para gerir contas e permissões de acesso ao equipamento. Este módulo engloba várias funcionalidades essenciais. Primeiro, trata da gestão de contas, oferecendo entidades para criar, atualizar, eliminar e consultar contas. Em segundo lugar, lida com a autorização, fornecendo entidades para definir funções de utilizador, permissões e fazer cumprir políticas de autorização. Em terceiro lugar, gera a autenticação, definindo entidades para mecanismos de autenticação de usuários e o processo de autenticação.

No contexto O-RAN o módulo YANG *o-ran-usrmgmt* serve a várias finalidades. Para operadores de rede, permite a criação de contas de utilizadores para engenheiros e técnicos de rede, concedendo-lhes as permissões apropriadas para acessar e gerir equipamentos O-RAN. Os fornecedores desses equipamentos podem aproveitar este módulo para implementar portais de gerenciamento, permitindo aos clientes criar e gerir as suas próprias contas de utilizador. Além disso, os operadores de rede podem usar este módulo para auditar a atividade de usuários da O-RU, garantindo o cumprimento de políticas de segurança rigorosas.

### 4.2.2 YANG Handler para Gerenciamento de Software

A função de gerenciamento de software é controlada pelo Software Management YANG Handler, permitindo uma administração eficiente e segura das atualizações de software da O-RU. A Figura 7 apresenta um diagrama mostrando os elementos relacionados a este serviço.

A O-RU apresenta um conjunto de espaços de armazenamento (*slots*) designados para diferentes versões do seu software, de acordo com a capacidade do dispositivo. Um número mínimo de dois *slots* graváveis é essencial para operações seguras de atualização.

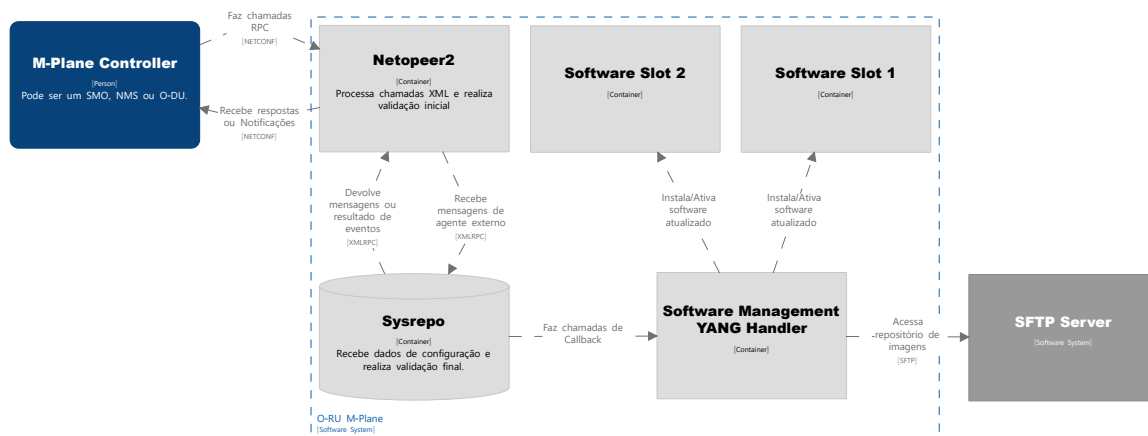


Figura 7 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de software.

O Software Manager YANG Handler desempenha um papel crucial nesse processo, supervisionando ações no inventário de software. Ele gerencia no mínimo dois *slots* para diferentes versões de software da O-RU. Este contêiner implementa soluções que possibilitam o *download*, instalação e ativação de versões de software armazenadas nos diferentes *slots*.

As operações de instalação e ativação interagem com o sistema operacional, utilizando recursos de ativação de firmware, modificando o comportamento de inicialização, manipulando arquivos binários carregados no FPGA (*Field Programmable Gate Array*) e alternando entre imagens do sistema de arquivos. Vale ressaltar que a ativação do software não implica automaticamente o uso operacional da nova versão ativada na O-RU. Dependendo do componente ou sistema de software atualizado pode ser necessária a reinicialização da unidade de rádio.

