DM

Building a Network Operations Center (NOC) Solution

MASTER DISSERTATION

Sebastião Rúben Henriques de Sousa
MASTER IN INFORMATICS ENGINEERING

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

February | 2016

Building a Network Operations Center (NOC) Solution

MASTER DISSERTATION

Sebastião Rúben Henriques de Sousa
MASTER IN INFORMATICS ENGINEERING

SUPERVISOR
Eduardo Miguel Dias Marques

CO-SUPERVISOR
Lina Maria Pestana Leão de Brito

Resumo

Esta dissertação contextualiza e descreve a actividade prática do quotidiano de um gestor de redes informáticas modernas. Define *Network Operations Center* (NOC) e a sua importância como provedor de serviços de Internet, e aborda a gestão de redes como alicerce teórico mediante a tomada de decisões que a tarefa de gestão coloca.

É com base na complexidade da tarefa de gestão que surge a necessidade e o aparecimento consequente de Plataformas de Gestão. Estas plataformas não só auxiliam na tarefa para a qual foram desenvolvidas como substituem todo um conjunto de ferramentas tradicionais, limitadas para a realidade das infra-estruturas de telecomunicações actuais.

O estudo do NOC da NOS Madeira enfatiza a problemática que as infra-estruturas de média e grande dimensão comportam para os gestores de redes que lá operam. As limitações da plataforma de gestão em uso referentes à definição e organização de utilizadores, dispositivos e grupos respectivos, conjuntamente com a falta de parametrização em notificações justificam uma solução.

Aqui, é feita uma análise e um levantamento de requisitos para a selecção da nova plataforma de gestão seguida da proposta de uma solução integrada para resolução do problema. Esta proposta envolve a definição de uma arquitectura, o desenho e a implementação de *software* próprios para um sistema particular ao qual designamos: ZenDash.

Palavras-chave: Gestão de Redes, Plataformas de Gestão, Centro de Operações de Rede

Abstract

This dissertation serves to contextualize and report the daily life of a manager of modern computer networks. It defines a *Network Operations Center* (NOC) and its importance as an Internet service provider, and addresses network management as the theoretical basis to improve the decision making process settled by the management task itself.

It is based on the complexity of the management task that comes the need and consequent forthcoming of the modern Network Management Systems. These systems do not only assist on the task they were developed for as they replace a whole set of standard tools, limited to the reality of the modern telecommunications infrastructures.

The study of the NOC from NOS Madeira emphasizes the issue that medium and large infrastructures cause to their managers. The limitations regarding user definition, user organization, device definition and group of both users and devices with the system they use, along with the lack of notification parameterization justifies a solution.

Here, an analysis and a requirements gathering takes place to pick a new system followed by a proposal of an integrated solution to solve the problem. This proposal encompasses a proper architecture definition, proper software design and implementation for a particular system that we designate: ZenDash.

Keywords: Network Management, Network Management Systems, Network Operations Center

Agradecimentos

Aproveito o tópico para deixar uma palavra de apreço e agradecimento a todos os que, de uma forma ou de outra, estiveram ao meu lado durante toda ou parte desta caminhada.

Quero agradecer, em primeiro lugar, aos orientadores Eduardo Miguel Dias Marques e Lina Maria Pestana Leão de Brito pela oportunidade de aprendizagem, pela confiança depositada e pelo esclarecimento de dúvidas oportuno sempre que solicitado. Foram incansáveis e imprescindíveis à formação exigida, evolução e concretização de cada uma das etapas e desafios colocados por esta dissertação.

Agradeço ao Sr. Nélio Vieira, ao David Sousa e à restante comitiva da NOS Madeira pelo suporte, acompanhamento e material disponibilizado para a realização deste projecto. Agradeço a discussão e a aprovação da proposta feita, e a oportunidade de contribuição para o desenvolvimento de uma solução integrada moderna.

Obrigado a todos os amigos que de alguma forma acompanharam o trabalho desenvolvido, que ajudaram e marcaram presença com contribuição e troca de ideias sempre que pertinente.

Um agradecimento especial aos pais e irmãos pelo suporte psicológico e emocional, e pelo suporte financeiro que proporcionou toda esta aventura. Estou eternamente grato pela disponibilidade e compreensão nos momentos difíceis, pela força e confiança transmitidos.

Um muito obrigado à namorada pelo espírito crítico e construtivo, substancial ao crescimento pessoal exigido no decorrer desta etapa. Agradeço a presença, a sensibilidade, o humor, a sinceridade, a compreensão e a confiança constante em mim depositada.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AJAX	*Asynchronous JavaScript And XML*
API	*Application Programming Interface*
ASN.1	*Abstract Syntax Notation One*
BER	*Basic Encoding Rules*
CARP	*Common Address Redundancy Protocol*
CCITT	*Consultative Committee for International Telephony and Telegraphy*
CFM	*Configuration File Management*
CIM	*Common Information Model*
CIMOM	*CIM Object Manager*
CMDB	*Configuration Management DataBase*
CMIP	*Common Management Information Protocol*
CMIS	*Common Management Information Service*
CMISE	*CMIS Elements*
CMTS	*Cable Modem Termination System*
CPU	*Central Processing Unit*
CSS	*Cascading Style Sheets*
CSV	*Comma-Separated Values*
DDNS	*Dynamic DNS*
DHCP	*Dynamic Host Configuration Protocol*
DMTF	*Distributed Management Task Force*
DNS	*Domain Name System*
DOCSIS	*Data Over Cable Service Interface Specification*

DPI	*Deep Packet Inspection*
DQL	*Doctrine Query Language*
ESXi	*Elastic Sky X integrated*
FTTH	*Fiber To The Home*
GGC	*Google Global Cache*
GNU	*GNU's Not Unix*
GPL	*GNU General Public License*
GUI	*Graphical UI*
HFC	*Hybrid Fiber Coaxial*
HMMP	*HyperMedia Management Protocol*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
HTTPS	*HTTP Secure*
ICMP	*Internet Control Message Protocol*
iDRAC	*integrated Dell Remote Access Controller*
IMAP	*Internet Message Access Protocol*
IP	*Internet Protocol*
IRC	*Internet Relay Chat*
ISO	*International Organization for Standardization*
ITIL	*Information Technology Infrastructure Library*
ITU	*International Telecommunication Union*
ITU-T	*ITU Telecommunication Standardization Sector*
JSON	*JavaScript Object Notation*
LDAP	*Lightweight Directory Access Protocol*
LE	*Layer Entity*
LME	*Layer Management Entity*
MIB	*Management Information Base*

MPS	*Multimedia Provisioning System*
MX	*Mail eXchanger*
NAT	*Network Address Translation*
NOC	*Network Operations Center*
OID	*Object IDentifier*
OOP	*Object-Oriented Programming*
OSI	*Open Systems Interconnection*
PC	*PacketCable*
PDU	*Protocol Data Unit*
PHP	*PHP: Hypertext Preprocessor*
POP3	*Post Office Protocol 3*
PRTG	*Paessler Router Traffic Grapher*
QoS	Quality of Service
REST	*REpresentational State Transfer*
RFC	*Request For Comments*
RMON	*Remote Network MONitoring*
RRD	*Round-Robin Database*
SCP	*Secure CoPy*
SGMP	*Simple Gateway Monitoring Protocol*
SIP	*Session Initiation Protocol*
SLA	*Service-Level Agreement*
SMAE	*System Management Application Entities*
SMAP	*System Management Application Processes*
SMF	*System Management Functions*
SMI	*Structure of Management Information*
SMS	*Short Message Service*
SMTP	*Simple Mail Transfer Protocol*

SNMP	*Simple Network Management Protocol*
SNPP	*Simple Network Paging Protocol*
SOAP	*Simple Object Access Protocol*
SQL	*Structured Query Language*
SSH	*Secure SHell*
SSL	*Secure Sockets Layer*
SVG	*Scalable Vector Graphics*
TCP	*Transmission Control Protocol*
TFTP	*Trivial File Transfer Protocol*
TMN	*Telecommunications Management Network*
TSDB	*Time Series DataBase*
UDP	*User Datagram Protocol*
UI	*User Interface*
UPS	*Uninterruptible Power Supply*
VM	*Virtual Machine*
VoD	*Video on Demand*
VoIP	*Voice over IP*
WBEM	*Web-Based Enterprise Management*
WMI	*Windows Management Instrumentation*
XML	*eXtensible Markup Language*
YAML	*YAML Ain't Markup Language*
ZODB	*Zope Object DataBase*

Capítulo I

Introdução

As redes informáticas norteiam cada vez mais o sucesso empresarial, independentemente da dimensão do negócio e da complexidade das infra-estruturas que as suportam. Quando uma rede falha, os clientes por ela abrangidos deixam de poder comunicar. Esta situação implica custos e redução de produtividade, e é quando um NOC se torna necessário.

Um NOC contém equipamentos para prevenir interrupções de serviço indesejadas, e garantir o desempenho exigido. Recorre a plataformas de gestão de redes que auxiliam a tarefa de gestão com monitorização, detecção e correcção de situações indevidas. Estas plataformas permitem aos utilizadores a centralização de eventos, a configuração de alarmes, a criação de relatórios automáticos para ajudar no despiste de erros, entre outros [1] [2].

O NOC da NOS Madeira usufrui, contudo, de uma plataforma de gestão antiga com vários problemas técnicos, limitada tecnologicamente. É com base nesta problemática e limitações que se pretende fazer um levantamento compreensivo das necessidades dos gestores de rede desta infra-estrutura, ajustá-las, propor e seleccionar uma de várias soluções alternativas e gratuitas que existem no mercado.

A solução eleita deve comportar uma plataforma de gestão escalável, robusta e o mais completa possível dentro de um conjunto de soluções bem definido. O desenvolvimento de componentes adicionais visa cobrir questões de redundância propostas internamente para tornar a solução e plataforma finais o mais altamente disponíveis, preparadas para a ocorrência de situações imprevistas referidas ao longo das reuniões com os gestores de rede da NOS Madeira.

Este capítulo descreve a dissertação, o propósito e o âmbito do projecto desenvolvido. Enumera os objectivos delineados para médio e longo prazos, e apresenta a metodologia de trabalho adoptada a cumprir com a equipa de gestão da infra-estrutura de telecomunicações da NOS Madeira. Sintetiza os capítulos e a distribuição respectiva ao longo do documento.

1.1 Contexto

Esta dissertação assenta na área da gestão de redes e sistemas, mais precisamente no âmbito da monitorização. Aqui, são descritas as plataformas de gestão e monitorização como solução integrada desenhada para dar resposta às necessidades dos gestores de rede e das infra-estruturas de telecomunicações modernas.

São vários os documentos que contribuem vivamente para o estado da arte na área da gestão e monitorização de redes e sistemas, e alguns os que se dedicam à exploração e detalhe de requisitos e funcionalidades basilares às plataformas de gestão mais recorrentes.

É com base nestes documentos que se pretende reunir informação especificamente relevante para contextualizar o leitor, introduzir conceitos fundamentais e levá-lo a compreender o porquê da gestão e monitorização de redes e sistemas, a sua necessidade, utilidade e os desafios mais relevantes da área tirando partido de duas realidades distintas: o mundo académico e o mundo real (ou mercado de trabalho).

1.2 Objectivos

O objectivo preliminar comporta o estudo da infra-estrutura de telecomunicações da NOS Madeira. É pretendido o levantamento de utilizadores, perfis de utilizadores, grupos de utilizadores, dispositivos, grupos de dispositivos, sistemas, processos internos chave e a alarmística respectiva.

O objectivo principal é, efectivamente, aprimorar todo o processo de gestão de redes com monitorização e alarmística necessárias. Isto implica pesquisar, analisar e comparar plataformas de gestão vigentes, e seleccionar a plataforma que melhor se enquadra no cenário descrito.

Outro objectivo assenta no teste da nova plataforma de gestão. Aqui, são frequentes as reuniões calendarizadas com os gestores de rede respectivos para verificar o cumprimento ou não cumprimento dos requisitos inicialmente propostos.

1.3 Metodologia

A metodologia adoptada para o desafio proposto pela presente dissertação está dividida em 4 passos:

1. Estudar a infra-estrutura de telecomunicações da NOS Madeira tendo por base a documentação e as reuniões com os gestores de rede devidos
2. Pesquisar, analisar e comparar plataformas de gestão conforme as características da rede, e seleccionar a que melhor se enquadra nos desafios de gestão colocados
3. Implementar a solução de gestão integrada, por fim, com componentes desenvolvidos para responder às questões de alta disponibilidade propostas internamente
4. Testar a solução implementada e aperfeiçoar parâmetros de configuração de acordo com os resultados obtidos

1.4 Organização

A presente dissertação é essencialmente composta por 7 capítulos que cobrem, da teoria à prática, o trabalho desenvolvido.

O primeiro capítulo contextualiza o tema, apresenta os objectivos estipulados para médio e longo prazos, e indica a metodologia adoptada para conclusão da dissertação.

O segundo capítulo comporta o estado da arte da gestão de redes, e aglomera aspectos teóricos que vão desde as funções de gestão às arquitecturas de gestão de redes.

O terceiro capítulo enumera e compara 4 plataformas de gestão *open-source* amplamente conhecidas.

O quarto capítulo comporta o caso de estudo NOS Madeira, descreve a organização interna da empresa e o problema experienciado. Analisa e compara as 4 plataformas de gestão *open-source* abordadas em termos de requisitos e funcionalidades discutidos, e selecciona a plataforma de gestão que melhor se enquadra no caso em questão.

O quinto capítulo indica quais as vantagens da plataforma de gestão seleccionada, onde e como é que esta resolve os problemas vividos.

O sexto capítulo apresenta e descreve o projecto prático desenvolvido para dar resposta à questão de redundância proposta internamente. Cobre os requisitos, a

arquitectura, o desenho, a implementação, o teste e depuração, a distribuição e manutenção deste.

O sétimo capítulo contém a síntese das sínteses dos vários capítulos apresentados, descreve pontos de reflexão e apresenta direcções futuras para trabalho igualmente futuro.

Capítulo II

Gestão de Redes

A Gestão de Redes é, na sua essência, um serviço que reúne todo um conjunto de ferramentas, aplicações e dispositivos que auxiliam os gestores de rede na tarefa de gestão, monitorização e manutenção de uma rede informática. A interligação de dispositivos em rede implica a comunicação entre eles, mesmo quando heterogéneos e/ou de fabricantes distintos.

Toda esta interoperabilidade requer normas sólidas, definidas por modelos de referência de redes amplamente conhecidos, como é o caso do modelo *Open Systems Interconnection* (OSI). Estes modelos especificam como é feita a comunicação entre dispositivos, e descrevem os mecanismos usados na troca de informação.

Foi no início da década de 80 que as empresas se aperceberam dos ganhos de produtividade trazidos com as novas tecnologias de informação, e com os produtos para elas desenhados. A criação e o planeamento estratégico de redes heterogéneas a ritmo acelerado acabou, contudo, por complicar as operações de gestão de redes do quotidiano, com emergência para um novo conceito: A Gestão de Redes Integrada e Automatizada [3] [4] [5].

A Gestão de Redes Integrada e Automatizada surge para dar resposta aos problemas dos gestores de redes modernas, com plataformas de gestão que permitem substituir um conjunto de ferramentas tradicionais. Estas plataformas tratam de tecnologias complexas e previnem interrupções de serviço inesperadas, na tentativa de garantir a *Quality of Service* (QoS) e os níveis de desempenho exigidos pelos utilizadores.

Este capítulo apresenta e descreve os aspectos fundamentais da Gestão de Redes, das funções de gestão à gestão de redes baseada na Web. Indica os modelos de referência e as arquitecturas de gestão respectivas para que o leitor reúna informação suficientemente detalhada a ponto de contextualizar, perceber o propósito, a interligação e as aplicabilidades práticas de cada um dos tópicos abordados.

2.1 Funções de Gestão

Foi já em meados dos anos 80 que surgiu o modelo de gestão de redes denominado FCAPS [6]. O termo foi apresentado pela *International Telecommunication Union Telecommunication Standardization Sector* (ITU-T), ex-*Consultative Committee for International Telephony and Telegraphy* (CCITT), para auxílio na gestão de redes de telecomunicações e como acrónimo para as 5 funções de gestão padronizadas pela *International Organization for Standardization* (ISO) na gestão de redes de dados: *Fault*, *Configuration*, *Accounting*, *Performance* e *Security* como representado na Figura 2.1.

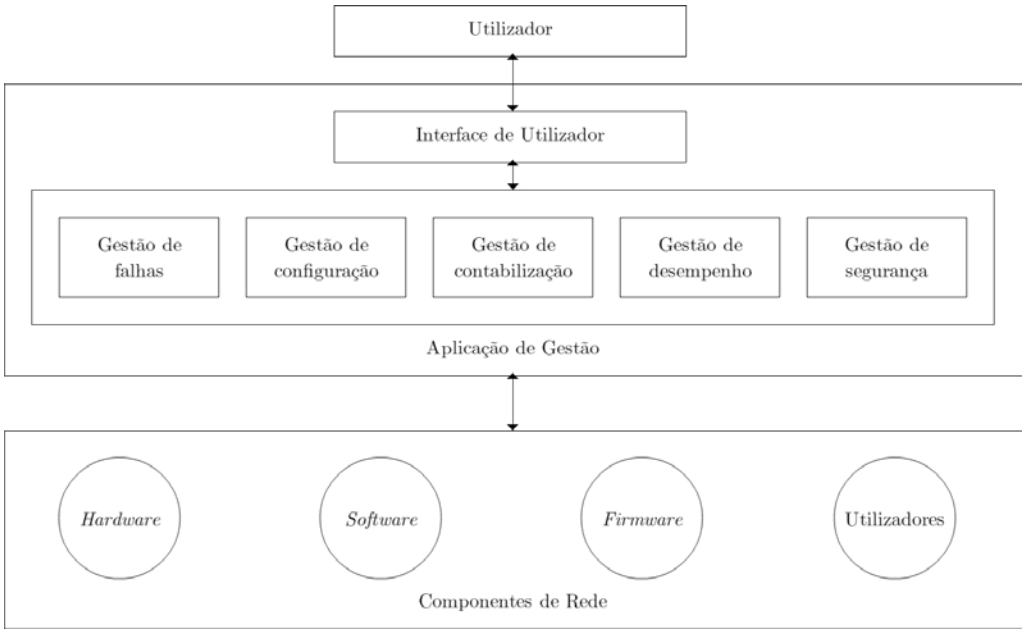


Figura 2.1: Componentes de aplicação de gestão de redes (baseada em [7])

É através das funções de gestão, e da interacção entre elas, que se classifica a informação recolhida pela Aplicação de Gestão. Esta informação, proveniente de Componentes de Rede tais como *Hardware* e *Software*, é analisada e apresentada na Interface de Utilizador mediante a configuração estipulada pelo Utilizador respectivo.

As responsabilidades das funções de gestão estão distribuídas de acordo com o âmbito de cada área funcional, descritas em pormenor nas secções que se seguem.

2.1.1 Gestão de Falhas

A Gestão de Falhas (*Fault Management*) é uma das áreas funcionais mais importantes na gestão de redes. É essencialmente responsável por detectar problemas, registar, notificar os gestores de rede, e recuperar de erros tanto quanto possível [8].

A detecção de erros é feita através dos eventos despoletados pelos dispositivos monitorizados, como *triggers*[1] que indicam a violação de *thresholds*[2]. Estes erros são registados para diagnóstico futuro.

É com o diagnóstico e a análise de erros que se determinam causas, e que se correlacionam situações idênticas. Os erros menores requerem, na maior parte dos casos, modificações menores na configuração ou substituição de recursos afectados. Os erros maiores justificam, na maior parte dos casos, a definição de *triggers* e notificações para a intervenção devida do gestor de rede responsável pelo componente/dispositivo envolvido.

2.1.2 Gestão de Configuração

A Gestão de Configuração (*Configuration Management*) é responsável pela recolha, monitorização e modificação de informação de configuração do sistema. É através desta informação que se descreve a topologia de rede com minúcia, os dispositivos, a localização, a cablagem, as ligações lógicas, o *hardware* e o *software* dos recursos.

Os dados recolhidos são armazenados em base(s) de dados, de acesso fácil, sobre as quais podemos pesquisar e encontrar dicas para resolução de problemas recorrentes.

2.1.3 Gestão de Contabilização

A Gestão de Contabilização (*Accounting Management*) assume um papel fundamental nas redes comerciais com registo de utilização de recursos. Os utilizadores e grupos de utilizadores que usufruam destes e/ou de outros serviços restritos são, portanto, taxados por isso.

[1]Configuração para avaliar situações específicas e despoletar acções como enviar notificações ou executar comandos remotamente.
[2]Valor mínimo ou máximo admitido para um ou mais dados de gestão monitorizados ao longo do tempo.

É com base no registo efectuado que se encontram padrões de utilização de recursos, e que se estabelecem políticas e quotas de utilização para os utilizadores e grupos de utilizadores envolvidos.

A auditoria das ligações dos utilizadores aos recursos de rede ajudam a classificar a correcta utilização destes, e a reflectir os custos de crescimento/actualização da rede.