### 4.2.3 YANG Handler para Gerenciamento do Plano de Sincronismo

Os tratamentos específicos relacionados ao gerenciamento do plano de sincronismo serão realizados pelo conjunto de elementos apresentados na Figura 8.

O módulo YANG *o-ran-sync* é utilizado pelo M-Plane da O-RU para implementar funcionalidades de sincronização, permitindo a configuração e monitoração de todo o processo a partir do contêiner Sync YANG Handler. Dessa forma, definem-se mecanismos padronizados para que o relógio interno da O-RU seja sincronizado com os demais equipamentos da rede a partir de um servidor PTP (*Precision Time Protocol*) mestre.

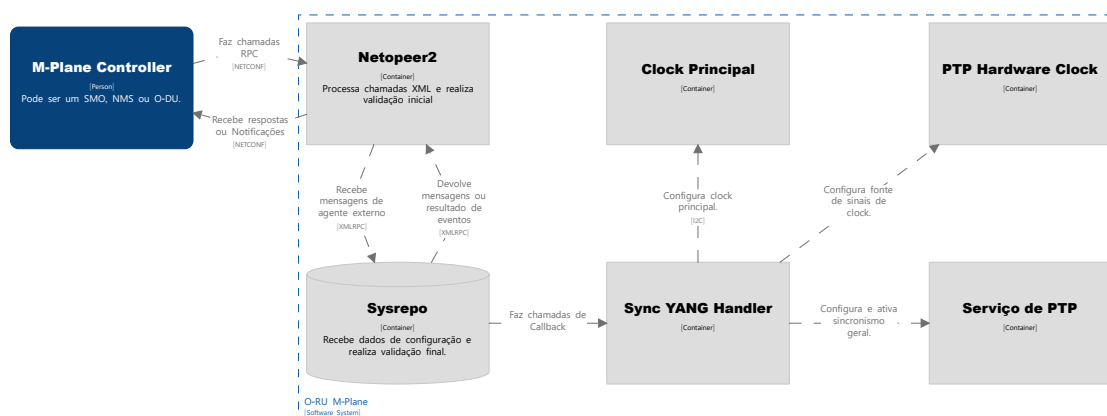


Figura 8 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de sincronismo.

O Sync YANG Handler é responsável por gerenciar as seguintes funcionalidades:

- Configuração de Sincronização: contém parâmetros de configuração para o mecanismo de sincronização;
- Estado de Sincronização: contém informações sobre o processo de sincronização;
- Destinos de Sincronização: contém informações sobre os equipamentos O-RAN com os quais o equipamento local está sincronizado.

O módulo YANG *o-ran-sync* também contém os seguintes parâmetros principais:

- Modo de Sincronização: indica o modo de sincronização que o equipamento local está utilizando;
- Temporizador de Sincronização: especifica o intervalo no qual o equipamento local sincronizará seu estado com os destinos de sincronização;
- Estado de Sincronização: indica o estado do processo de sincronização;
- Erro de Sincronização: indica o erro que ocorreu durante o processo de sincronização, se houver algum.

#### 4.2.4 YANG Handler para Gerenciamento do Plano de Usuário

Os tratamentos específicos relacionados ao gerenciamento do plano de usuário serão realizados pelo conjunto de elementos apresentados na Figura 9. O módulo YANG *o-ran-uplane-conf* define os parâmetros de configuração para o plano de usuário (*user plane* – U-Plane) da O-RU. O contêiner U-Plane YANG Handler gerenciará os seguintes aspectos fundamentais de configuração de uma O-RU:

- *Static Low Level Receiver (RX) Endpoints*: contém uma lista de *static low-level RX endpoints*;
- *Low Level Transmitter (TX) Endpoints*: contém uma lista de *low-level TX endpoints*;
- *Compression*: contém parâmetros configuráveis globalmente para a compressão.