2.1.4 Gestão de Desempenho

A Gestão de Desempenho (*Performance Management*) é responsável pela recolha, análise e tratamento de informação dos dispositivos monitorizados. Os dados obtidos ajudam na detecção de anomalias, na previsão de comportamentos e no processo de tomada de decisões.

Auxilia no processo de configuração, na gestão de falhas, no planeamento de infra-estruturas e reflecte, na verdade, a monitorização dos dispositivos de rede e a determinação de *thresholds* apropriados para os mesmos [3].

2.1.5 Gestão de Segurança

A Gestão de Segurança (*Security Management*) enfatiza a monitorização e o controlo dos mecanismos de segurança praticados pelos sistemas de gestão. Estes sistemas devem minimizar os acessos ilícitos, promover a confidencialidade e a integridade dos dados registados [9] [10].

Aqui, há toda uma preocupação com a definição de utilizadores, grupos de utilizadores, permissões e controlo de acesso aos recursos de rede. O levantamento dos requisitos de segurança ajuda na configuração e atribuição de permissões adequadas, na monitorização e registo de situações particularmente relevantes sobre recursos de maior confidencialidade.

2.2 Arquitecturas de Gestão de Redes

São várias as arquitecturas de gestão de redes desenvolvidas até ao momento. Aqui, é levantado um conjunto de aspectos importantes referente às arquitecturas de gestão OSI, TCP/IP, TMN e arquitectura de gestão baseada na Web. A aplicação de conceitos explorados, conjuntamente com a arquitectura de gestão TCP/IP, são a base para o trabalho que se pretende desenvolver.

Uma arquitectura de gestão de redes é essencialmente composta pelos elementos fundamentais: Sistema Gestor e Sistema Gerido. A arquitectura de gestão consolida o modelo Gestor-Agente, descreve a interligação entre os elementos e detalha o método de comunicação levado a cabo pelo Protocolo de Gestão, como mostra a Figura 2.2.

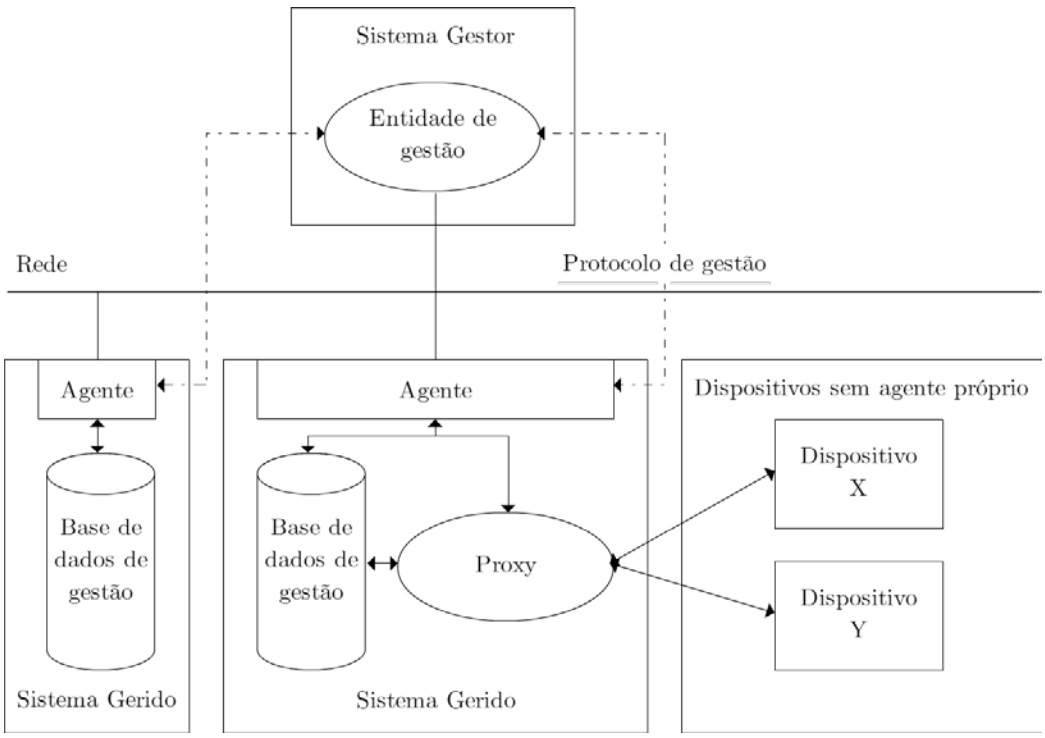


Figura 2.2: Modelo de gestão Gestor-Agente (baseada em [3] e [10])

O Sistema Gestor é a entidade de gestão que pede dados de interesse aos sistemas geridos [10]. O pedido, automático ou não, e a análise dos dados obtidos pode despoletar uma acção ou um conjunto de acções. As acções são definidas pelos utilizadores e variam do registo de eventos ao envio de notificações para o gestor de rede.

O Sistema Gerido é o dispositivo de rede que implementa um protocolo de gestão, com agente para fornecer dados e notificar o sistema gestor [3]. Este agente é responsável por manter os dados dos objectos geridos, armazenados na base de dados de gestão conceptual denominada *Management Information Base* (MIB). Os dispositivos de rede que não implementam protocolos de gestão são geridos

por agentes *proxy*, que traduzem acções do ambiente nativo para o ambiente externo.

As arquitecturas de gestão descritas ao longo desta secção comportam modelos de informação, modelos organizacionais, modelos de comunicação e modelos funcionais específicos, pelo que o modelo de gestão Gestor-Agente enquadra-se nestes 4 sub-modelos de gestão suplementares como representa a Figura 2.3.

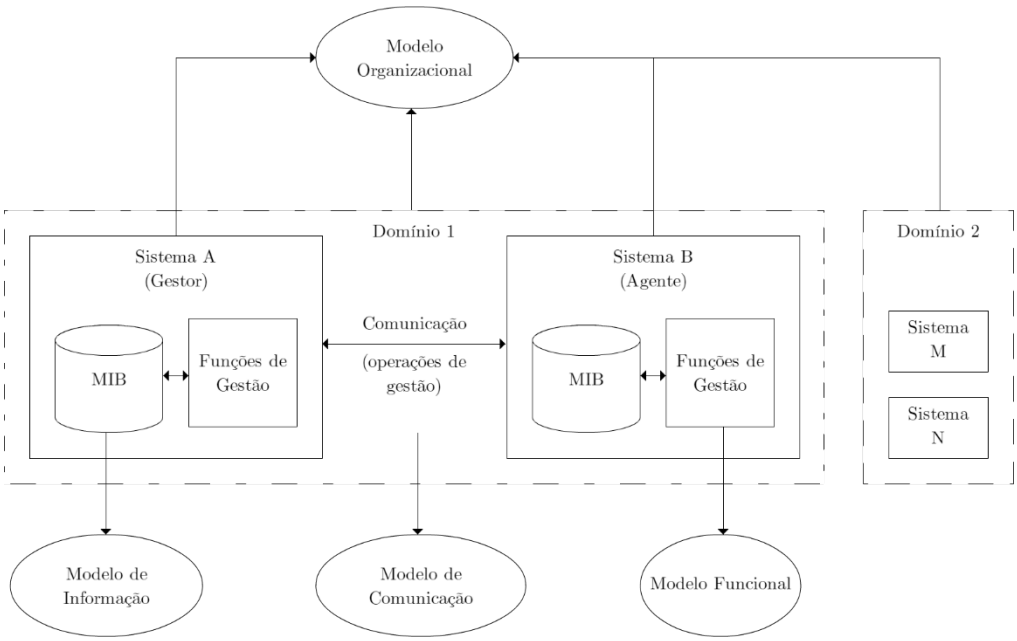


Figura 2.3: Sub-modelos de gestão de redes (baseada em [10])

O Modelo de Informação contém os dados para a descrição e modelação dos objectos geridos. Estes dados incluem sintaxe, semântica, definem propriedades e relações, e fazem o mapeamento entre a informação da MIB e os recursos por ela descritos.

O Modelo Organizacional define os domínios de gestão, e estabelece as responsabilidades intra e inter-domínios [10]. Distribui os objectos geridos e especifica as responsabilidades e os papéis, como agente ou gestor, das entidades de gestão envolvidas. Aqui, são considerados aspectos de contabilização, de segurança, aspectos administrativos e políticos.

O Modelo de Comunicação determina a sintaxe e a semântica da comunicação, e especifica os métodos usados na troca de informação [11]. Aqui, são também

definidos os protocolos e serviços disponíveis para a configuração de objectos geridos, obtenção de estados e envio de notificações.

O Modelo Funcional divide a tarefa de gestão em vários componentes com funcionalidades dedicadas. Descreve os serviços e os objectos geridos relevantes de cada uma das funções de gestão, e identifica a relação estabelecida entre elas para a concretização das funcionalidades esperadas.

2.2.1 Arquitectura de Gestão OSI

A Arquitectura de Gestão OSI, desenvolvida no fim da década de 80, é a primeira a incorporar os 4 sub-modelos de gestão descritos anteriormente [10] [12]. Fornece uma base comum para o desenvolvimento coordenado de normas de gestão, modelo de referência para a definição de outras arquitecturas de gestão como a Arquitectura de Gestão de Redes de Telecomunicações.

2.2.1.1 Modelo de Informação

Aqui, o Modelo de Informação descreve os atributos, as operações e as relações entre objectos do mesmo tipo [5] (à semelhança do que acontece no paradigma orientado a objectos).

São agrupados, em classes, os objectos que partilham os mesmos atributos e as mesmas propriedades. A herança permite que um objecto seja uma instância de uma classe, ou sub-classe de uma ou mais superclasses. As propriedades são herdadas e podem ser refinadas, ou estendidas.

Os dados armazenados seguem a notação *Abstract Syntax Notation One* (ASN.1), e constituem a MIB local do Sistema Gerido.

2.2.1.2 Modelo Organizacional

No Modelo Organizacional, o Sistema Gestor (Gestor) e o Sistema Gerido (Agente) interagem através de protocolos de gestão para execução de operações nos objectos geridos como representa a Figura 2.4.

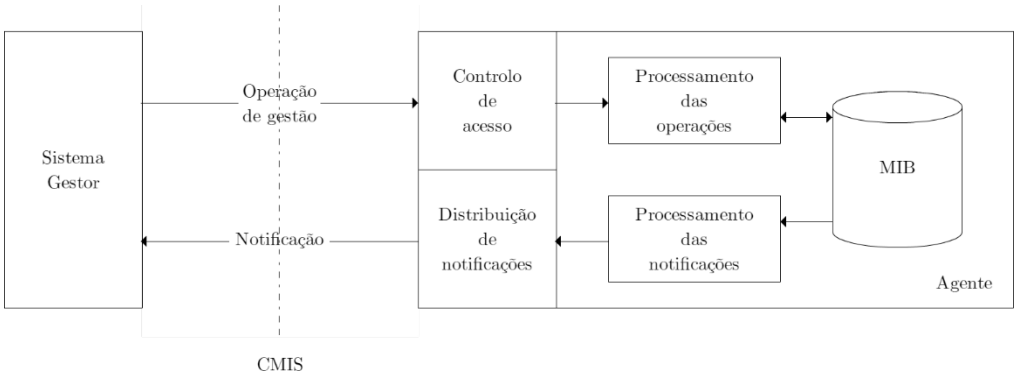


Figura 2.4: Interacção no modelo Gestor-Agente (baseada em [10])

Algumas das operações sobre os objectos geridos implicam a troca dinâmica de papéis entre os sistemas envolvidos, onde o Gestor recebe os resultados, as notificações e as mensagens de erro geradas.

O *Common Management Information Service* (CMIS) representa os serviços comuns de informação de gestão utilizados para sustentar a interacção Gestor-Agente esboçada na Figura 2.4, suportados pelo protocolo comum para informação de gestão *Common Management Information Protocol* (CMIP). Este protocolo especifica os procedimentos de troca de informação de gestão em sistemas abertos, e requer a associação de processos CMIP comunicantes para o envio de mensagens de gestão [10] [13].

2.2.1.3 Modelo de Comunicação

O Modelo de Comunicação é uma representação sistemática, idealizada, do processo de comunicação das arquitecturas de gestão [14]. São três as áreas de gestão distintas do modelo de comunicação da arquitectura de gestão OSI, com diferentes tipos de protocolos e serviços de comunicação: Gestão de Sistemas (*Systems Management*), Gestão de Camada (*Layer Management*) e Operação de Camada (*Layer Operation*) como representado na Figura 2.5.

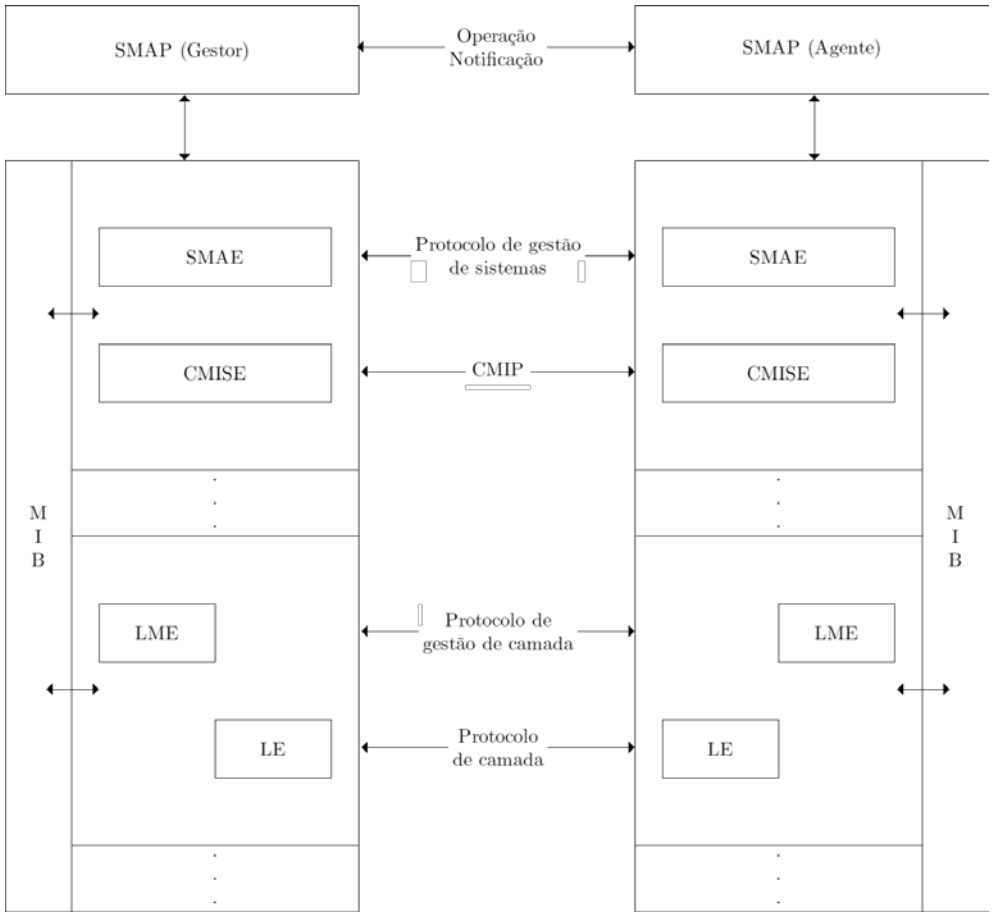


Figura 2.5: Modelo de comunicação OSI (baseada em [10])

A Gestão de Sistema é responsável por acções de gestão conduzidas pelos processos de aplicação de gestão de sistema *System Management Application Processes* (SMAP), não restritas a um sub-sistema ou recurso de rede. Os SMAP comunicam pelas entidades de aplicação de gestão de sistema *System Management Application Entities* (SMAE), que recorrem aos serviços comuns de informação de gestão CMIS.

A Gestão de Camada agrupa as funções, os serviços e os protocolos de uma camada conduzidos pelas entidades de gestão de camada *Layer Management Entity* (LME), e protocolos de gestão de camada.

A Operação de Camada diz respeito às funções de controlo (de sequência, de fluxo e de erros) e operação presentes nos protocolos de camada.

2.2.1.4 Modelo Funcional

O Modelo Funcional comporta as 5 áreas funcionais para atender às necessidades específicas de gestão [15]. Identifica para cada uma das áreas as funcionalidades, o conjunto de funções de gestão de sistema auxiliares predefinidas *System Management Functions* (SMF), e as classes de objectos relevantes correspondentes como representa a Figura 2.6.

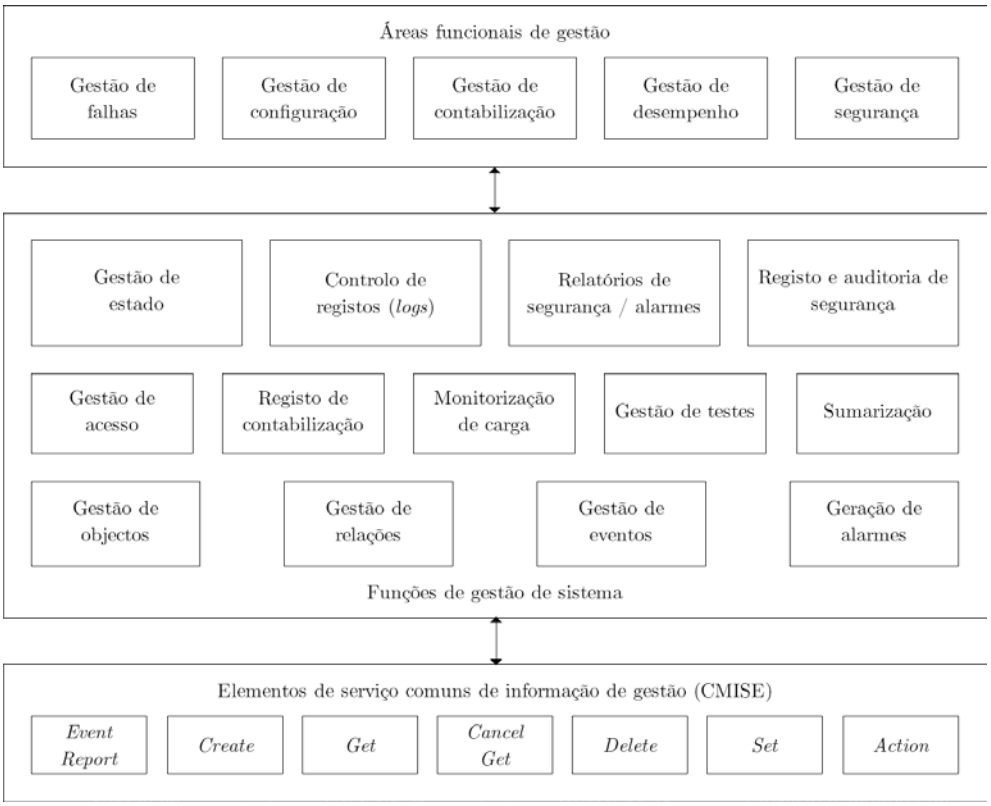


Figura 2.6: Modelo funcional OSI (baseada em [10])

É com base nos elementos de serviço comuns de informação de gestão *CMIS Elements* (CMISE) que as SMF produzem a funcionalidade de gestão esperada.

2.2.2 Arquitectura de Gestão TCP/IP

A Arquitectura de Gestão TCP/IP serve de base à maioria das soluções de gestão para redes de comunicação de dados. Assenta no modelo Gestor-Agente, análogo

à arquitectura cliente-servidor, tipicamente usado nos sistemas de gestão como mostra a Figura 2.7.

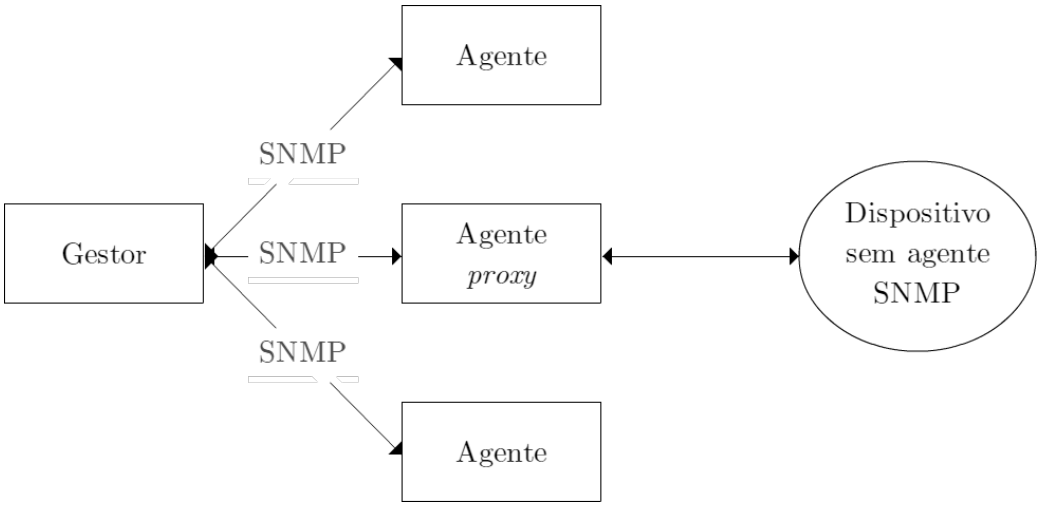


Figura 2.7: Modelo de gestão TCP/IP (baseada em [10])

O Gestor desempenha o papel de cliente, e o Agente desempenha o papel de servidor. A troca de informação Gestor-Agente é feita através do protocolo *Simple Network Management Protocol* (SNMP), disponível para qualquer recurso de rede com suporte TCP/IP [16]. Os agentes *proxy*, omissos na arquitectura de gestão OSI, permitem a gestão de recursos que não tenham um agente próprio ou que não suportem os mesmos protocolos de comunicação e gestão.

O Modelo de Informação não segue o paradigma orientado a objectos presente na arquitectura de gestão OSI, e não se incorporam modelos organizacional e funcional muito distintos [10] [17].

2.2.2.1 Modelo de Informação

O Modelo de Informação, detalhado no *Request For Comments* (RFC) 1155, especifica a estrutura genérica da informação de gestão. Esta informação, organizada em bases de dados de informação de gestão MIB, segue as regras de construção descritas pela *Structure of Management Information* (SMI).

A MIB representa a árvore de registo que contém os dados de gestão do Sistema Gerido. Um nó, também conhecido por *Object IDentifier* (OID), identifica um objecto de gestão como representado na Figura 2.8.

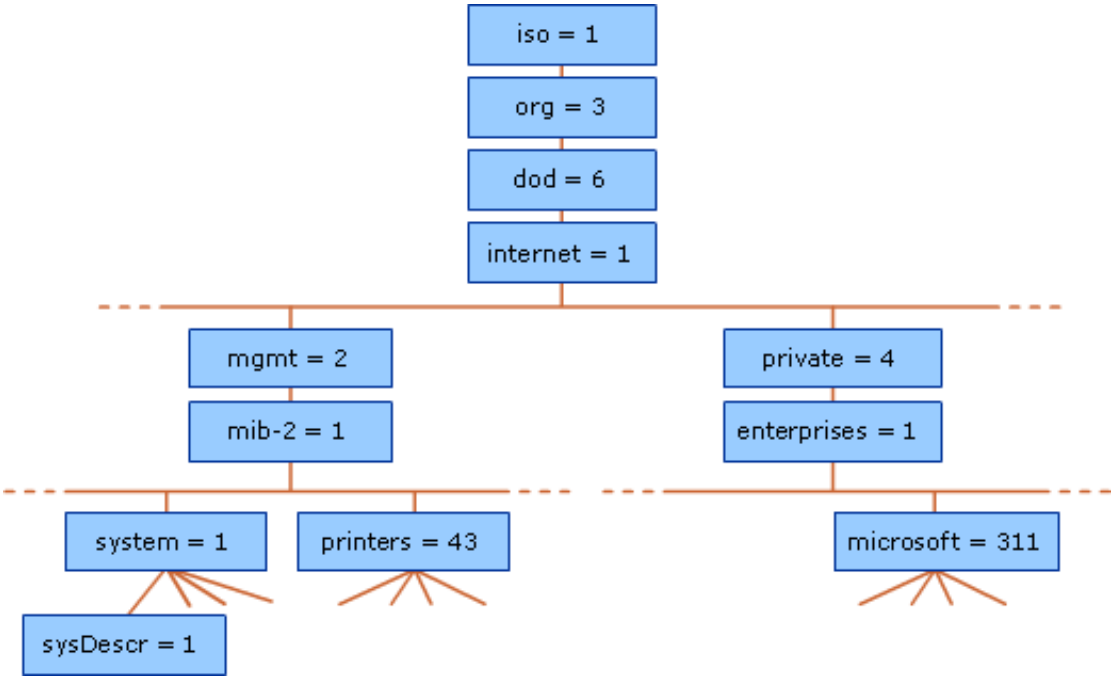


Figura 2.8: Árvore de identificadores de objectos [18]

Um nó sem filhos, folha desta árvore de registo, contém um valor de gestão [19] [20]. Os valores de gestão, codificados em *Basic Encoding Rules* (BER) [21], são números inteiros ou cadeias de caracteres (na maior parte dos casos) com significados específicos.

Na Figura 2.8, iso.org.dod.internet.mgmt.mib-2.system.sysDescr e 1.3.6.1.2.1.1.1 representam o mesmo OID, folha da árvore de registo. Este OID contém uma cadeia de caracteres para descrição do sistema. Como ele, existem mais dos quais podemos e devemos obter informação de gestão sobre registos de interfaces de rede, ligações virtuais, e outros.

A SMI, sub-conjunto da notação formal ASN.1 [22], está divida em três partes: definição de módulos através da macro *MODULE-DEFINITION*, definição de objectos geridos através da macro *OBJECT-TYPE*, e definição de notificações através da macro *NOTIFICATION-TYPE* [23].

ifNumber *OBJECT-TYPE*

SYNTAX *INTEGER*

ACCESS *read-only*

STATUS *mandatory*

DESCRIPTION

"The number of network interfaces (regardless of their current state) present on this system."

::= *{ interfaces 1 }*

Aqui, *interfaces* e 1.3.6.1.2.1.2 representam o mesmo OID. *ifNumber* é o nome atribuído ao OID 1.3.6.1.2.1.2.1 (identificador de objecto). *SYNTAX*, ou sintaxe, define a estrutura de dados abstracta que corresponde ao tipo do objecto. *ACCESS*, ou acesso, define quando é que o valor do objecto pode ser lido, ou lido e modificado. *DESCRIPTION*, ou descrição, contém uma definição textual do objecto.

2.2.2.2 Modelo de Comunicação

O Modelo de Comunicação assenta no protocolo de gestão SNMP, padrão da Internet, com três tipos de operações de comunicação: Gestor-Agente, Agente-Gestor e Gestor-Gestor.

Na comunicação Gestor-Agente, o Gestor acede à MIB do Agente para consultar ou modificar objectos geridos. A modificação de objectos geridos é levada a cabo pelo Agente, que indica o resultado das operações pedidas.

Na comunicação Agente-Gestor, o Agente notifica ao Gestor a ocorrência de situações particularmente relevantes com *traps* sem solicitação prévia.

Na comunicação Gestor-Gestor, a troca de informação possibilita a descrição de uma MIB (ou várias, conforme necessário).

Foi no fim da década de 80 que o SNMP substituiu o protocolo *Simple Gateway Monitoring Protocol* (SGMP), como mostra a Figura 2.9.

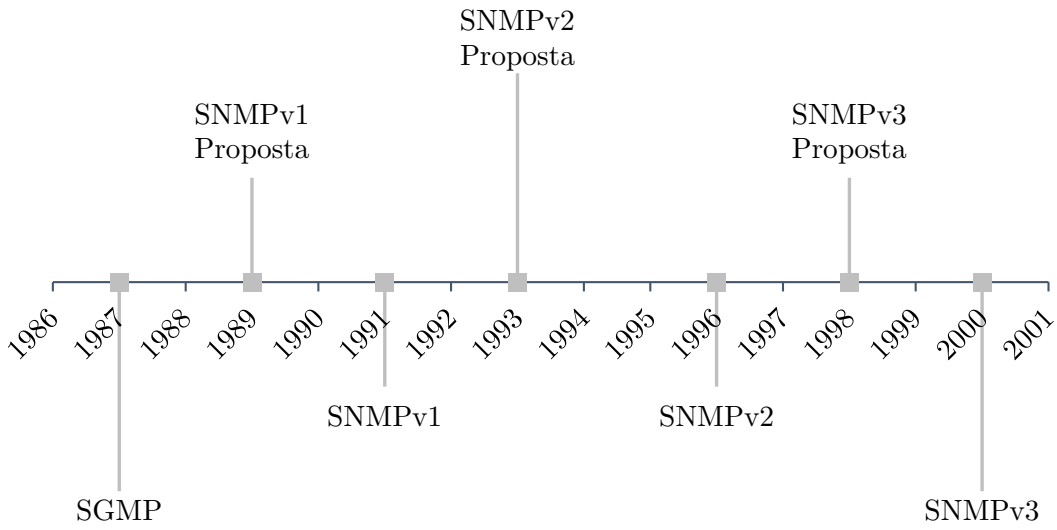


Figura 2.9: Evolução do protocolo SNMP

O protocolo SNMP recorre ao protocolo *User Datagram Protocol* (UDP) para transmissão de dados com maior eficiência, e usa a SMI (SNMP SMI [24]) para lidar com objectos usados nas mensagens SNMP como representa a Figura 2.10.

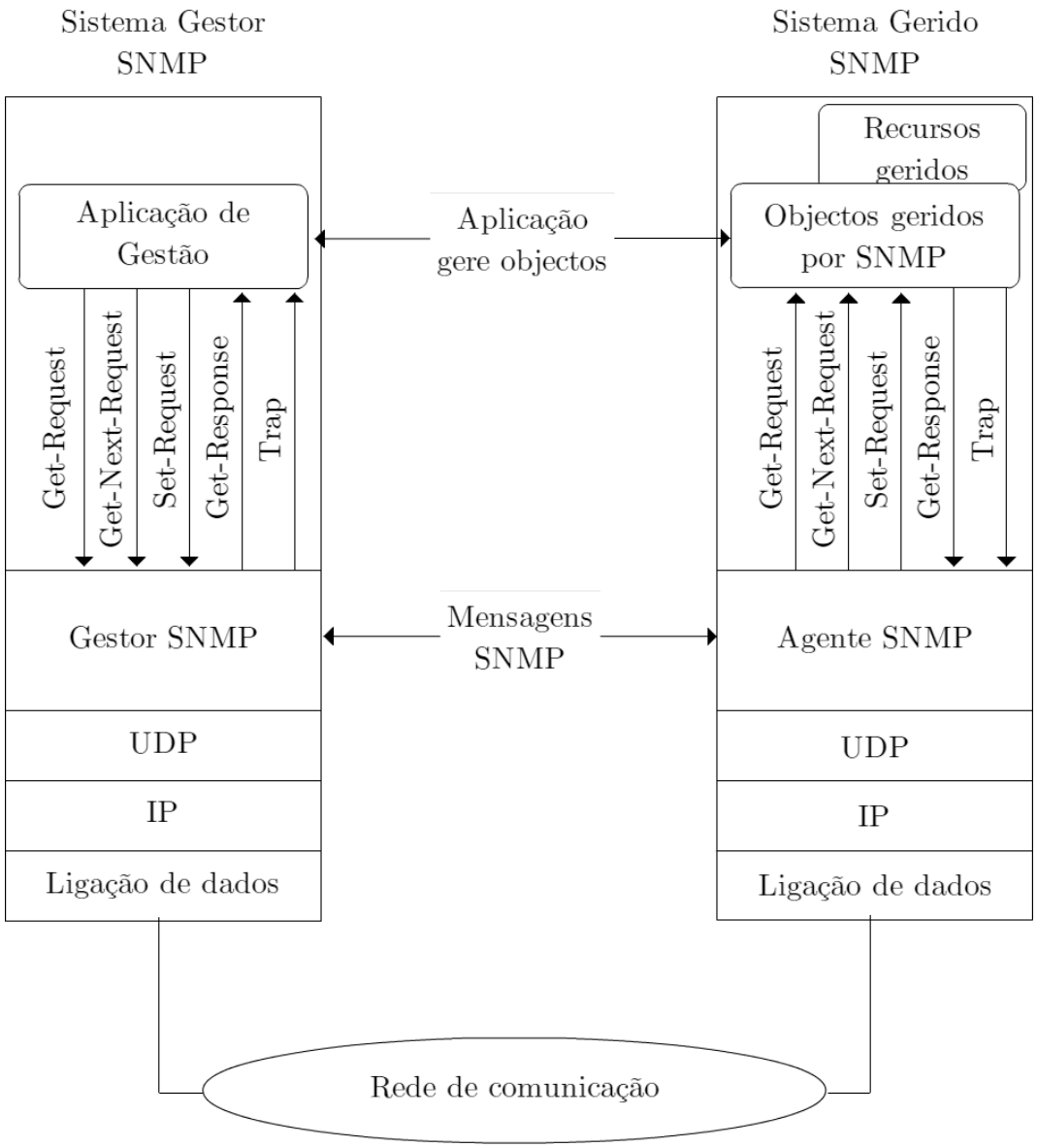


Figura 2.10: Arquitectura do protocolo SNMP (baseada em [25])

As mensagens SNMP contêm um cabeçalho e uma *Protocol Data Unit* (PDU). O cabeçalho especifica a versão do protocolo e o nome da comunidade SNMP para autenticação. A PDU [19] [26] está associada à comunicação, com formato e conteúdo diferente para cada operação.

O SNMP foi estendido para incluir novas funcionalidades e melhorar aspectos de segurança, dando origem às versões SNMPv1, SNMPv2 e SNMPv3 [10]. A estrutura da informação de gestão destas versões, SNMPv1 SMI (SMIv1),

SNMPv2 SMI (SMIv2) e SNMPv3 SMI (SMIv2) [23] é definida nos RFC 1155 e RFC 1902 [3].

A primeira versão, SNMPv1, dispõe das 4 operações Get-Request, Get-Next-Request, Set-Request e Trap, onde:

- Get-Request permite, ao Sistema Gestor, ler informação de gestão da MIB do Sistema Gerido através do agente respectivo. Este tipo de operação implica uma resposta Get-Response com o mesmo identificador de pedido, e com os valores solicitados. A resposta contém, em situações de falha, o código de erro para identificação do problema.
- Get-Next-Request permite a leitura sequencial de objectos, útil quando não se conhece a estrutura da MIB em questão. Este tipo de operação resulta, também, numa resposta do tipo Get-Response.
- Set-Request permite, ao Sistema Gestor, modificar um ou mais objectos da MIB em questão com os valores indicados. A resposta, do tipo Get-Response, indica o sucesso ou a falha da operação mediante a existência do(s) objecto(s) especificado(s), as permissões de acesso ao(s) mesmo(s), e a(s) gama(s) de valores para ele(s) estipulada(s).
- Trap permite, ao Sistema Gerido, notificar ao Sistema Gestor a ocorrência de situações particularmente relevantes sempre que necessário de forma assíncrona e sem solicitação prévia.

A segunda versão, SNMPv2, permite ao Gestor desempenhar o papel de Agente e ao Agente desempenhar o papel de Gestor, em interacções devidas. É mais seguro, oferece um modelo de informação com suporte para novos tipos de dados, e um modelo de comunicação com suporte para as duas novas operações Get-Bulk-Request e Inform-Request, onde:

- Get-Bulk-Request permite ler uma tabela de dados inteira, e optimiza as transferências de grandes volumes de dados. Devolve os valores de todos os objectos pedidos, não sujeitos a qualquer tipo de erro.
- Inform-Request permite a comunicação Gestor-Gestor para troca de informação de uma MIB (ou várias) entre sistemas gestores.

A terceira versão, SNMPv3, resolve as questões de seguranças adiadas pelas versões precedentes dado que as tecnologias propostas para o SNMPv2 não se revelaram totalmente seguras.

O modelo de gestão *Remote Network MONitoring* (RMON), extensão do protocolo SNMPv2, é amplamente usado na monitorização de redes [27] [28].

Requer dispositivos de gestão compatíveis que, sem interferir nas operações de gestão tradicionais, obtenham informação de gestão específica. Os dispositivos, para além da comunicação com o Gestor, têm funcionalidade de gestão própria para processar e armazenar a informação obtida. A informação é descrita pela RMON MIB, presente nos dispositivos de gestão respectivos como mostra a Figura 2.11.

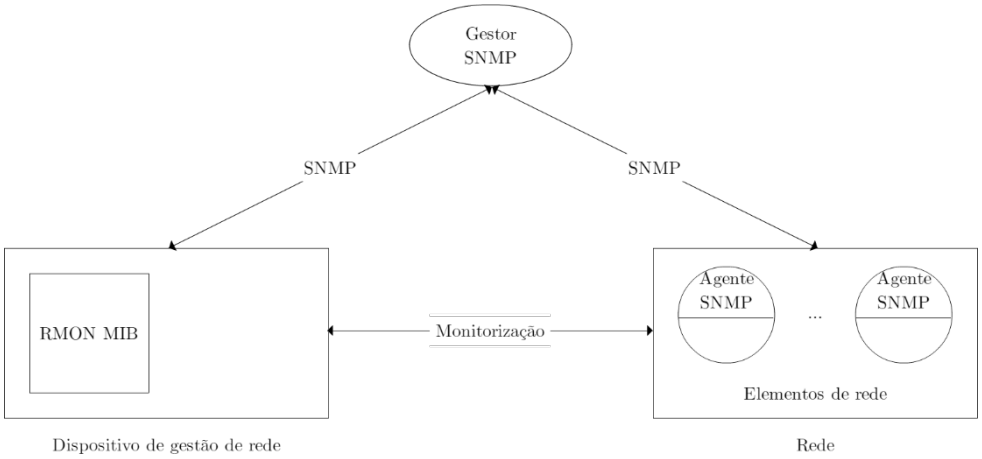


Figura 2.11: Modelo de gestão RMON (baseada em [10])

A RMON MIB, definida no RFC 2819, especifica 9 grupos de objectos que permitem armazenar estatísticas. A análise da informação armazenada possibilita definir *thresholds*, configurar *triggers*, controlar a geração de eventos e filtrar dados de tráfego da rede local monitorizada.

A RMON2 MIB, definida no RFC 4502, permite a monitorização fora do âmbito das redes locais através da adição de objectos de níveis protocolares superiores ao nível de ligação de dados do modelo OSI.

2.2.3 Arquitectura de Gestão TMN

A Arquitectura de Gestão TMN foi desenvolvida pela ITU-T nas décadas de 80 e 90, mas acompanha a evolução das redes de telecomunicações actuais [10]. É fortemente baseada na arquitectura de gestão OSI, e tem como principal objectivo a gestão homogénea de redes heterogéneas comuns nas operadoras de telecomunicações.

Assenta no conceito de rede sobreposta, o que implica uma gestão feita através de uma rede de gestão distinta. A rede de gestão, que interage com pontos de comutação e transmissão da rede de telecomunicações gerida, permite aceder às estações de trabalho de gestão, e a comunicação com os sistemas de operações como representado na Figura 2.12.

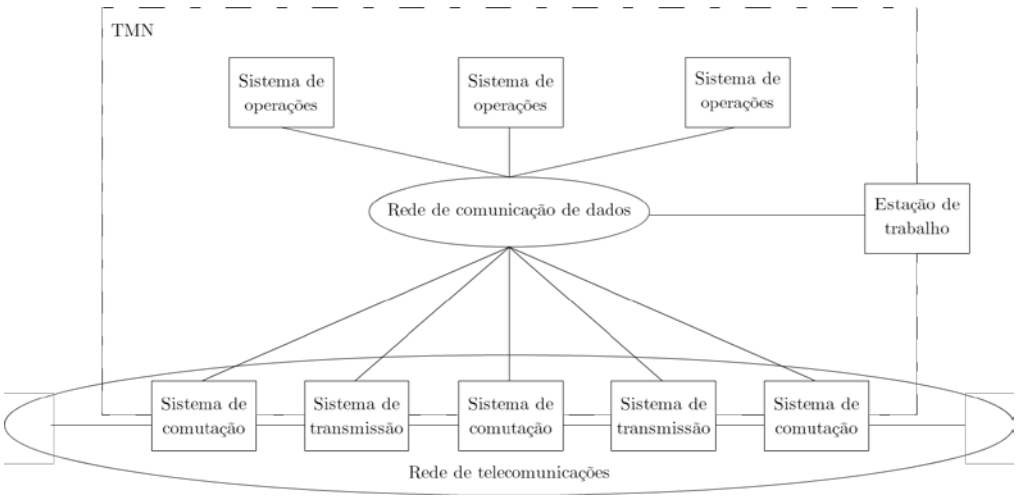


Figura 2.12: Arquitectura de gestão TMN (baseada em [10])

De acordo com [29], são 4 as arquitecturas de gestão definidas conforme níveis de abstracção diferentes:

- Arquitectura Funcional
- Arquitectura Física
- Arquitectura de Informação
- Arquitectura Lógica

A Arquitectura Funcional descreve um conjunto de funções de gestão. Identifica os tipos de blocos funcionais *Operations System Functions* (OSF), *Work Station Functions* (WSF), *Q Adaptor Functions* (QAF), *Network Element Functions* (NEF) e *Mediation Functions* (MF) como representado na Figura 2.13.