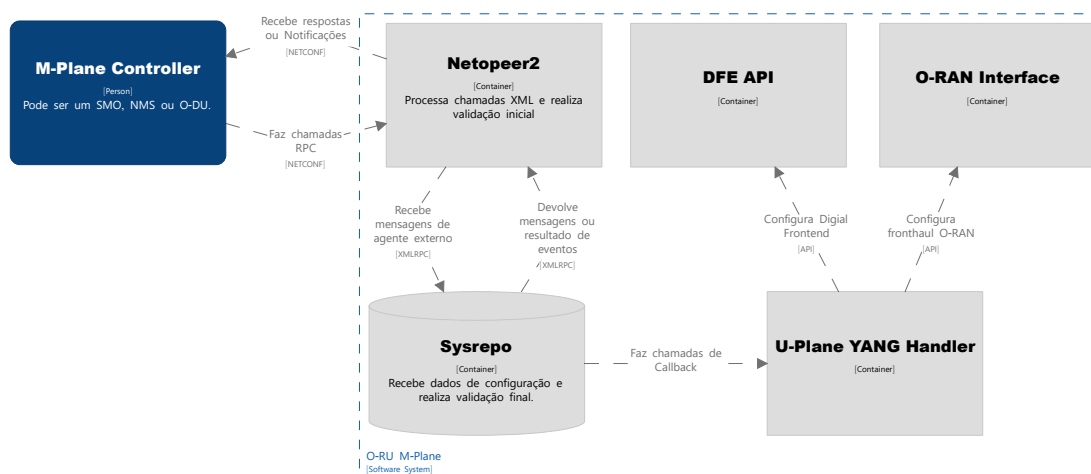


Figura 9 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento do plano de usuário.

Os elementos de *Low Level Endpoints* na O-RAN são as interfaces físicas em uma O-RU, usadas para transmitir e receber dados da antena. Esses elementos podem ser implementados usando uma variedade de tecnologias, como interfaces ópticas, elétricas ou de RF. São tipicamente usados para transmitir e receber dados em uma forma bruta (e.g. formas de onda ou símbolos de uma constelação específica).

Esses dados podem ser posteriormente processados pelo software da O-RU para extrair as informações desejadas, como dados do usuário ou sinais de controle. O campo *Compression* é um parâmetro relacionado à configuração do algoritmo de compressão e aos parâmetros usados para comprimir dados transmitidos à O-RU e à O-DU.

Dentro deste contêiner existem as configurações para tipo e nível da compressão desejada. Isso pode ajudar a melhorar o desempenho e a eficiência da O-RU e a reduzir a quantidade de dados que precisa ser transmitida no U-Plane. O módulo YANG *o-ran-uplane-conf* também contém as seguintes informações obrigatórias:

- *Downlink Radio Frame Offset*: especifica o deslocamento da posição inicial do quadro de rádio de 10ms;
- *Downlink SFN Offset*: especifica o deslocamento do valor SFN.

#### 4.2.5 YANG Handler para Gerenciamento de Falhas

O módulo YANG *o-ran-fm* descreve e estrutura parâmetros relacionados ao gerenciamento de falhas para a O-RU. O M-Plane possui um sistema de coleta de dados de falhas de diferentes elementos da unidade de rádio. A Figura 10 ilustra um diagrama de contêineres e suas relações que implementam o gerenciamento de falhas.