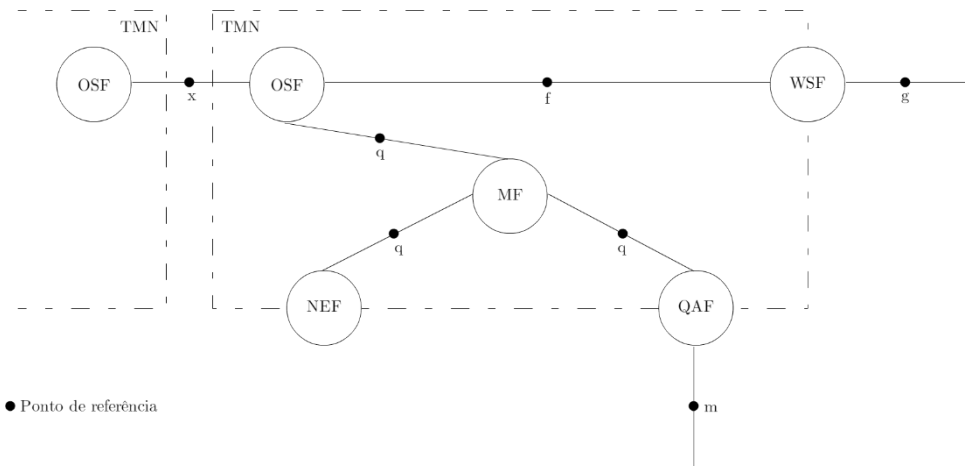


Figura 2.13: Pontos de referência entre blocos funcionais (baseada em [10])

A interacção entre os tipos de blocos funcionais é feita através de 5 pontos de referência, onde:

- OSF: Desempenha as funções de gestor
  Isto é, inicia operações de gestão e recebe notificações.
- WSF: Permite ao utilizador humano aceder à informação de gestão
- QAF: Estabelece ligação entre a rede de gestão TMN e as entidades ou sistemas que não suportam a gestão TMN
- NEF: É o bloco associado aos Elementos de Rede (NE[3])
  Desempenha as funções de agente, e gera notificações.
- MF: É por onde passa a informação entre os blocos NEF, ou QAF, e OSF
  Pode filtrar, transformar ou armazenar informação de gestão.

A Arquitectura Física define como implementar as funções de gestão em equipamento físico.

A Arquitectura de Informação descreve os conceitos que foram e são adoptados da gestão OSI.

A Arquitectura Lógica inclui um modelo que indica como estruturar a gestão com responsabilidades distintas. Expõe 5 perspectivas de gestão, em pirâmide, dos Elementos de Rede à Gestão de Negócio. As fatias inferiores da pirâmide concentram funções de gestão mais granulares, e as fatias superiores comportam funções de gestão tangentes às grandes linhas de acção do operador de telecomunicações como representa a Figura 2.14.

[3]Network Elements

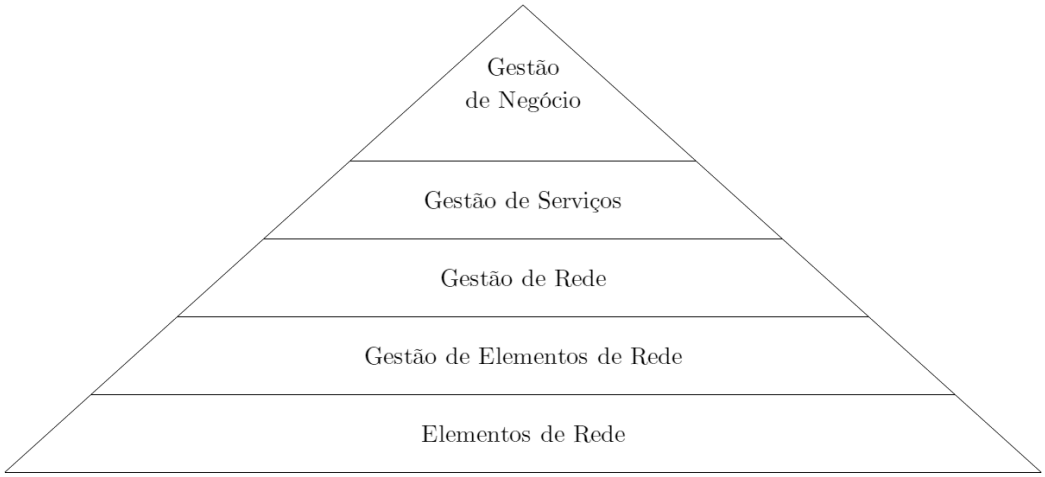


Figura 2.14: Arquitectura lógica de gestão TMN (baseada em [10])

A Gestão de Elementos de Rede monitoriza o nível de desempenho dos Elementos de Rede (NE). A informação recolhida é armazenada numa base de dados, e analisada por funções de gestão de nível superior.

A Gestão de Rede comporta aspectos de conectividade, desempenho, encaminhamento de informação, congestão e falhas de rede.

A Gestão de Serviços é responsável por construir, monitorizar e manter serviços procurados pelos utilizadores. A gestão de utilizadores, a contabilização de recursos e a qualidade de serviço contratada são aspectos de interesse neste nível de gestão.

A Gestão de Negócio envolve decisões que afectam o desempenho do negócio. Aqui temos a análise de custos e lucros, análise de adesão e análise de serviços bem conseguidos.

2.2.4 Arquitectura de Gestão baseada na Web

A aceitação e a divulgação na Web de hoje, juntamente com a disponibilidade das tecnologias inerentes, a independência face aos sistemas operativos (e fabricantes, consequentemente), a representação da informação, a facilidade de comunicação e outras vantagens oferecidas suportam a integração de soluções de gestão de redes e sistemas modernos [10].

A integração de gestão na plataforma Web comporta uma de duas abordagens básicas, ou a combinação delas: Abordagem integrada para gestão baseada na Web e Abordagem com *proxy* para gestão baseada na Web.

A Abordagem integrada para gestão baseada na Web representada na Figura 2.15 é útil para redes pequenas, pois evita a utilização de uma plataforma de gestão com o propósito de auxiliar nas actividades de gestão.

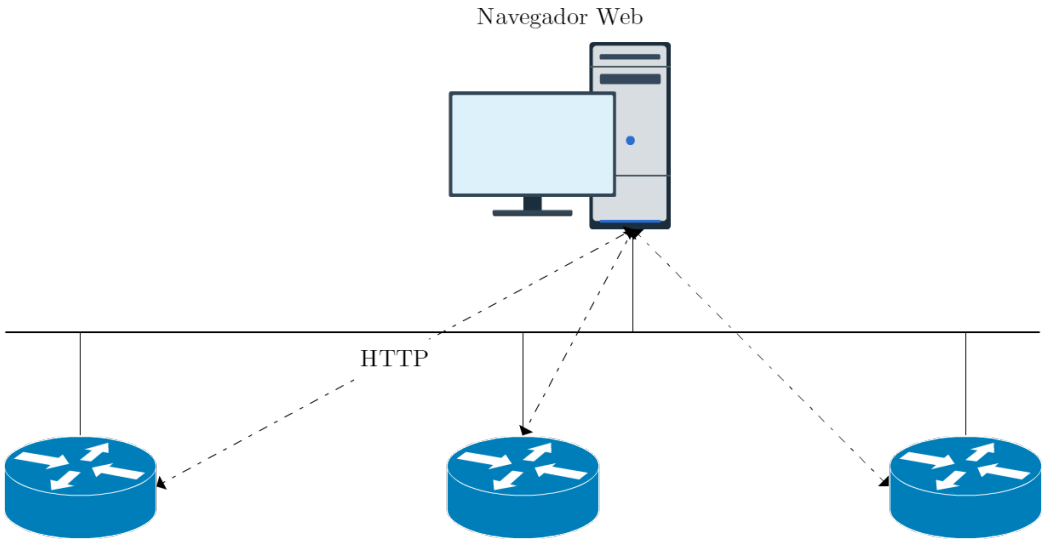


Figura 2.15: Abordagem integrada para gestão baseada na Web

Evitar a utilização de uma plataforma de gestão com o propósito de auxiliar nas actividades de gestão reduz custos, mesmo quando se substitui ferramentas de gestão tradicionais como a linha de comandos.

A Abordagem com *proxy* para gestão baseada na Web representada na Figura 2.16 é útil para redes de maior dimensão, com dispositivos centralizados responsáveis por recolher, correlacionar e processar grande volume de dados.

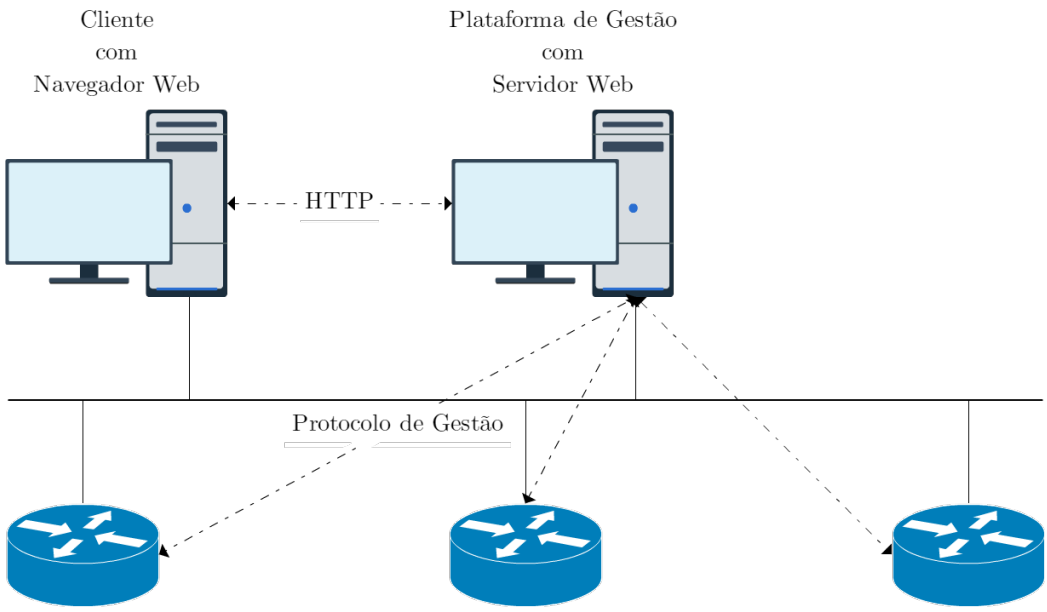


Figura 2.16: Abordagem com *proxy* para gestão baseada na Web

Os dispositivos centralizados responsáveis por recolher, correlacionar e processar grande volume de dados comportam custos mas substituem ferramentas de gestão tradicionais com uma plataforma de gestão complexa, robusta e completa.

2.2.4.1 WBEM

A iniciativa WBEM esteve, em 1996, a cargo das empresas Microsoft, Intel, BMC Software, Compaq e Cisco Systems [30]. A tecnologia, adoptada pela *Distributed Management Task Force* (DMTF), é uma norma de facto amplamente aceite pelos principais fabricantes de equipamento e *software*. Permite o acesso transparente aos objectos geridos com uma arquitectura simples do lado do cliente, como mostra a Figura 2.17.

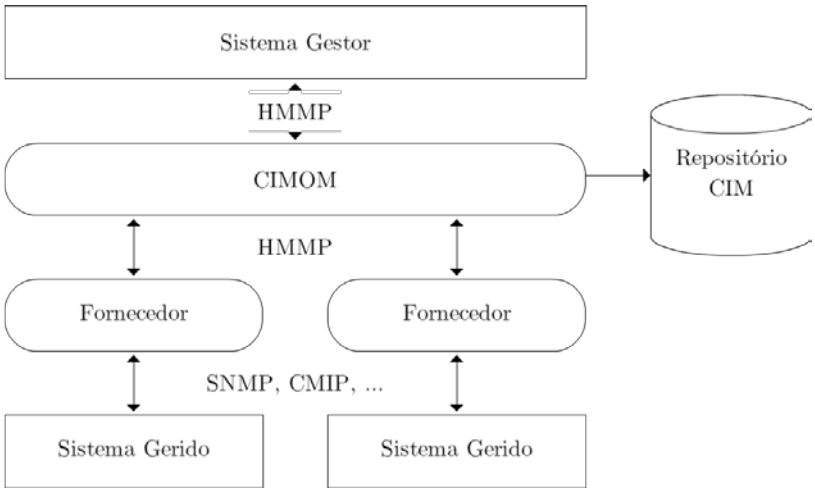


Figura 2.17: Arquitectura WBEM (baseada em [30])

O modelo de informação assenta no *Common Information Model* (CIM), que contém uma definição normalizada de informação de gestão de redes, sistemas, aplicações e serviços, e que possibilita a criação de extensões específicas de fabricantes.

A codificação dos objectos, feita em *eXtensible Markup Language* (XML), torna a informação independente da plataforma de suporte. O transporte desta informação, por *HyperText Transfer Protocol* (HTTP), facilita a comunicação e o desenvolvimento de aplicações de gestão.

Outra iniciativa da DMTF é a gestão baseada em serviços Web, que representa um protocolo de gestão de propósito geral para sistemas baseados em *Simple Object Access Protocol* (SOAP). Oferece, segundo [31], um conjunto de operações de gestão típicas para:

- Procurar e descobrir recursos de gestão
- Criar, obter, actualizar e apagar recursos de gestão
- Subscrever a notificações geradas pelos recursos geridos
- Executar métodos de gestão específicos

Este conjunto de operações deve ser estendido com a implementação de funcionalidades pretendidas, e reduzido com a remoção de operações inapropriadas para o dispositivo ou sistema em questão.

2.3 Conclusão

Neste capítulo, foram apresentados e descritos os aspectos fundamentais da gestão de redes. Foram expostas as áreas funcionais FCAPS, o modelo de gestão Gestor-Agente e as arquitecturas de gestão OSI, TCP/IP, TMN e arquitectura de gestão na baseada na Web.

As arquitecturas de gestão consolidam o modelo Gestor-Agente, análogo à arquitectura cliente-servidor. Este modelo enquadra-se, normalmente, em 4 sub-modelos de gestão complementares: Modelo de Informação, Modelo Organizacional, Modelo de Comunicação e Modelo Funcional.

O Modelo de Informação contém os dados para a descrição e modelação dos objectos geridos. O Modelo Organizacional define os domínios de gestão, e estabelece as responsabilidades intra e inter-domínios. O Modelo de Comunicação especifica os métodos utilizados na troca de informação. O Modelo Funcional divide a tarefa de gestão em vários componentes com funcionalidades dedicadas.

A Arquitectura de Gestão OSI foi a primeira a incorporar estes sub-modelos de gestão, e serviu de base para a definição de outras arquitecturas de gestão. A Arquitectura de Gestão TCP/IP, referência nas soluções de gestão para redes de comunicação de dados, comporta agentes *proxy* para gerir recursos sem agente próprio ou que não suportem os mesmos protocolos de comunicação e gestão. A Arquitectura de Gestão TMN tem como objectivo a gestão homogénea de redes heterogéneas comuns na área das telecomunicações.

A criação e o planeamento estratégico de redes heterogéneas a ritmo acelerado acabou, contudo, por complicar as operações de gestão de redes do quotidiano, com emergência para um novo conceito: A Gestão de Redes Integrada e Automatizada.

A Gestão de Redes Integrada e Automatizada surge para dar resposta aos problemas dos gestores de redes modernas, com plataformas de gestão capazes de substituir um conjunto de ferramentas tradicionais. Algumas destas plataformas são apresentadas e descritas no capítulo seguinte.

Capítulo III

Plataformas de Gestão de Redes

As Plataformas de Gestão de Redes substituem um conjunto de ferramentas tradicionais demasiado específicas para a realidade das infra-estruturas de telecomunicações modernas. Comportam uma solução integrada para auxiliar na tarefa de gestão através de *Graphical UI* (GUI) para interagir com o gestor, aplicações de gestão para executar funções de gestão essenciais, núcleo para coordenar a interacção entre componentes, base de dados para armazenar dados, mecanismos de tratamento de informação e protocolos de gestão para a comunicação.

São dotadas de mecanismos de autenticação e autorização que consentem ao gestor de redes o acesso devido. Permitem definir utilizadores e grupos de utilizadores, e atribuir as permissões de utilizador que melhor se adequam em cada caso. Possibilitam a definição e o agrupamento lógico de dispositivos, serviços e grupos de ambos, gerir e monitorizá-los contínua e automaticamente [32] [33]. Permitem definir *triggers* específicos para *thresholds* igualmente específicos sobre dados recolhidos, e associar notificações para alertar ao gestor a ocorrência de problemas e situações particularmente relevantes através de *Short Message Service* (SMS) ou correio electrónico. Possibilitam a geração de gráficos e relatórios personalizados com dados para tratamento estatístico, e a criação de mapas de rede (automáticos ou não).

SolarWinds, Zenoss (Service Dynamics) e *Paessler Router Traffic Grapher* (PRTG) Network Monitor são exemplos de plataformas comerciais disponíveis no mercado, referidas ao longo das reuniões com os gestores de rede da infra-estrutura de telecomunicações da NOS Madeira. Icinga, Nagios Core, Zabbix e o próprio Zenoss (Core) são as alternativas *open-source* igualmente completas propostas internamente e, por isso, abordadas neste capítulo.

Neste capítulo é também apresentada e descrita a arquitectura genérica das plataformas de gestão. São evidenciados critérios de selecção, realçadas funcionalidades particulares e determinadas vantagens/desvantagens quando se opta por uma solução no lugar de outra.

3.1 Arquitectura

A arquitectura genérica das plataformas de gestão comporta 5 módulos funcionais com responsabilidades distintas: Módulo de Controlo, Módulo de Comunicação, Módulo de Informação, Módulo de Aplicações e Módulo de Interface como representa a Figura 3.1.

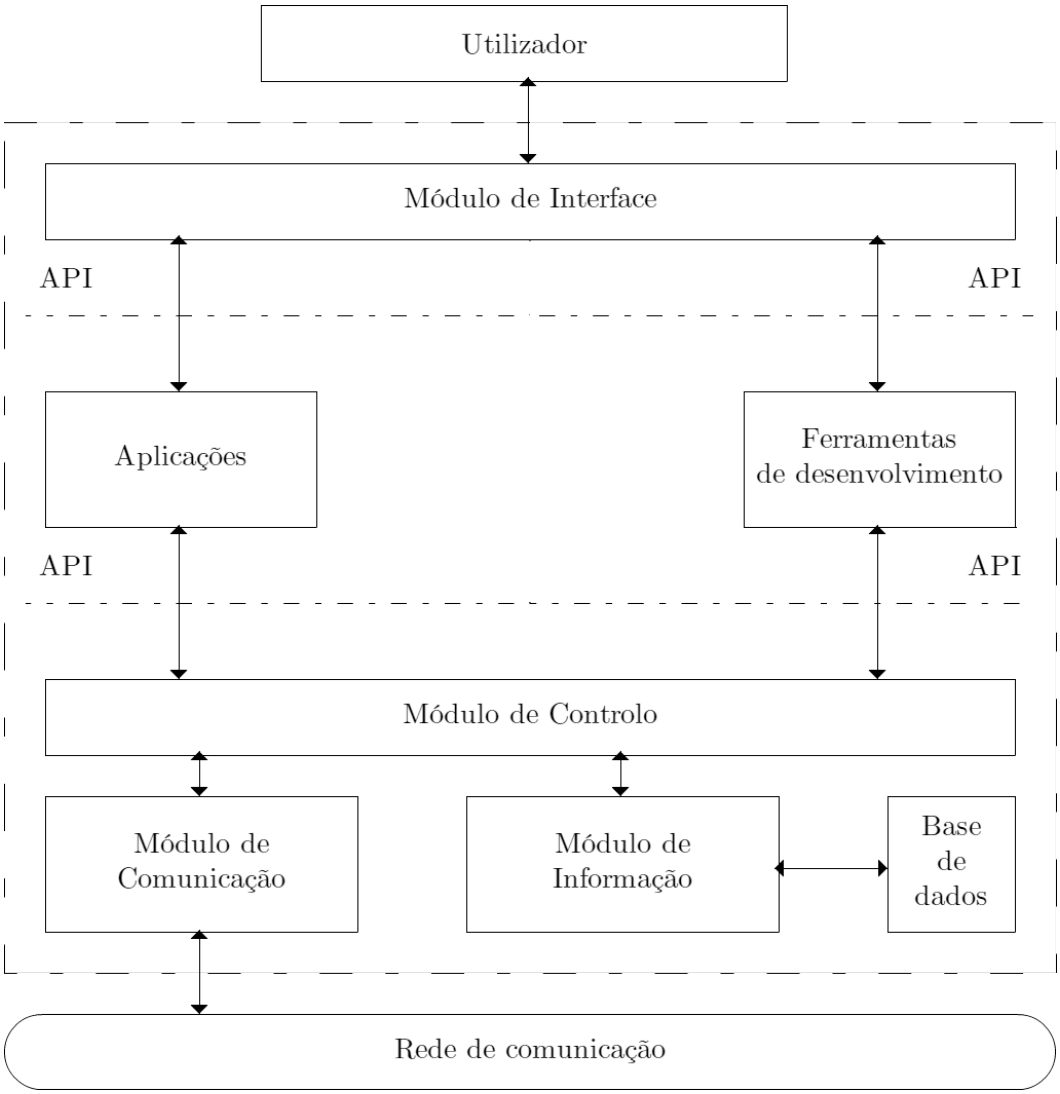


Figura 3.1: Arquitectura genérica das plataformas de gestão (baseada em [10])

O Módulo de Controlo é o ponto central de cada plataforma. Coordena (escalona, sincroniza) as acções dos outros módulos, e comuta a informação entre eles.

O Módulo de Comunicação sustenta a interacção com outros sistemas, através de uma interface de comunicação independente dos vários protocolos de comunicação utilizados.

O Módulo de Informação comporta o modelo de informação da plataforma de gestão armazenado na Base de Dados, comummente orientado a objectos.

O Módulo de Aplicações de gestão congrega:

- Aplicações Básicas: Executam as funções de gestão essenciais com vista à Gestão de Configuração, Gestão de Topologia (descoberta automática da topologia de rede com periodicidade), Gestão de Eventos (recepção, filtragem, processamento, armazenamento, correlação e diagnóstico de eventos internos e eventos externos), Monitorização de Estado (ICMP, SNMP e CMIP) e Monitorização de Desempenho (*thresholds*)
- Aplicações de Desenvolvimento (Ferramentas de Desenvolvimento): Possibilitam o desenvolvimento de aplicações e a modificação de aplicações presentes na plataforma original
- Aplicações de Gestão: Permitem, em conjunto com as Aplicações Básicas, a gestão de sistemas complexos e de grandes dimensões

O Módulo de Interface permite ao utilizador manipular e controlar facilmente a plataforma através de interfaces gráficas. Estas interfaces tornam possível a representação física e lógica da rede gerida, normalmente, e permitem visualizar os recursos monitorizados com diferentes níveis de detalhe.

3.2 Critérios de Selecção

Uma plataforma de gestão deve possibilitar aos gestores de rede controlar a generalidade dos recursos necessários, como pretendem. Só assim, e mediante uma análise sobre aspectos de funcionalidade, extensibilidade, interoperabilidade, segurança, tecnologia e aplicações é que o gestor de redes adequa a oferta à procura, e estima o custo do investimento.

3.2.1 Funcionalidades

É com a funcionalidade de gestão da plataforma que o gestor de redes concretiza a tarefa que lhe foi atribuída. A plataforma deve, com isto, excluir a necessidade de mecanismos de gestão terceiros e suprimir a intervenção manual do gestor tanto quanto possível.

Deve ser configurável, permitir cópias de segurança, estar preparada para ser actualizada sempre que necessário e registar a actividade dos utilizadores respectivos. Deve permitir a gestão de redes e sistemas heterogéneos, processar eventos internos/externos e ajudar na detecção, prevenção e correcção de erros recorrentes.

Deve gerar relatórios sobre processos, largura de banda utilizada e outros aspectos relevantes no âmbito das infra-estruturas de telecomunicações. Deve idealmente oferecer opção para gerar mapas de rede automáticos ou permitir defini-los manualmente, de acordo com as preferências de quem os cria.

3.2.2 Extensibilidade

É com base na extensibilidade que o gestor de redes moderno desperta interesse numa solução em prol de outra. Uma plataforma de gestão de redes extensível oferece portabilidade para funcionar noutro sistema de rede, e adaptabilidade para desenvolvimento futuro [34].

A utilização de uma arquitectura modular e escalável permite o desenvolvimento e a integração de módulos funcionais que interagem com módulos de sistemas diferentes. Este tipo de arquitectura permite delegar actividades de gestão em sub-sistemas remotos que constituem sub-domínios.

3.2.3 Interoperabilidade

É cada vez mais comum o gestor de redes exigir uma plataforma de gestão de redes aberta. Este tipo de solução permite a interoperabilidade [35] entre dispositivos de natureza e fabricantes distintos, mediante interfaces e tecnologias normalizadas.

São poucos os casos em que se utiliza um dispositivo ou uma plataforma de gestão de redes isolados, e ainda menos os que se consegue qualquer tipo de integração adicional. A integração com plataformas e ferramentas de gestão terceiras, e a utilização de protocolos abertos conferem graus de interoperabilidade desejados.

3.2.4 Segurança

Uma plataforma de gestão de redes deve fornecer um sistema de autenticação e autorização de utilizadores, e registar as operações de gestão por eles efectuadas. Só assim para prevenir acções críticas indevidas de utilizadores igualmente indevidos, ou não autorizados [33].

A configuração indevida pode, em certos casos, tornar inoperacional toda uma infra-estrutura de telecomunicações. O envio de notificações para o gestor responsável é uma mais-valia procurada neste tipo de situações.

3.2.5 Tecnologia

Só uma plataforma de gestão de redes moderna pode ser preferida pelo gestor de redes moderno. A utilização de tecnologias e *software* aplicacional vigente, conjuntamente com *hardware* e sistemas operativos actuais, dá resposta às necessidades do administrador de sistemas.

A adopção da plataforma Web [36] para cumprir a meta tem sido verificada na maioria das soluções de gestão integradas ou não integradas, comerciais ou não comerciais.

3.2.6 Aplicações

A possibilidade de incluir aplicações para automatização da tarefa (ou parte da tarefa) de gestão é outra mais-valia desejada pelo gestor de redes moderno.

Aqui, é expectável a presença ou a liberdade para adicionar aplicações que permitam identificar problemas, propor ou iniciar acções correctivas com impacto na redução da intervenção manual do responsável.

3.2.7 Custo

O custo é, cada vez mais, um factor determinante na selecção de uma plataforma de gestão de redes.

As plataformas *open-source* comportam custos na adaptação e no desenvolvimento de aplicações. As plataformas comerciais comportam custos de investimento, manutenção e actualização [34].

Os recursos humanos que as controlam são um custo comum, pelo que a adopção de qualquer uma das soluções é adequada somente nas redes de média-grande dimensão (e onde a gestão assume um papel fundamental).

3.3 Plataformas Open-Source

As Plataformas *Open-Source* presentes no mercado são cada vez mais opção para os gestores das infra-estruturas de telecomunicações actuais. Isto porque, contrariamente às plataformas comerciais, podem ser descarregadas e testadas sem qualquer tipo de custo por isso.

Este tipo de plataformas dá preferência aos padrões abertos, o que melhora a interoperabilidade entre aplicações *open-source*. O código-fonte está exposto a quem quer que as utilize pelo que a correcção de *bugs* e vulnerabilidades de segurança é, naturalmente, mais rápida e pertinente.

3.3.1 Icinga

O Icinga é uma plataforma de gestão *open-source* escalável, utilizada para monitorizar infra-estruturas de telecomunicações. Permite notificar, auxiliar na resolução de problemas, e gerar relatórios que reflectem a disponibilidade dos dispositivos monitorizados.

É uma bifurcação do Nagios, pelo que são vários os extras disponíveis para estender a funcionalidade base: o NagVis para visualizar o estado da monitorização, o PNP4Nagios para tratar dos gráficos, o NConf e o NagiosQL para gerar ficheiros de configuração através da Web [37].

Consiste em vários componentes e interfaces Web como Icinga Classic e Thruk que permitem agendar interrupções de serviço, adicionar comentários, desactivar verificações de serviços, e outros como representa a Figura 3.2.

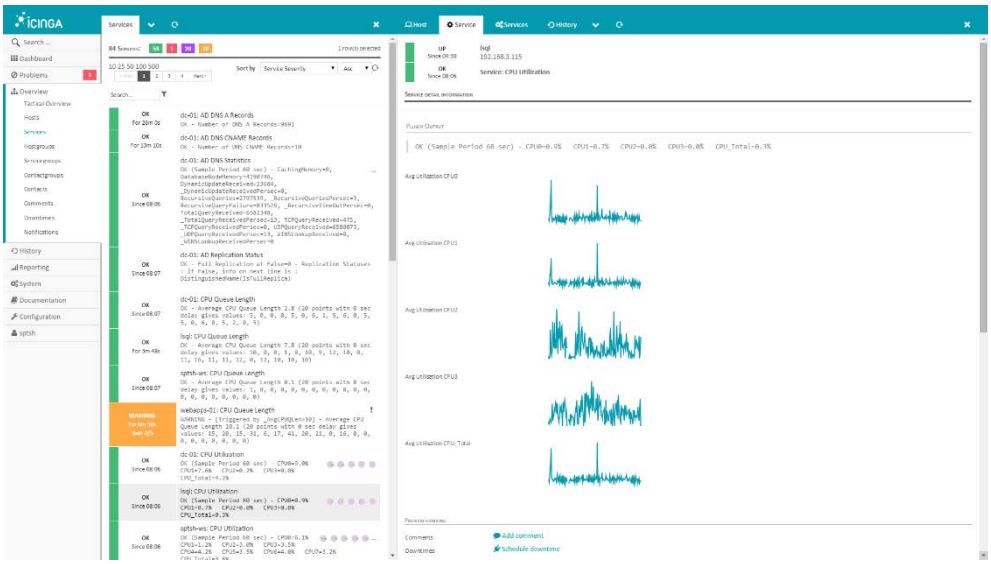


Figura 3.2: Interface Web do Icinga

Está em desenvolvimento activo, e conta com uma versão nova por ano desde 2009. A referência temporal ilustrada na Figura 3.3 data o lançamento das versões mais importantes.

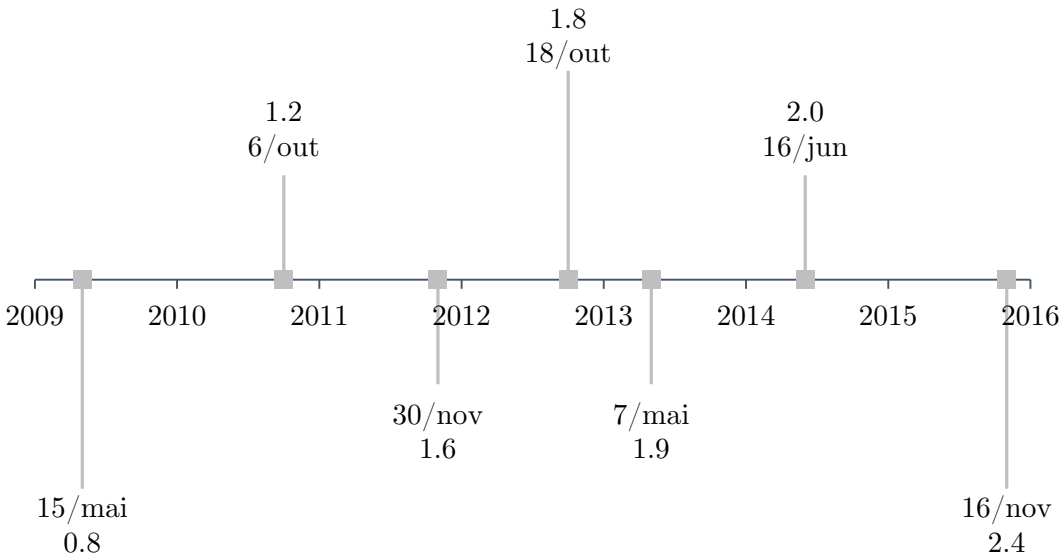


Figura 3.3: Evolução do Icinga

As características da plataforma Icinga são:

- Arquitectura modular e flexível
- Interface Web dinâmica
- Extensibilidade para empresas
- Configuração baseada em objectos
- Atribuição e aplicação de atributos
- Comandos e macros inteligentes
- Dependências lógicas
- Notificações dinâmicas

Os componentes arquitecturais que sustentam as características descritas são o Icinga Core, o *Icinga Data Out DataBase* (IDODB) e o Icinga Web, interligados como mostra a Figura 3.4.

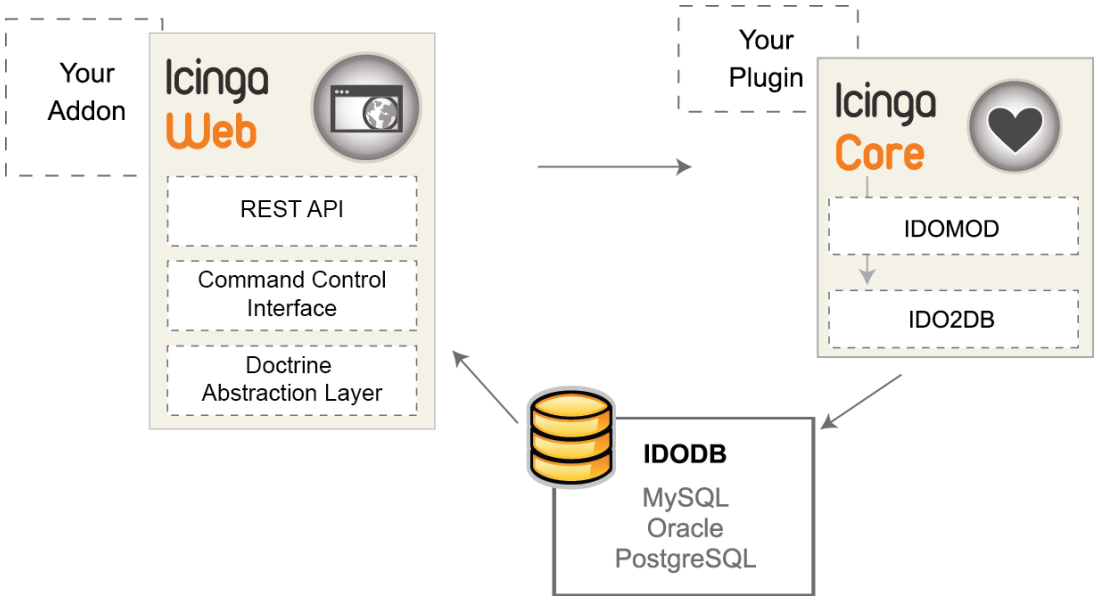


Figura 3.4: Arquitectura do Icinga [38]

O Icinga Core gere as tarefas de gestão. Recebe os resultados das verificações de serviços dos vários *plug-ins* e envia-os para o IDODB através da interface *Icinga Data Out MODule* (IDOMOD) e do serviço *Icinga Data Out to DataBase* (IDO2DB), com *Secure Sockets Layer* (SSL). O IDODB e o IDOMOD podem estar separados para distribuição de dados (e processos) por vários servidores.

O IDODB contém os dados de monitorização do Icinga Core, interpretados pelo Icinga Web, e oferece suporte para MySQL, Oracle e PostgreSQL.

O Icinga Web, composto por três camadas, é uma interface Web baseada em *Asynchronous JavaScript And XML* (AJAX) que permite adicionar extras com funcionalidades acrescidas, e enviar comandos para o Icinga Core. A *Doctrine Abstraction Layer* (DAL) recolhe os dados de monitorização da base de dados, e oferece vistas *Doctrine Query Language* (DQL) para os desenvolvedores e utilizadores experientes desenharem módulos próprios para obter dados de sistemas e bases de dados externos. A *REpresentational State Transfer Application Programming Interface* (REST API), conjuntamente com a DAL, permite aos utilizadores devidos a leitura de dados do IDODB em *JavaScript Object Notation* (JSON) ou XML via HTTP. A Command Control Interface permite correr comandos *Secure SHell* (SSH) pelo Icinga Web, locais ou remotos, através da execução de binários respectivos [39].

3.3.2 Nagios Core

O Nagios Core é uma plataforma de gestão *open-source* configurável, desenhada para verificar o estado de qualquer tipo de dispositivo de rede e certificar de que os serviços nele monitorizados funcionam como devido.

É essencialmente composto por três blocos de configuração: Hosts, Services e Contacts. Hosts são computadores físicos ou virtuais acessíveis através da rede. Services representam aspectos de monitorização relevantes desde a verificação ICMP à leitura de um OID, ou vários, via SNMP. Contacts são, na sua essência, utilizadores alvo de notificações na ocorrência de problemas ou situações particularmente equivalentes.

Oferece um leque de *plug-ins* que permitem verificar, analisar dados e indicar a violação, ou não, de limites estipulados com códigos de retorno Unknown, Ok, Warning e Critical como representa a Figura 3.5.

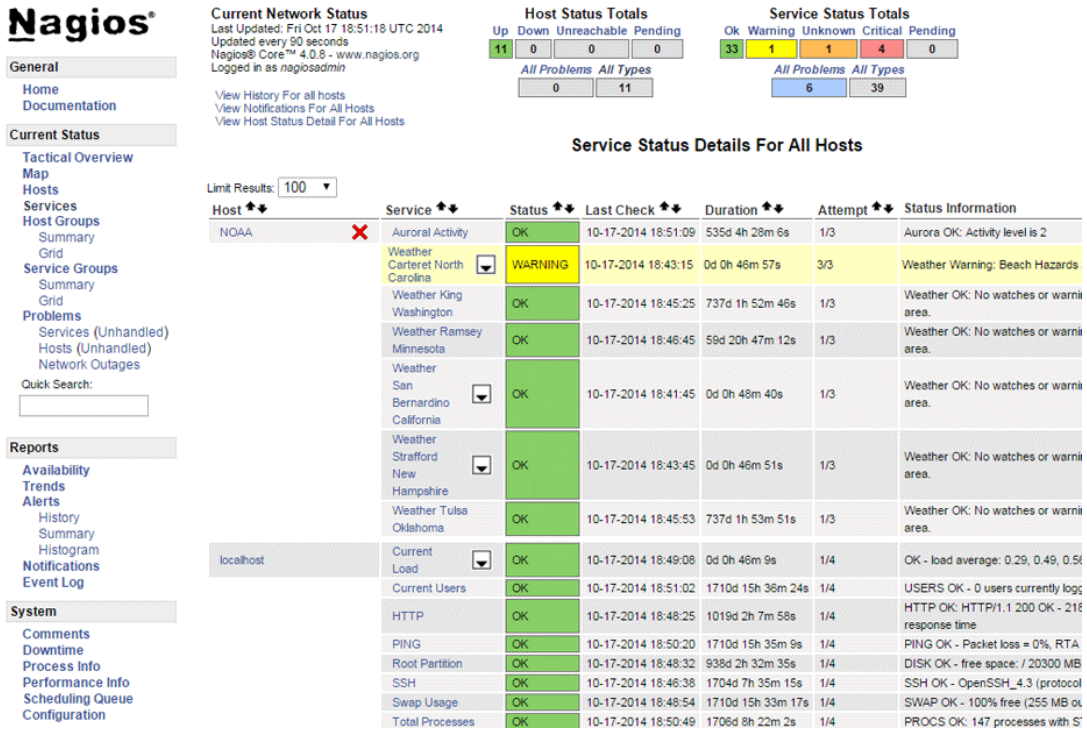


Figura 3.5: Interface Web do Nagios Core

A utilização de códigos evita a interpretação de valores numéricos variados, e permite a geração automática de relatórios que agrupam um conjunto de serviços com estados significativos para o gestor.

Os *plug-ins* genéricos são a base de desenvolvimento para os *plug-ins* particulares. A inclusão de *plug-ins* particulares permite refinar a verificação de serviços, configurar o período de notificações específicas para utilizadores igualmente específicos, gerar relatórios de estatísticas sobre dados recolhidos, e outros [40] [41].

A primeira versão foi lançada a 24 de Novembro de 2002, e a última versão a 19 de Agosto de 2015. A referência temporal ilustrada na Figura 3.6 data o lançamento das versões mais importantes.

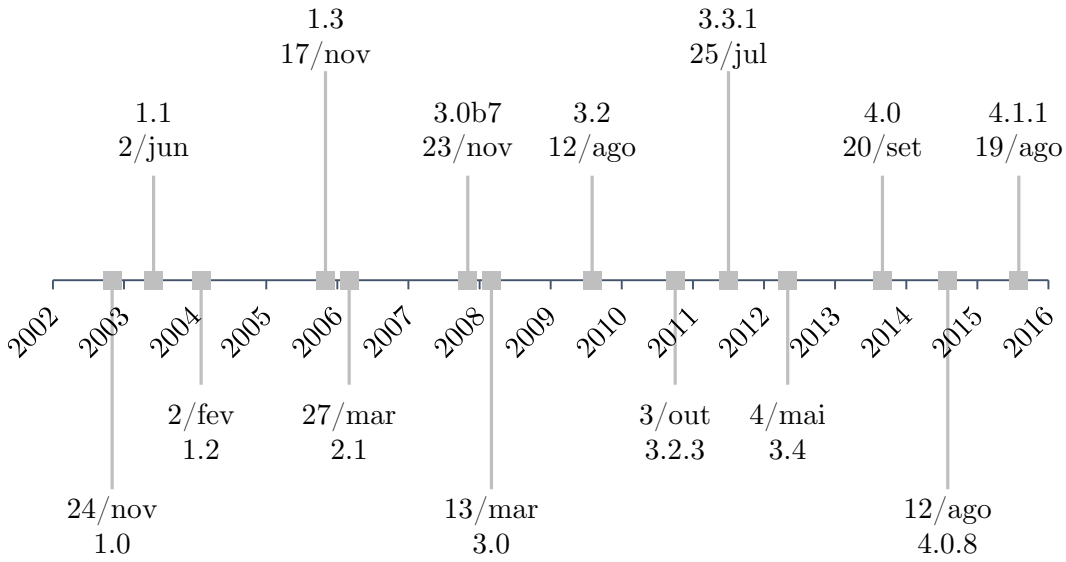


Figura 3.6: Evolução do Nagios Core

As características da plataforma Nagios Core são:

- Monitorização de serviços de rede
- Monitorização de componentes dos dispositivos
- Verificação de serviços configurável
- Verificação de serviços paralela
- Detecção e distinção de dispositivos em baixo, ou inacessíveis
- Notificações por SMS e correio electrónico
- Execução de comandos automática
- Rotação do ficheiro de registo automática

Os componentes arquitecturais que sustentam as características descritas são os *Plug-ins*, os Agentes e a base de dados *Round-Robin Database* (RRD) representada por RRD Database.