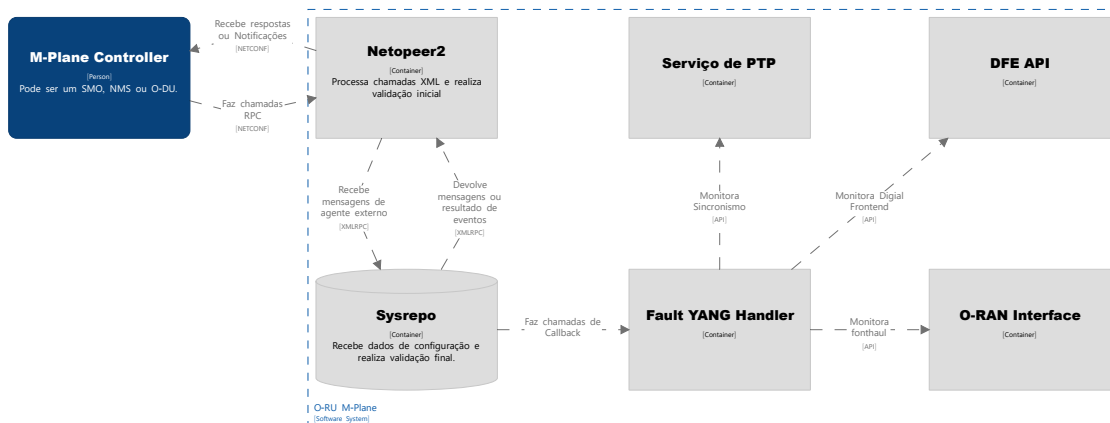


Figura 10 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de falhas.

O Fault YANG Handler analisa esses dados para identificar falhas e suas causas raízes, dentro dos seguintes aspectos:

- Falhas de hardware: como de fonte de alimentação, de CPU e de memória;
- Falhas de software: como de firmware, de *driver* e do sistema operacional;
- Falhas de configuração: relacionadas à configurações incorretas e arquivos de configuração ausentes;
- Falhas de desempenho: como alta utilização da CPU ou da memória e alta latência.

Os dados coletados são utilizados para:

- Gerar alarmes quando as falhas ocorrem;
- Isolar a causa raiz das falhas;
- Recomendar ações corretivas para resolver falhas;
- Rastrear o status das falhas e sua resolução;
- Gerar relatórios sobre falhas e sua resolução.

O Fault YANG Handler também implementa mecanismos de tolerância a falhas. Por exemplo, na situação de falha geral o sistema pode alternar automaticamente para uma versão de software de backup se a principal falhar. A seguir estão listados alguns exemplos específicos de como o sistema de gerenciamento de falhas pode ser usado na O-RU:

- Gerar alarmes quando uma O-RU experimenta uma falha na fonte de alimentação. O operador pode então investigar o alarme para determinar a causa da falha e tomar medidas corretivas;

- Isolar a causa raiz de uma falha de firmware. O operador pode então usar essas informações para baixar uma nova imagem de firmware para a unidade de rádio;
- Recomendar ações corretivas para resolver uma falha de desempenho. Por exemplo, o módulo pode recomendar aumentar os recursos de CPU ou memória alocados à O-RU;
- Rastrear o status de uma falha até que ela seja resolvida. O operador pode então usar essas informações para garantir que a falha seja tratada com rapidez e eficiência;
- Gerar um relatório sobre as falhas que ocorreram em uma O-RU ao longo de um período de tempo. Esse relatório pode ser usado pelo operador para identificar tendências e padrões em falhas.

#### 4.2.6 YANG Handler para Gerenciamento de Desempenho

Os tratamentos específicos relacionados ao gerenciamento de desempenho serão realizados pelo conjunto de elementos apresentados na Figura 11. O módulo YANG *o-ran-performance-manager* descreve parâmetros para uma visão unificada do desempenho de uma O-RU. O sistema coleta dados de diferentes elementos da unidade de rádio, os quais são analisados para identificar gargalos e tendências de desempenho. As principais métricas analisadas pelo Performance YANG Handler em uma O-RU são:

- Desempenho de gerenciamento de recursos de rádio: inclui parâmetros como utilização de canal, taxa de sucesso de *handover* e níveis de interferência;
- Desempenho de seleção de *Modulation Code Scheme* (MCS): inclui parâmetros como a porcentagem do tempo que a O-RU está usando o MCS ideal para as condições de rádio atuais;
- Desempenho da camada física: inclui parâmetros como taxa de erro de bits (BER), relação sinal-ruído (*signal-to-noise ratio* – SNR) e potência de transmissão;
- Desempenho da camada de transporte: inclui parâmetros como latência, *jitter* e perda de pacotes.