Os *Plug-ins* são responsáveis pela verificação de serviços activa, pela análise e devolução de dados Performance Data ao Nagios Core. Permitem a integração e configuração de notificações através de SMS e correio electrónico.

Os Agentes, presentes nos dispositivos remotos, são essencialmente responsáveis pela verificação passiva de serviços e recursos locais, pela análise e devolução de dados Performance Data ao Nagios Core.

A base de dados RRD Database é onde são armazenados os dados Performance Data gerados pelos *Plug-ins* e pelos Agentes envolvidos. Estes dados são

utilizados, posteriormente, para representações gráficas visíveis através da interface Web.

A Figura 3.7 ilustra os princípios de funcionamento entre os componentes abordados.

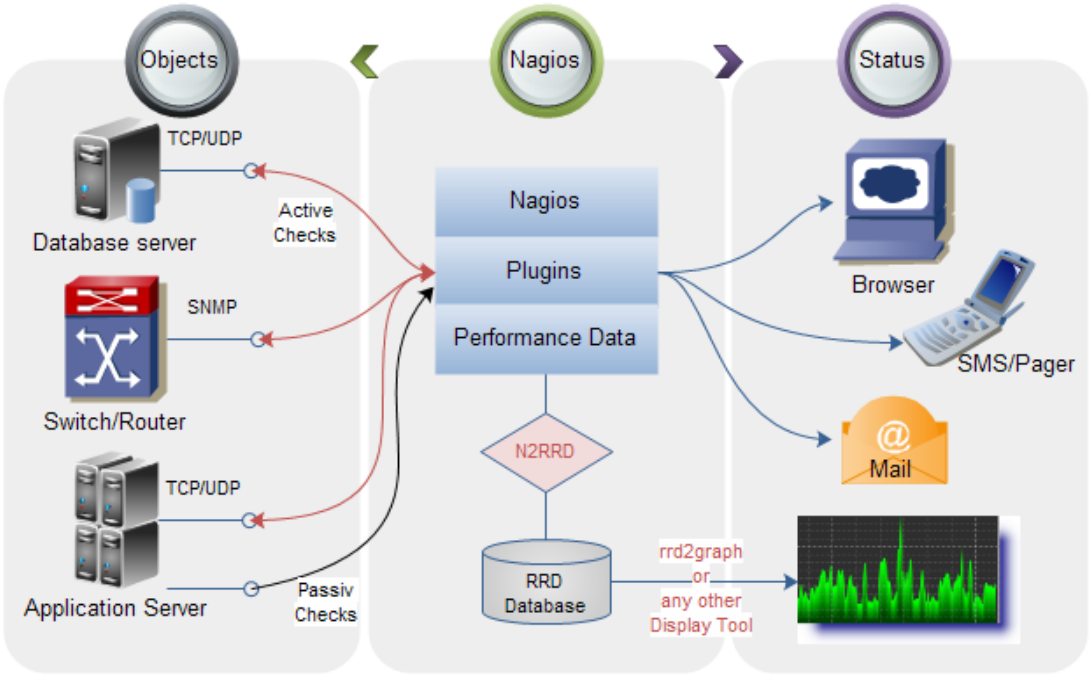


Figura 3.7: Arquitectura do Nagios Core [42]

3.3.2.1 NRDP, NRPE e NSClient++

Exemplos comuns de agentes de monitorização remota desenvolvidos para sistemas específicos são: NRDP, NRPE, e NSClient++.

O NRDP constitui um mecanismo de transporte de dados flexível que oferece uma arquitectura extensível, personalizável conforme as necessidades dos gestores. Utiliza portas e protocolos Web comuns, permite verificações passivas e a execução de comandos remota [43].

O NRPE permite executar *plug-ins* no dispositivo remoto para monitorização de recursos locais como a sobrecarga de *Central Processing Unit* (CPU) e a utilização de memória, normalmente inacessíveis por dispositivos externos. O Nagios executa o *plug-in* check_nrpe, que comunica com o NRPE. O NRPE executa o *plug-in*

remoto indicado e devolve o resultado ao check_nrpe que, por sua vez, devolve o resultado ao Nagios [44].

O NSClient++ deve ser instalado e configurado em dispositivos Windows para monitorização de métricas de sistema, serviços, processos e outros dados de interesse. A configuração comporta um conjunto de secções descritas no ficheiro de configuração NSC.ini que utiliza, por defeito, a porta TCP 12489 para comunicar com o *plug-in* check_nt [45] [46].

3.3.3 Zabbix

O Zabbix é uma plataforma de gestão *open-source* que permite verificar a disponibilidade e a integridade de servidores, e monitorizar componentes de rede particularmente relevantes. A monitorização pode ser activa (com *traps*) ou passiva (com *polling*), devidamente configurada pela interface Web representada na Figura 3.8.

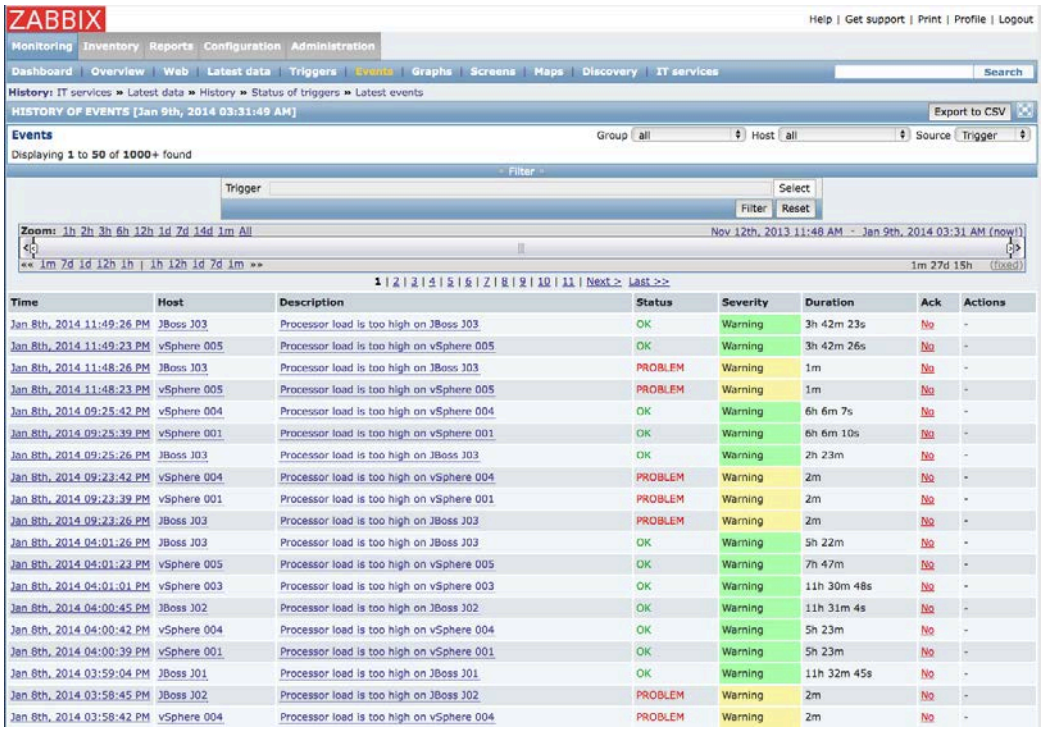


Figura 3.8: Interface Web do Zabbix

A interface Web permite visualizar relatórios com estatísticas sobre dados armazenados, e oferece estratégias de configuração flexíveis para o envio de notificações por correio electrónico [47]. Permite a configuração de *scripts* com comandos executados remota e automaticamente sempre que necessário, e quando a sequência destes é sabida (à partida).

A referência temporal ilustrada na Figura 3.9 data o lançamento das versões mais importantes.

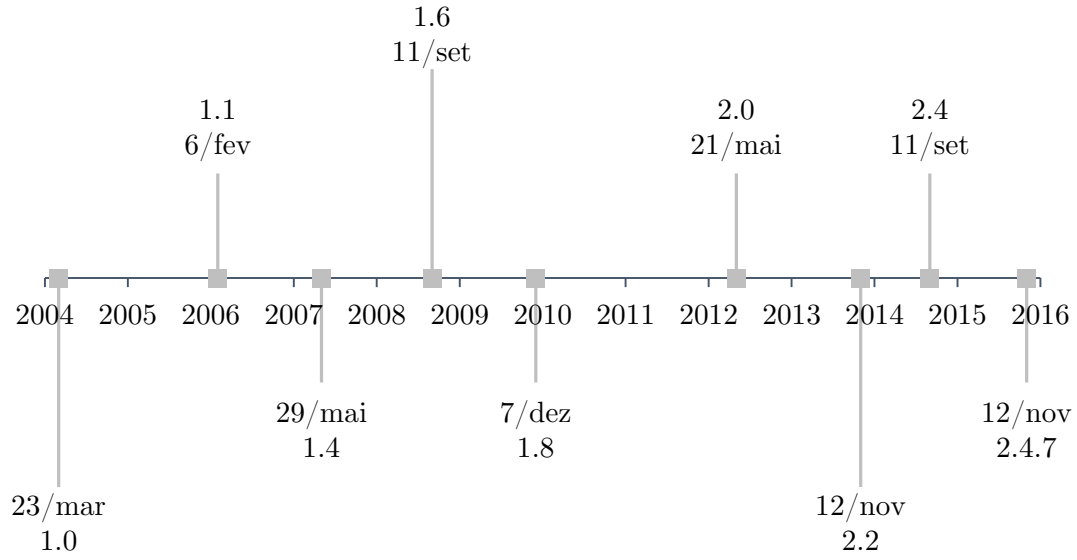


Figura 3.9: Evolução do Zabbix

As características da plataforma Zabbix são:

- Solução integrada
- Monitorização de aplicações
- Suporte para pequenos e grandes ambientes
- Arquitecturas variadas
- Execução de comandos remota
- Gráficos personalizáveis
- Mapas de rede personalizáveis
- Monitorização distribuída com *proxies*

Os componentes arquitecturais que sustentam as características descritas são a base de dados Database, o servidor Zabbix Server, o Zabbix Proxy e a base de dados Proxy Database, o agente Zabbix Agent, e a interface Web centralizada Zabbix Web Interface, interligados como mostra a Figura 3.10.

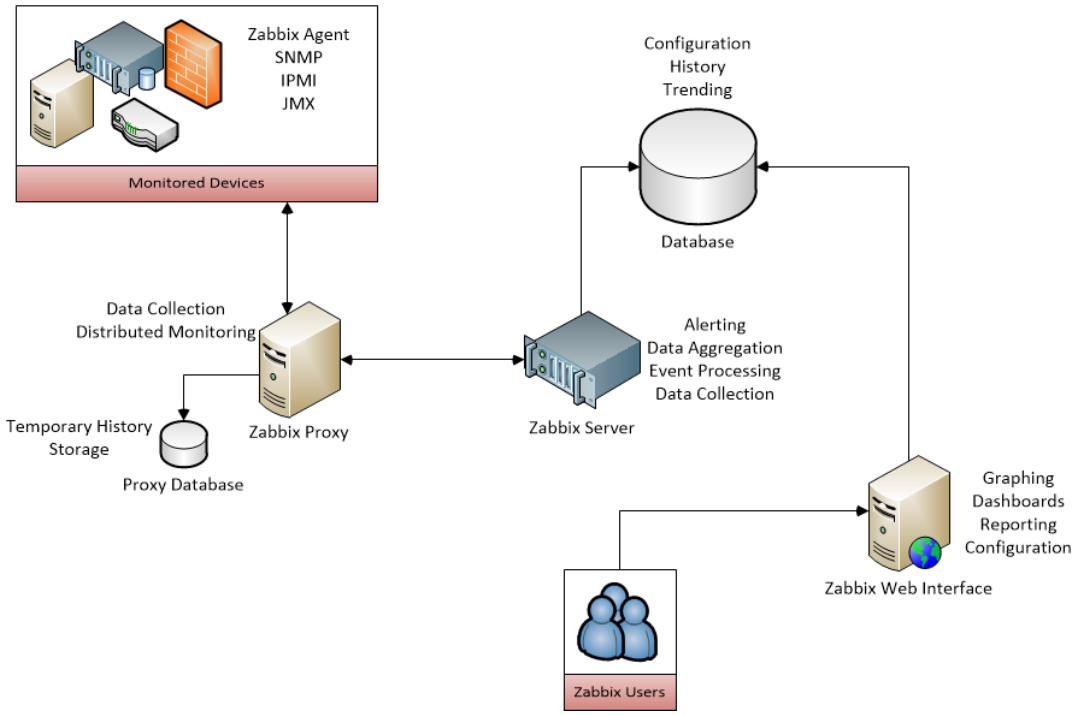


Figura 3.10: Arquitectura do Zabbix [48]

Database é a base de dados central onde são armazenadas as configurações, os dados de monitorização e a previsão de tendências dos mesmos ao longo do tempo. Com suporte para vários *back-ends* [49], armazena também os dados dos vários Zabbix Proxy configurados.

O Zabbix Server é a unidade de processamento central. Pede, recebe e agrega os dados de monitorização dos vários Zabbix Proxy configurados. Analisa-os, despoleta *triggers* na violação de *thresholds*, gera eventos e notificações de acordo com a situação e configuração registados.

O Zabbix Proxy é um processo especial (*daemon process*) que realiza parte das tarefas do Zabbix Server, útil para distribuição de carga. Os dados de monitorização são armazenados na Proxy Database, temporariamente, e enviados para o Zabbix Server (e consequentemente para a Database) quando pertinente.

O Zabbix Agent, suportado por várias plataformas, é um processo implantado nos dispositivos para monitorização activa de aplicações e recursos locais como disco, memória e sistemas de ficheiros [50]. É idealmente combinado com o Zabbix Proxy para monitorização distribuída.

A Zabbix Web Interface, num servidor Web com *PHP: Hypertext Preprocessor* (PHP), permite visualizar o quadro resumo da rede monitorizada, os gráficos e os relatórios gerados, e refinar configurações eventualmente inadequadas.

3.3.3.1 Zabbix Agent

O Zabbix Agent funciona em modo passivo ou em modo activo, como mostra a Figura 3.11.

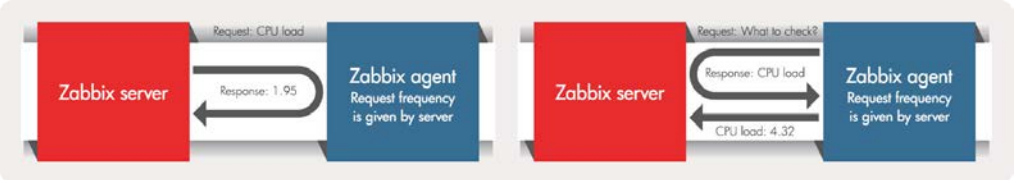


Figura 3.11: Modos do Zabbix Agent [50]

O modo passivo (à esquerda) permite a recolha de dados com intervalos de tempo flexíveis. É naturalmente intuitivo em termos de pedido/resposta, é mais fácil de configurar, de detectar e resolver problemas.

O modo activo (à direita) é mais seguro, pode ser utilizado na presença de *Network Address Translation* (NAT), e reduz a sobrecarga no servidor Zabbix Server [50].

3.3.4 Zenoss Core

O Zenoss Core é uma plataforma de gestão *open-source* que verifica a disponibilidade dos recursos de rede, e oferece toda uma gestão de desempenho e gestão de problemas para resolução de situações irregulares através da interface Web representada na Figura 3.12.

Figura 3.12: Interface Web do Zenoss

A verificação de recursos pode ser feita com *plug-ins* do Nagios, e é essencialmente baseada em SNMP, *Windows Management Instrumentation* (WMI), Syslog, Telnet e SSH.

Corre no servidor de aplicações Web denominado Zope, baseado em Python, e segue o conceito de base de dados de gestão de configuração *Configuration Management DataBase* (CMDB) descrito pela *Information Technology Infrastructure Library* (ITIL). Os objectos (em Python) e os estados respectivos são armazenados na base de dados de objectos *Zope Object DataBase* (ZODB) (orientada a objectos), que representa a CMDB armazenada em MySQL [51].

A referência temporal ilustrada na Figura 3.13 data o lançamento das versões mais importantes.

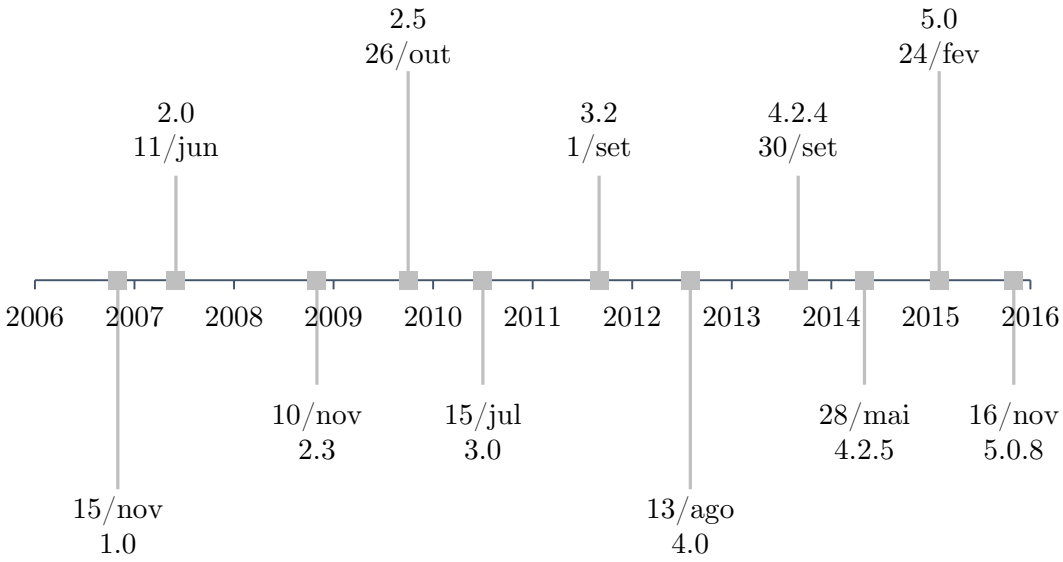


Figura 3.13: Evolução do Zenoss Core

As características da plataforma Zenoss Core são:

- Gráficos/*dashboards* dinâmicos e compartilháveis
- Solução integrada
- Arquitectura com escalabilidade horizontal
- Suporte de *snapshot*, *commit* e *rollback*
- Extensível com *plug-ins* Modeler/ZenPack
- Gestão de *triggers* e *thresholds*
- Notificações por SMS e correio electrónico
- Relatórios personalizáveis

Os componentes arquitecturais basilares que sustentam as características descritas são o Docker, o Control Center e os Collectors, interligados como mostra a Figura 3.14.

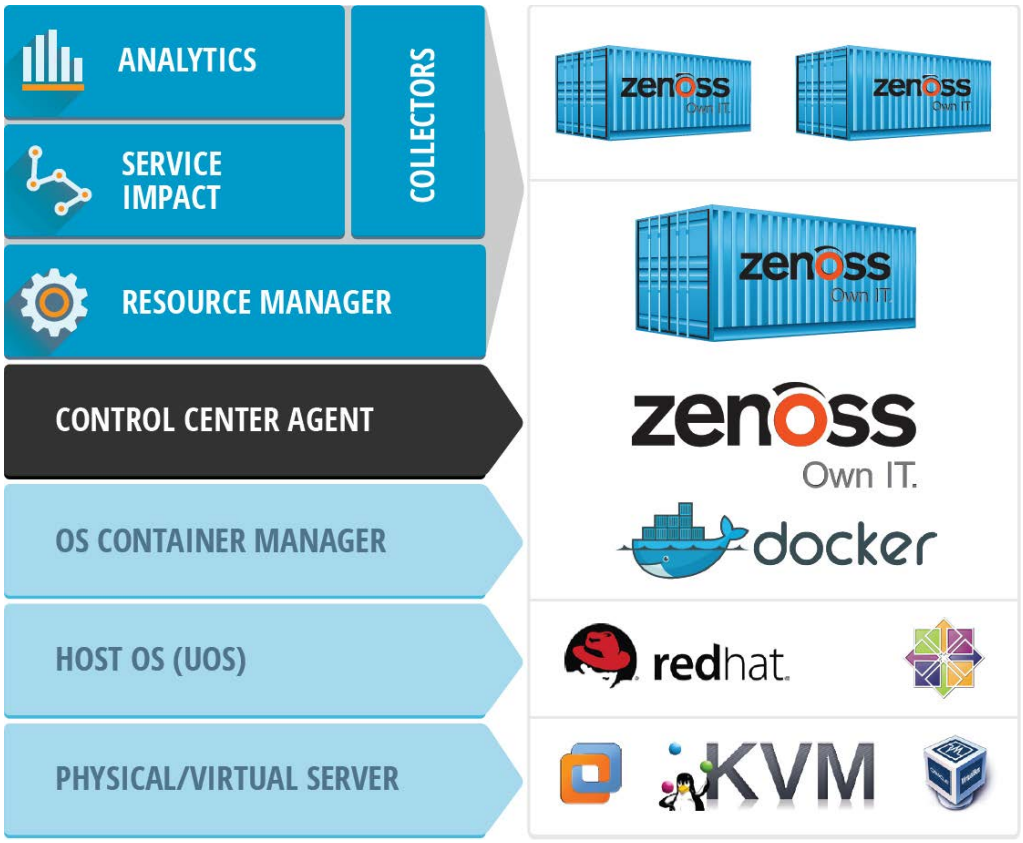


Figura 3.14: Anatomia do Control Center [52]

O Analytics, o Service Impact e o Resource Manager estão apenas disponíveis na versão comercial Zenoss Service Dynamics.

O Docker é uma plataforma aberta que permite aos desenvolvedores e administradores de sistemas implantar, transportar e correr aplicações distribuídas. Os componentes correm em contentores que comportam as dependências exigidas. Os contentores evitam qualquer tipo de conflito, pois estão isolados do dispositivo real onde residem. As actualizações aos contentores tornam-se permanentes com *commit*, ou imediatamente descartadas com *rollback* voltando ao estado anterior. São, por isso, leves e muito rápidas de se aplicar [53].

O Control Center é uma plataforma eficiente para implantar aplicações e serviços respectivos, com ficheiro de registo centralizado para ajudar na detecção de erros. Indica serviços com problemas, permite alocar e realocar serviços (componentes que correm em contentores Docker) em situações irregulares, registar métricas e administrar o Zenoss Core juntamente com os vários serviços associados.

Os Collectors recolhem os dados dos dispositivos monitorizados através dos protocolos comuns. Um *collector* pode monitorizar até 100.000 tipos de dados num intervalo de *polling* de 5 minutos. A carga é distribuída automaticamente, e é possível adicionar *collectors* em situações de sobrecarga como representa a Figura 3.15.

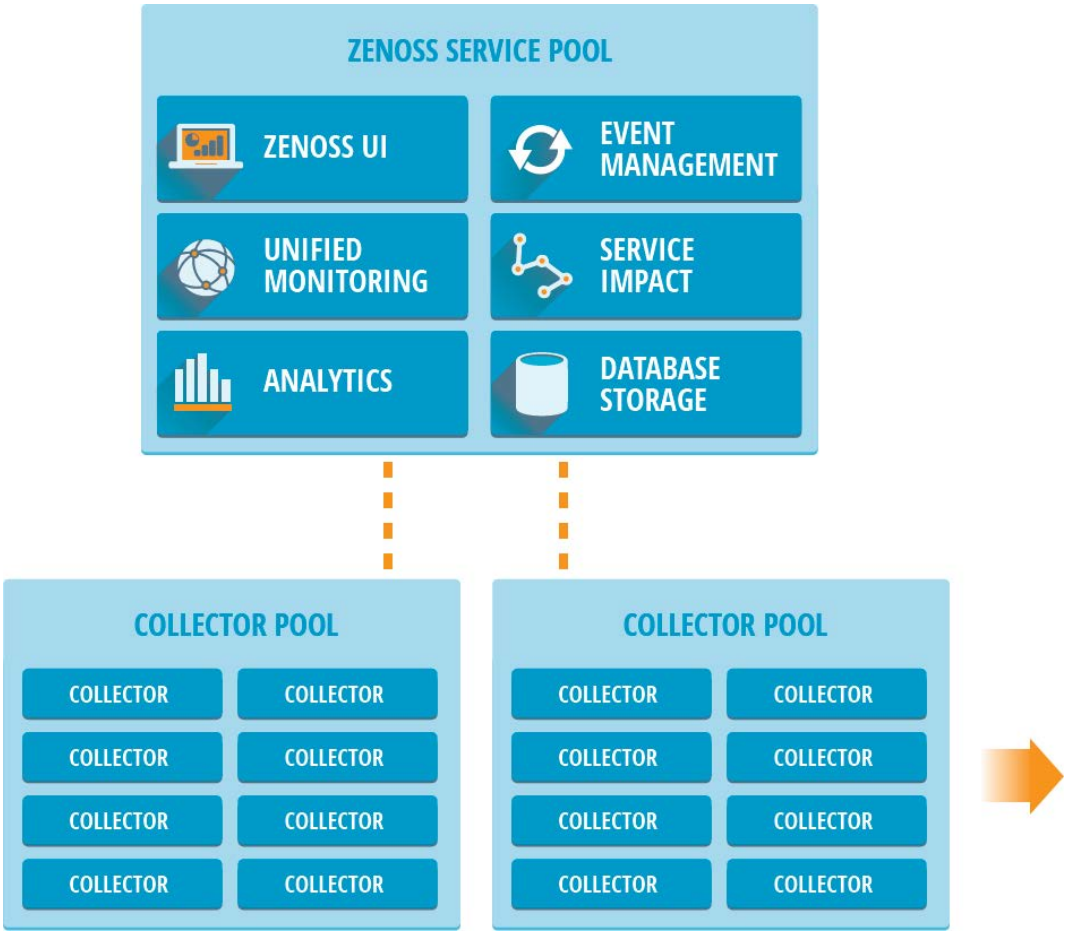


Figura 3.15: Zenoss no Control Center [52]

É utilizada uma Resource Pool, ou várias, na implantação de aplicações e serviços respectivos. As *resource pools* partilham recursos de computação e armazenamento dinamicamente, e permitem escalar de acordo com as exigências de monitorização. Os serviços são distribuídos pelos recursos da Resource Pool à qual estão associados onde podemos, em situação de sobrecarga, adicionar novos recursos e aumentar o número de instâncias dos serviços sobrecarregados (redistribuídas pelos novos recursos automaticamente).

Na Figura 3.15, Zenoss UI é a interface Web que apresenta os recursos monitorizados, a gestão centralizada de eventos, o estado da infra-estrutura, os serviços configurados e as análises de desempenho realizadas.

A Unified Monitoring descobre, modela, aplica *monitoring templates*[1] e monitoriza recursos. A modelação permite o levantamento de aplicações e componentes como interfaces de rede, serviços e processos em execução.

A Event Management processa mais de 100 milhões de eventos por dia, com um processador de eventos desenhado para grande volume de dados. Oferece filtros, agrupa eventos duplicados com base num contador e limpa-os automaticamente na correcção de problemas [52].

O Analytics e o Service Impact estão apenas disponíveis na versão comercial Zenoss Service Dynamics.

Os dados obtidos em tempo real são armazenados numa base de dados Redis local, enviados continuamente para a base de dados OpenTSDB. Os dados recebidos pela base de dados OpenTSDB são escritos na base de dados Hbase[2], da Apache, como mostra a Figura 3.16.

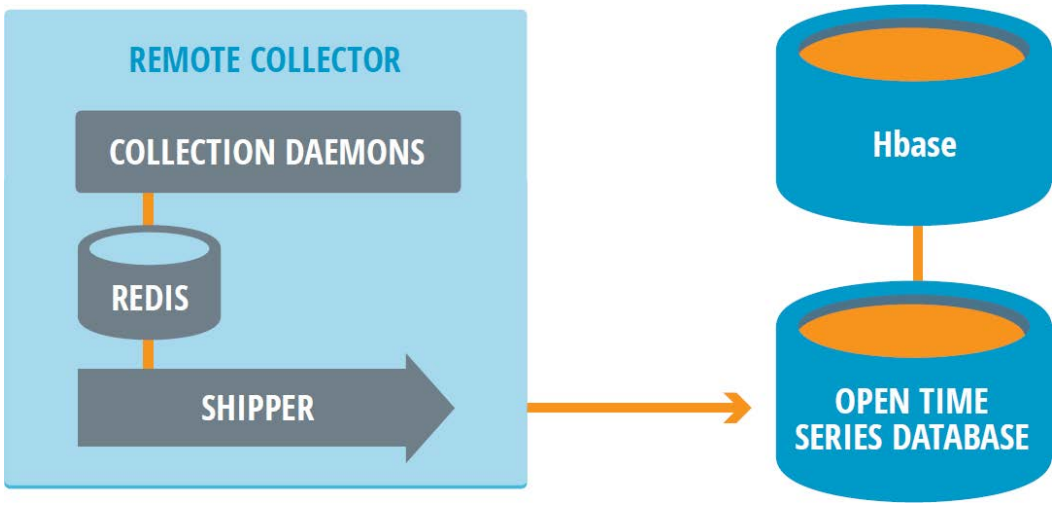


Figura 3.16: Armazenamento de dados centralizado [52]

[1]Modelo de configuração com parâmetros predefinidos para dados de monitorização particularmente relevantes de um ou mais tipos de dispositivos específicos.
[2]Hadoop *database*

A adopção de bases de dados não-relacionais contribui para a escalabilidade horizontal, que permite utilizar (e distribuir carga por) vários servidores [54]. Esta solução, desenhada para grande volume de dados, permite a recolha de dados com maior nível de granularidade (no intervalo de tempo pretendido). Permite ajustar, também e automaticamente, o intervalo de recolha de dados para obter mais informação sobre um evento na violação de um *threshold* ou no despoletar de um *trigger*.

3.4 Quadro Comparativo

São vários os campos da Tabela 3.1 que colocam as plataformas de gestão *open-source* lado a lado e permitem compará-las em termos do que disponibilizam, ou não. Aqui, temos:

- Relatórios IP SLA: Suporte IP *Service-Level Agreement* (SLA) da Cisco
- Grupos de dispositivos: Agrupamento lógico de dispositivos
- Tendências: Provisão de tendências dos dados de rede ao longo do tempo
- Previsão de tendências: Com algoritmos para prever estatísticas de rede
- Descoberta automática: Detecção de dispositivos com SNMP
- Agente(s): Integração de agentes nos dispositivos monitorizados
- SNMP: Obtenção de dados SNMP e geração de relatórios de estatísticas
- Syslog: Envio e recepção de mensagens Syslog
- *Plug-ins*: Integração de extras para funcionalidades acrescidas
- *Triggers*: Notificação na violação de *thresholds*
- Aplicação Web: Controlo pela interface Web
- Monitorização distribuída: Distribuição de carga por vários servidores
- Inventário: Registo de *hardware* e *software* dos dispositivos monitorizados
- Plataforma: Linguagens de programação dos componentes fundamentais
- Armazenamento de dados: Métodos de armazenamento de dados
- Licença: Licença de *software*
- Mapas: Inclusão de mapas de rede, com dispositivos e ligações entre eles
- Controlo de acesso: Com base no utilizador e/ou grupos de utilizadores
- IPv6: Monitorização SNMP com IPv6
- Versão: Última versão de lançamento
- Data: Da última versão de lançamento

Tabela 3.1: Comparação das plataformas de gestão (baseada em [55])

	Icinga	Nagios Core	Zabbix	Zenoss Own IT.
Relatórios IP SLA	*Plug-in*	*Plug-in*	✓	✓
Grupos de dispositivos	✓	✓	✓	✓
Tendências	✓	✓	✓	✓
Previsão de tendências	×	×	×	✓
Descoberta automática	*Plug-in*	*Plug-in*	✓	✓
Agente(s)	✓	✓	✓	✓
SNMP	*Plug-in*	*Plug-in*	✓	✓
Syslog	*Plug-in*	*Plug-in*	✓	✓
Plug-ins	✓	✓	✓	✓
Triggers	✓	✓	✓	✓
Aplicação Web	Limitada	Limitada	✓	✓
Monitorização distribuída	✓	✓	✓	✓
Inventário	*Plug-in*	*Plug-in*	✓	✓
Plataforma	C	C PHP	C PHP	Java Python
Armazenamento de dados	MySQL Oracle PostgreSQL	Ficheiro SQL	MySQL Oracle PostgreSQL	MySQL OpenTSDB ZODB
Licença	GPL	GPL	GPL	Comercial GPL
Mapas	✓	✓	✓	✓
Controlo de acesso	✓	✓	✓	✓
IPv6	✓	✓	✓	✓
Versão	2.4	4.1.1	2.4.7	5.0.8
Data	nov/2015	ago/2015	nov/2015	nov/2015

É através da Tabela 3.1 que se cruzam as plataformas de gestão *open-source* abordadas com os critérios de selecção descritos na secção 3.2.

Quer o Icinga quer o Nagios Core, como o Zabbix e o Zenoss Core oferecem todo um conjunto de funcionalidades satisfatório para os desafios da gestão de redes que se fazem sentir nas infra-estruturas de telecomunicações actuais. Qualquer uma destas plataformas de gestão são gratuitas e tiram partido da Web para autenticação e autorização de utilizadores, apresentação e edição de conteúdo.

A extensibilidade é garantida através de *plug-ins* compatíveis, na maior parte dos casos, entre cada uma destas plataformas. A interoperabilidade é favorecida com a utilização de padrões abertos naturalmente preferidos em aplicações de natureza *open-source*.

3.5 Conclusão

Neste capítulo, foi feito o levantamento de 4 plataformas de gestão *open-source* populares no mercado da gestão de redes e sistemas, descritas as características principais, as arquitecturas que as sustentam e elaborado um quadro comparativo para análise de informação relevante sobre cada uma delas.

Uma plataforma de gestão comporta todo um conjunto de aspectos de funcionalidade, extensibilidade, interoperabilidade, segurança, tecnologia, aplicações e custo. Automatiza a tarefa de gestão, identifica problemas recorrentes, propõe e inicia acções correctivas que suprimem a intervenção manual do gestor responsável tanto quanto possível. Deve oferecer uma arquitectura modular e escalável, e permitir o desenvolvimento e a integração de módulos funcionais que interajam com módulos de sistemas terceiros. Deve ser dotada de um sistema de autenticação e autorização de utilizadores, e registar as operações de gestão por eles efectuadas.

O capítulo seguinte expõe um caso real demarcado pelas limitações e problemas técnicos da plataforma de gestão de redes em uso. Comporta o levantamento da infra-estrutura de telecomunicações para a qual é seleccionada a plataforma de gestão *open-source* abordada que melhor se enquadra no cenário apresentado.

Capítulo IV

Caso de Estudo: NOS Madeira

É com base numa infra-estrutura de telecomunicações real e nos problemas que uma plataforma de gestão antiga causa que se apresenta e descreve o NOC da NOS Madeira. Aqui, é feito um levantamento da organização interna em termos de utilizadores, grupos de utilizadores, dispositivos, grupos de dispositivos, serviços, *thresholds*, *triggers* e notificações importantes, e expostas as limitações da plataforma SNMPc que dão origem aos requisitos e funcionalidades a considerar durante a análise e comparação das plataformas abordadas para seleccionar a plataforma de gestão *open-source* que melhor se enquadra no cenário em questão.

4.1 Introdução

A NOS Madeira é a infra-estrutura de telecomunicações da NOS responsável por fornecer soluções fixas e móveis de última geração, televisão, Internet, voz e dados para os segmentos de mercado pessoal, residencial e empresarial em toda a ilha da Madeira.

Conta com um elevado número de dispositivos que têm de estar idealmente disponíveis 24 horas por dia, onde a falha de serviço é, naturalmente, inconveniente e indesejável. Estes dispositivos estão agrupados por categoria/sistema, e associados a grupos de utilizadores por eles responsáveis.

Recorre, portanto, a uma plataforma de gestão de redes essencialmente baseada em SNMP para a recolha e análise de dados de monitorização, configuração de *thresholds*, *triggers* e envio de notificações para os gestores de rede sempre que uma situação imprevista coloque ou possa colocar em causa a qualidade do serviço fornecido.

4.2 Organização

Os dispositivos, os grupos de utilizadores, os serviços, os *thresholds*, os *triggers* e notificações da infra-estrutura de telecomunicações da NOS Madeira estão organizados como descrito nas secções que se seguem.

4.2.1 Dispositivos

A maior parte de dispositivos como *switches* de camada 3, *switches* Gigabit Ethernet, *routers*, servidores, *Virtual Machine* (VM) *containers*, analisadores de tráfego/gestores de largura de banda, e dispositivos ambientais para medição de temperatura e humidade está organizada no SNMPc como representa a Figura 4.1.

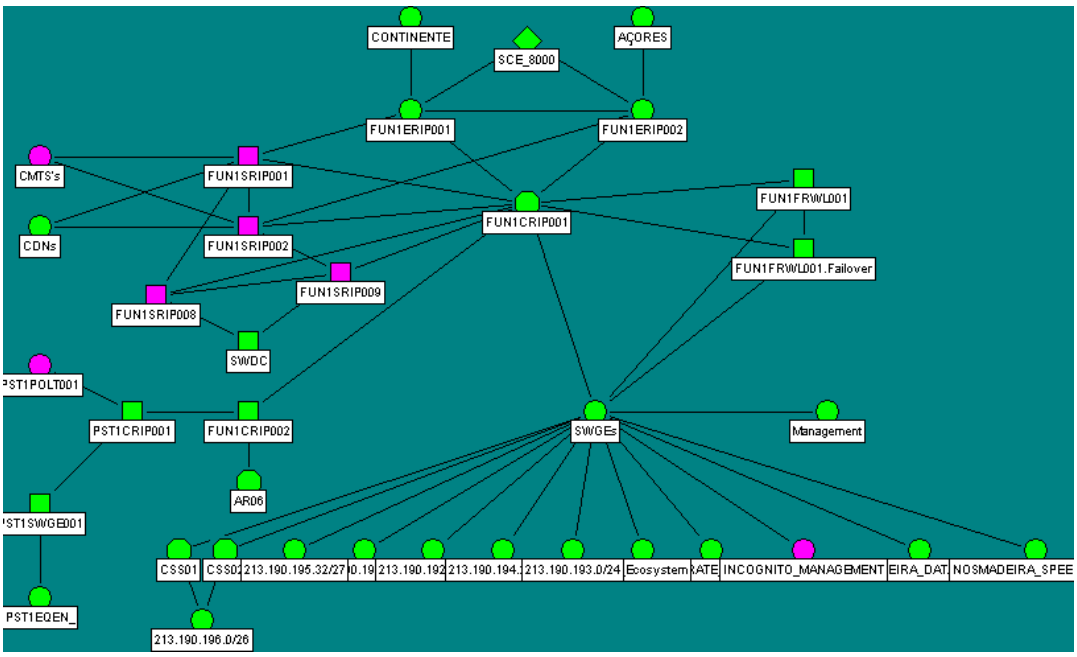


Figura 4.1: Mapa de rede parcial no SNMPc

O SNMPc não permite agrupar dispositivos quer lógica quer fisicamente. São vários, contudo, os dispositivos responsáveis por serviços basilares ao correcto funcionamento da infra-estrutura de telecomunicações da NOS Madeira que compõem grupos heterogéneos onde Cisco e Dell são as marcas que se destacam, ideal e mentalmente organizados como mostra a Tabela 4.1.

Tabela 4.1: Dispositivos de rede em estudo

Dispositivo	**Grupo**	**Localização**
FUN1CMTS001 ... FUN1CMTS006	CMTS	FUN1
FUN1CRIP001	CRIP	
FUN1DPIN001	DPI	
Ar_Condicionado_03 FUN1SENSOR001 Temperatura Temperatura_Hosting	Environment	
FUN1ERIP001 FUN1ERIP002	ERIP	
GGC 01 GGC 02 GGC 03	GGC	
Acores Continente	ICMP	
IMPERIAL1_ESXi IMPERIAL2_ESXi IMPERIAL1_iDRAC IMPERIAL2_iDRAC	IMPERIAL	
Incognito-01 ... Incognito-06	Incognito	
Fulliautomatix Krusty.WebMail1 Moe.WebMail2 MXST1 MXST11 MXST12 SV32.Mail1 SV34.Mail2 SV40.Mail3 SV39.MX1 SV36.MX2	Mail	
FUN1EQEN005 FUN1EQEN006	PDU[1]	

[1]Power Distribution Unit

FUN1EQEN062		
FUN1SRIP001 FUN1SRIP002 FUN1SRIP008 FUN1SRIP009	SRIP	
FUN1SWDC001 ... FUN1SWDC009	SWDC	
FUN1SWGE007 FUN1SWGE021 FUN1SWGE023 FUN1SWGE024 FUN1SWGE032	SWGE	
FUN1UPS001 FUN1UPS002	UPS	

Aqui, *Cable Modem Termination System* (CMTS) ou concentradores de *cable modems* faz referência aos 6 dispositivos utilizados para fornecer serviços de dados de alta velocidade como o *Voice over IP* (VoIP) para clientes com Internet por cabo.