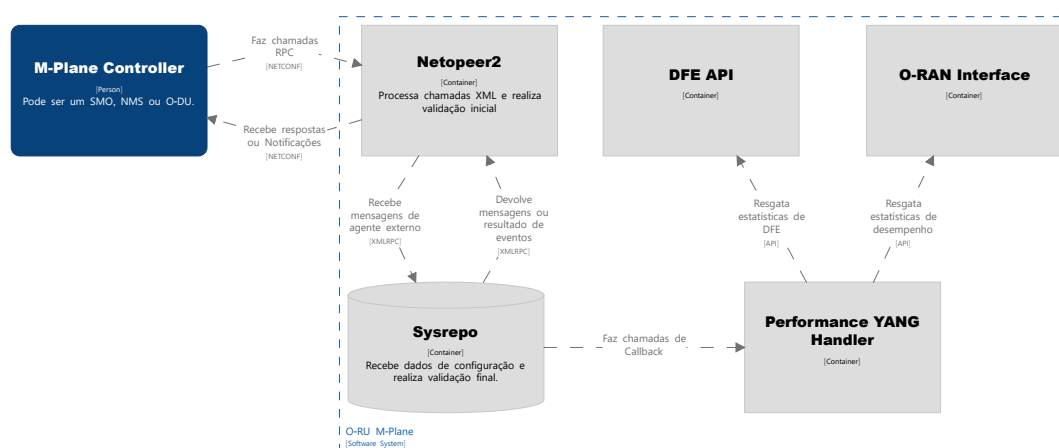


Figura 11 - Diagrama de contêineres ilustrando elementos relativos ao gerenciamento de desempenho.

Os dados pertencentes às métricas citadas são utilizados para determinar aspectos de desempenho. É possível, por exemplo, otimizar o funcionamento da O-RU para diferentes tipos de tráfego e grupos de usuários. Além disso, a análise de dados pode fornecer uma previsão das tendências de desempenho e auxiliar na realização de ajustes na rede.

Exemplos específicos de como as métricas podem ser usadas para gerenciar o desempenho em uma O-RU são:

- Identificar O-RUs específicas que estão experimentando altas taxas de falha de *handover*. O operador pode então investigar essas unidades para determinar a causa do problema e tomar medidas corretivas;
- Identificar O-RUs que não estão usando o MCS ideal para as condições de rádio atuais. O operador pode então usar essas informações para otimizar o algoritmo de seleção de MCS para essas RUs;



- Identificar O-RUs que estão experimentando alta latência ou perda de pacotes na camada de transporte. O operador pode então investigar essas unidades para determinar a causa do problema e tomar medidas corretivas.

## 5 Referências

- [1] "C4 model for software architecture", <http://www.c4model.com>, acessado em 25/01/2024.
- [2] "O-RAN Alliance Working Group 4 Management Plane Specification", O-RAN Alliance, Especificação de sistema.
- [3] "Meta open m-plane repository", <https://github.com/lf-connectivity/open-mplane>, acessado em 25/01/2024.
- [4] "Zynq ultrascale+ rfsoc data sheet: Overview (ds889)", <https://docs.xilinx.com/v/u/en-US/ds889-zynq-usp-rfsoc-overview>, acessado em 25/01/2024.
- [5] "Petalinux tools documentation: Reference guide (ug1144)", <https://docs.xilinx.com/r/en-US/ug1144-petalinux-tools-reference-guide/Overview>, acessado em 25/01/2024.
- [6] "Vivado design suite", <https://www.xilinx.com/products/design-tools/vivado.html>, acessado em 25/01/2024.

## 6 Histórico de Alterações do Documento Consolidado

Data de emissão	Versão	Descrições das alterações realizadas
04/Mar/24	AA	Versão inicial.

## 7 Execução e Aprovação

**Elaborado por:**

Claudio Ferreira Dias

Eduardo Mobilon

Fundação CPQD – Centro de Pesquisa e Desenvolvimento em Telecomunicações

**Revisado por:**

Eduardo Mobilon

Fundação CPQD – Centro de Pesquisa e Desenvolvimento em Telecomunicações

**Aprovado por:**

RNP – Rede Nacional de Ensino e Pesquisa