Incognito fornece o serviço *Configuration File Management* (CFM) para criar os ficheiros de configuração necessários dinamicamente, evitando armazená-los. Os Incognito-03 e Incognito-04 fornecem *Domain Name System* (DNS) para resolução de nomes de domínio, *Dynamic DNS* (DDNS), e permitem a transferência de zonas. Os Incognito-05 e Incognito-06 fornecem *Multimedia Provisioning System* (MPS) com provisões de configurações multimédia *PacketCable* (PC) e *Session Initiation Protocol* (SIP) precisas para os dispositivos dos clientes, e armazenamento destas. Os Incognito-01, Incognito-02, Incognito-05 e Incognito-06 fornecem *Dynamic Host Configuration Protocol* (DHCP) para gestão automática da alocação de endereços IP dinâmicos, configurações de terminais *Data Over Cable Service Interface Specification* (DOCSIS) segundo políticas definidas pelo utilizador, e *Trivial File Transfer Protocol* (TFTP) para a transferência de ficheiros de configuração.

SWDC e SWGE de SWitch Data Center e SWitch Gigabit Ethernet, respectivamente, representam os *switches* aos quais se ligam os dispositivos terminais do *datacenter* (e de gestão deste).

SRIP, de Switch Router IP, são os dispositivos aos quais se ligam os dispositivos CMTS. Os dispositivos ERIP, de Edge Router IP, estão a cargo das ligações com Portugal continental e com a ilha dos Açores. Os dispositivos CRIP, de Core

Router IP, são os *routers* aos quais se ligam os dispositivos ERIP e SRIP descritos neste parágrafo.

Deep Packet Inspection (DPI) é essencialmente responsável pela gestão de largura de banda dos dispositivos dos clientes, atribuição de prioridades, filtragem e análise de pacotes.

Environment é responsável pela medição de temperatura, humidade e nível de água no *datacenter*, com sensores para o efeito.

Google Global Cache (GGC) contém os dispositivos que fazem parte da plataforma de distribuição de conteúdo à escala global, do Google. Aqui, o conteúdo e serviços do Google são fornecidos o mais próximo possível dos clientes, com melhor desempenho e menor custo para as operadoras de rede.

ICMP é composto pelos pontos de ligação entre a rede local e as redes do Continente e Açores. A restrição de acesso em causa faz com que apenas o protocolo ICMP esteja disponível para monitorização.

IMPERIAL representa dois VM *containers Elastic Sky X integrated* (ESXi) que servem dispositivos virtuais para os membros do grupo de utilizadores SIs, descrito posteriormente. São controlados por interfaces *integrated Dell Remote Access Controller* (iDRAC), específicas da Dell, que permitem acesso remoto aos recursos através da consola de gestão.

Mail, constituído por FE, *Mail eXchanger* (MX), MXST e WebMail, fornece os serviços de correio electrónico da NOS Madeira. FE, de Front-End, é essencialmente responsável pelos serviços *Internet Message Access Protocol* (IMAP) e *Post Office Protocol 3* (POP3) através dos servidores SV32.Mail1, SV34.Mail2 e SV40.Mail3. MX é responsável pelo serviço *Simple Mail Transfer Protocol* (SMTP) utilizado para receber e enviar mensagens de acordo com as prioridades estabelecidas para os servidores SV39.MX1 e SV36.MX2. MXST, de MX Storage, é responsável pelo armazenamento do correio electrónico dos clientes através dos servidores MXST1, MXST11 e MXST12. WebMail é responsável pelo acesso aos serviços de correio electrónico via Web, amplamente procurado pelos clientes da NOS Madeira.

Os dispositivos PDU e os dispositivos *Uninterruptible Power Supply* (UPS) fornecem e distribuem energia para os dispositivos de toda rede, respectivamente. Os dispositivos UPS estão preparados para fornecer alimentação em situações de falha/corte de energia (geral).

4.2.2 Grupos de Utilizadores

É com base nos grupos de utilizadores que se definem responsabilidades sobre dispositivos ou conjuntos de dispositivos específicos. Os grupos de utilizadores definidos até ao momento, são:

- PIP, de Plataformas e IP: Responsável pelos dispositivos do *datacenter*
- SIs, de Sistemas de Informação: Responsável por bases de dados/facturação
- VoIP: Responsável pelo telefone fixo
- VoD: Responsável pelo serviço de vídeo
- Rede de Acesso: Responsável pelas redes *Hybrid Fiber Coaxial* (HFC) e *Fiber To The Home* (FTTH), e pela rede de acesso exterior (clientes)
- IMC, de Instalação e Manutenção de Clientes: Responsável pelo apoio técnico ao cliente
- 2Linha, de Segunda Linha: Responsável pelo apoio técnico ao cliente no teste e despiste de avarias
- Comercial Empresarial: Responsável pelo segmento empresarial
- Headend, ou "Cabeça de sistema": Responsável pelos dispositivos de energia, temperatura, ar condicionado e relacionados

A Tabela 4.2 mostra a relação entre os dispositivos e os grupos de utilizadores definidos até ao momento, útil para sumarizar as questões de alarmística discutidas posteriormente.

Tabela 4.2: Grupos de utilizadores por dispositivos

Dispositivo	**Grupos de utilizadores**
FUN1SWGE023 FUN1SWGE024 FUN1SWGE032	Comercial Empresarial
Ar_Condicionado_03 FUN1SENSOR001 Temperatura Temperatura_Hosting FUN1EQEN062	Headend e PIP
FUN1EQEN005 FUN1EQEN006 FUN1SWGE007 FUN1SWGE021	SIs

IMPERIAL1_ESXi IMPERIAL2_ESXi IMPERIAL1_iDRAC IMPERIAL2_iDRAC	SIs e PIP
...	PIP

4.2.3 Serviços

A maior parte dos serviços IP monitorizados incide sobre os servidores de correio electrónico. É importante saber quando é que os clientes deixam de conseguir aceder à sua conta, e quando é que deixam de ter espaço livre no servidor.

A verificação do serviço MySQL (porta TCP 3306) previne e reduz as interrupções de serviço, e a verificação do espaço em disco disponível via SNMP auxilia nas tarefas de *housekeeping*.

O envio e a recepção de correio electrónico entre os clientes e os servidores respectivos é igualmente importante, pelo que também se verificam os serviços SMTP, POP3 e IMAP nas portas TCP 25, 110 e 143 (respectivamente).

O acesso aos serviços de correio electrónico pela Web faz com que o serviço HTTP (porta TCP 80) seja verificado com *send/expect strings*, especialmente por abranger a maior fatia de clientes da NOS Madeira.

A Tabela 4.3 resume o conjunto de dispositivos com monitorização activa no que diz respeito aos serviços IP.

Tabela 4.3: Serviços de correio electrónico

Dispositivo	**Serviços**
Fulliautomatix	MySQL
Krusty.WebMail1 Moe.WebMail2	SMTP e HTTP
MXST1 MXST11 MXST12 SV32.Mail1 SV34.Mail2 SV40.Mail3	SMTP, POP3 e IMAP

Uma vez que o SNMPc não permite agrupar dispositivos quer lógica quer fisicamente, a verificação de serviços IP nesta plataforma é feita manualmente (de dispositivo a dispositivo).

4.2.4 Thresholds, Triggers e Notificações

Os *thresholds* e os *triggers* definidos cobrem grande parte dos dispositivos e a totalidade dos serviços descritos na secção 4.2.3. As notificações associadas estão configuradas com um formato específico para que se dê resposta às eventualidades com a maior brevidade possível. É com a combinação dos *thresholds*, *triggers* e notificações que se verifica o estado de interfaces de rede específicas e o estado dos dispositivos em si, via SNMP.

Existem *triggers* específicos para SNMP *up* e *down* com notificações para os grupos de utilizadores Headend, SIs e PIP. Há também outro *trigger* para ICMP, com notificação para o grupo de utilizadores PIP.

Estão definidos dois *thresholds*/*triggers* sobre interfaces de rede com notificação para o grupo de utilizadores PIP. Um das configurações assenta nas interfaces eth1 dos dispositivos Incognito, e outra nas interfaces TenGigabitEthernet1/0/1 e TenGigabitEthernet1/0/2 dos dispositivos SWDC.

Os serviços SMTP, HTTP (e HTTP com *send*/*expect strings*), *HTTP Secure* (HTTPS), POP3, IMAP e MySQL têm também *thresholds*/*triggers* associados com notificação para o grupo de utilizadores PIP.

Os dois *triggers* e notificações restantes, também para o grupo de utilizadores PIP, têm por base um conjunto de métricas avaliadas por *thresholds*. O primeiro *threshold* verifica o espaço disponível em cada partição do disco rígido dos servidores de correio electrónico, e o segundo verifica os níveis de *upstream* e *downstream* (nos dispositivos DPI).

No SNMPc, os *thresholds* pertencem a *triggers* e os *triggers* filtram os eventos. As notificações nesta plataforma são atribuídas manualmente (de evento a evento).

4.3 Problemas

O SNMPc, de 2000, é a plataforma de gestão de redes em uso desenvolvida pela Castle Rock Computing [56]. É comercial, funciona em Windows apenas e inadequada essencialmente pelas razões seguintes:

- Não permite agrupar dispositivos quer lógica quer fisicamente
- Não permite criar nem visualizar gráficos (utilizam o Cacti para isso)
- A verificação de serviços é feita manualmente de dispositivo a dispositivo
- As notificações são atribuídas manualmente de evento a evento
- A alarmística não é fiável (falsos alarmes)

A aplicação tem fraco desempenho e vai abaixo quando há sobrecarga de CPU. A interface de gestão requer elevado custo operacional dada a pouca acessibilidade e a falta de usabilidade desta, pelo que uma nova plataforma de gestão que ultrapasse estes problemas é solução.

4.4 Requisitos Funcionais

A nova plataforma de gestão deve responder aos problemas do SNMPc e facilitar a definição dos utilizadores, grupos de utilizadores, dispositivos, grupos de dispositivos, serviços, *thresholds*, *triggers* e notificações descritos na secção 4.2.

De uma forma geral, a nova plataforma de gestão deve:

R1. Oferecer uma interface simples, com boa usabilidade
R2. Suportar SNMP, ICMP e protocolos relacionados
R3. Proporcionar monitorização distribuída
R4. Aceitar ficheiros MIB de terceiros
R5. Receber e processar *traps*
R6. Indicar o estado dos dispositivos e serviços com cores
R7. Executar comandos remota e automaticamente
R8. Possibilitar a activação/desactivação de notificações
R9. Evitar o envio de notificações duplicadas

Permitir a definição de:

R10. Utilizadores, grupos de utilizadores e permissões respectivas
R11. Dispositivos e grupos de dispositivos
R12. *Monitoring templates* para recolha e monitorização de dados
R13. *Thresholds*/*triggers*
R14. Gráficos personalizados com métricas e *thresholds* associados
R15. Dispositivos dependentes
R16. Serviços e grupos de serviços
R17. Serviços dependentes
R18. Notificações para utilizadores e grupos de utilizadores
R19. Notificações por SMS e correio electrónico

R20. Mapas de redes (manuais)
R21. Relatórios personalizados

Possibilitar ver, em hierarquia:

R22. Utilizadores e grupos de utilizadores
R23. Dispositivos e grupos de dispositivos
R24. Serviços e grupos de serviços
R25. *Monitoring templates* para recolha e monitorização de dados

Monitorizar:

R26. Dispositivos heterogéneos (como sensores)
R27. *Software* instalado nos dispositivos monitorizados
R28. Serviços e conteúdo de serviços (nomeadamente Web)

Ser:

R29. Livre e *open-source*
R30. Altamente disponível

4.4.1 Análise e Comparação

Foi utilizado um cenário de rede simplificado para teste e comparação das plataformas de gestão *open-source* abordadas, durante 4 meses. Mesmo limitado, permitiu desvendar aspectos básicos de instalação, configuração, utilização, manutenção, e destacar as funcionalidades particulares de cada plataforma de gestão mediante o meio monitorizado.

Este cenário, ilustrado na Figura 4.2, é composto por 4 dispositivos virtuais: dois Windows e dois Unix. Estes dispositivos foram configurados no *software* VMware Player, em modo *bridged*, para que tivessem endereços IP da rede local à qual o computador real estivesse ligado.

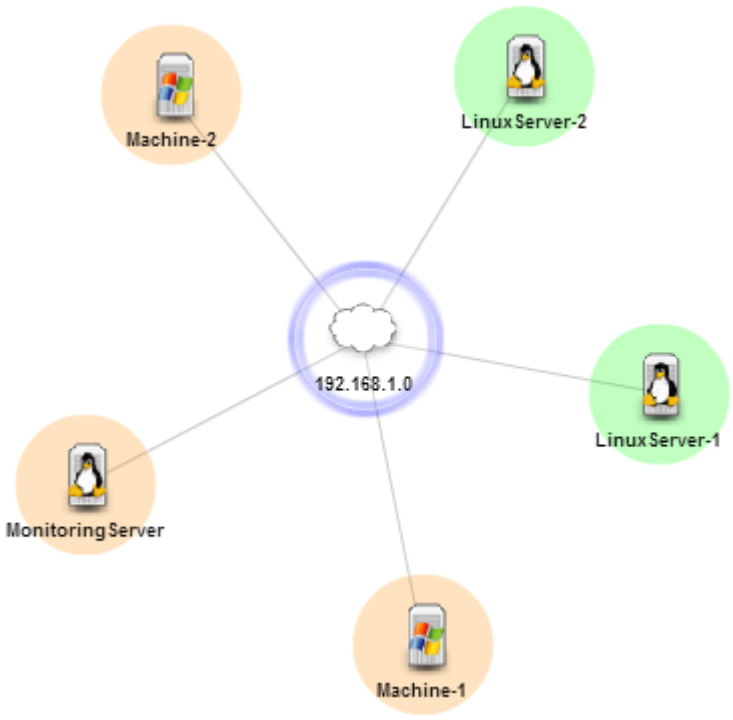


Figura 4.2: Cenário de teste e demonstração

A Tabela 4.4 cruza os requisitos funcionais da secção 4.4 com as funcionalidades de cada uma das plataformas de gestão *open-source* abordadas, testadas e comparadas no cenário de rede da Figura 4.2.

Tabela 4.4: Comparação das plataformas de gestão

	Icinga	Nagios Core	Zabbix	Zenoss Own IT.
R1	✓	✓	✓	✓
R2	✓	✓	✓	✓
R3	✓	✓	✓	✓
R4	✓	✓	✓	✓
R5	✓	✓	✓	✓
R6	✓	✓	✓	✓
R7	✓	✓	✓	✓
R8	✓	✓	✓	✓
R9	✓	✓	✓	✓
R10	✓	✓	✓	✓
R11	✓	✓	✓	✓

R12	✓	✓	✓	✓
R13	✓	✓	✓	✓
R14	✓	✓	✓	✓
R15	✓	✓	✓	✓
R16	✓	✓	✓	✓
R17	✓	✓	✓	✓
R18	✓	✓	✓	✓
R19	✓	✓	✓	✓
R20	✓	✓	✓	✓
R21	✓	×	✓	✓
R22	✓	✓	✓	✓
R23	✓	✓	✓	✓
R24	✓	✓	✓	✓
R25	×	×	✓	✓
R26	✓	✓	✓	✓
R27	✓	✓	✓	✓
R28	✓	✓	✓	✓
R29	✓	✓	✓	✓
R30	✓*	✓*	✓*	✓**

*Implementada pelo utilizador
*Nativa para *resource pools*, implementada pelo utilizador sobre o Control Center

É através dos requisitos, do cenário de teste e demonstração e da comparação entre as plataformas de gestão que se prefere uma plataforma que ofereça um modelo de informação próximo da abordagem *Object-Oriented Programming* (OOP), e que a aplicação de princípios como a herança permita organizar dispositivos de forma hierárquica, categorizar eventos, e agrupar *triggers* com parâmetros conjuntos.

Os grupos de utilizadores identificados requerem a selecção de uma plataforma de gestão que permita adicionar, editar utilizadores e atribuir permissões de acesso para cada um deles tendo, por base, o protocolo *Lightweight Directory Access Protocol* (LDAP). A análise e comparação de funcionalidades, juntamente com o quadro comparativo das plataformas de gestão na secção 3.4, exclui o Icinga, o Nagios (Core) e o Zabbix da lista de prioridades.

O Icinga e o Nagios utilizam ficheiros de texto regulares para qualquer tipo de configuração, passível de erro e rudimentar quando comparado com as restantes soluções (o Nagios requer ferramentas de terceiros para suporte visual em aspectos de configuração). O Zabbix comporta as limitações de escrita em bases de dados

relacionais ao utilizar MySQL, reduzido à escalabilidade vertical do dispositivo de gestão.

O Zenoss Core tira proveito do Docker e do Control Center para maior eficiência em sistemas distribuídos, verificação e manutenção dos *daemons* envolvidos, configuração de instâncias, *backup* e sincronização destes. Oferece uma arquitectura com escalabilidade horizontal, diferente das restantes plataformas, com *resource pools* para auxílio nas tarefas distribuídas desde o *polling* ao envio de correio electrónico. Organiza os dispositivos de forma hierárquica e agrupa os eventos com base no conjunto de propriedades do objecto evento, com contadores para escalar e notificar apenas quando necessário. Permite criar e agrupar *triggers* com parâmetros conjuntos, e associar uma notificação só para duas situações: *up* e *down*.

O Zenoss Core oferece três ambientes de programação Python, em *run-time*. O primeiro para adicionar e modificar funcionalidades *core*, com *scripts*. O segundo para alterar (*transform*) propriedades dos (objectos) eventos que chegam, e para converter as excepções de código geradas em outros novos eventos (opcional). O terceiro para executar acções, em massa ou não, nem sempre possíveis através da interface Web.

O Zenoss Core permite desenvolver e adicionar *plug-ins* Modeler para modelar componentes, e promove a criação de *plug-ins* ZenPack via Web ou com *YAML Ain't Markup Language* (YAML) tornando a configuração, importação de ficheiros MIB, e a definição de *monitoring templates* (e outros) única. Viabiliza a integração de *scripts* terceiros, como os *plug-ins* do Nagios, para verificar serviços e implementar agentes de monitorização activa. Admite a definição de gráficos com operações sobre métricas, *aliases* para conversões do tipo de dados, formatação dos dados de saída, e outros referidos pelos gestores de rede em questão. Utiliza a biblioteca D3 (D3.js, escrita em JavaScript) que combina *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e *Scalable Vector Graphics* (SVG) para apresentar gráficos compartilháveis com a qualidade da realidade moderna.

O Zenoss Core é, também, a plataforma de gestão com desenvolvimento mais activo. Conta com uma versão *major* e 8 versões *minor* entre Fevereiro e Novembro de 2015. Dispõe de um canal de *Internet Relay Chat* (IRC) e de um *bug tracking system* proprietário, onde se indicam problemas e se obtêm *patches* através de *commits* no repositório de código oficial, acessível pelo GitHub.

O Zenoss Core é, por isto, a plataforma de gestão indicada para a NOS Madeira.

4.5 Conclusão

Neste capítulo, foi apresentada a infra-estrutura de telecomunicações da NOS Madeira. Foi descrita a empresa e a organização desta em termos de utilizadores, grupos de utilizadores, dispositivos, grupos de dispositivos, serviços, *thresholds*, *triggers* e notificações importantes para a gestão efectiva dos recursos mencionados.

Foram enfatizados, sempre que aplicáveis, os problemas que a plataforma de gestão em uso comporta para a tarefa de gestão diária lá exigida. São vários os aspectos de organização que são constantemente transmitidos de boca em boca, por vezes registados em papel ou noutro suporte de dados persistente parecido que não o SNMPc.

O Zenoss Core é a plataforma de gestão *open-source* abordada mais adequada para resolver as limitações e os problemas descritos. No capítulo seguinte, são apresentados exemplos reais que tornam claro onde e como é que esta plataforma se destaca das restantes, e como é que ultrapassa as dificuldades vividas com a plataforma de gestão actual.

Capítulo V

Solução: Zenoss Core

Este capítulo indica onde e como é que o Zenoss Core se destaca das restantes plataformas de gestão *open-source* abordadas, para a gestão da infra-estrutura de telecomunicações da NOS Madeira. É com base nos problemas vividos com a plataforma de gestão em uso que se apresentam exemplos reais que contrastam o antes e o depois, o que não era possível fazer antes mas é possível fazer agora.

Os exemplos apresentados mostram como é que a nova plataforma de gestão suporta a informação fornecida pelos gestores de rede em questão. Mostra, para além disso, como é que se configuram serviços, *thresholds*, *triggers* e notificações imprescindíveis para amenizar o impacto de situações imprevistas.

5.1 Organização

Os dispositivos, os grupos de utilizadores, os serviços, os *thresholds*, os *triggers* e notificações da infra-estrutura de telecomunicações da NOS Madeira estão organizados, na nova plataforma de gestão, como descrito nas secções que se seguem.

5.1.1 Utilizadores

O Zenoss Core permite adicionar utilizadores e grupos de utilizadores como representa a Figura 5.1. Aqui, estão definidos os grupos de utilizadores da Tabela 4.1 e associados os utilizadores definidos a cargo de cada um deles.

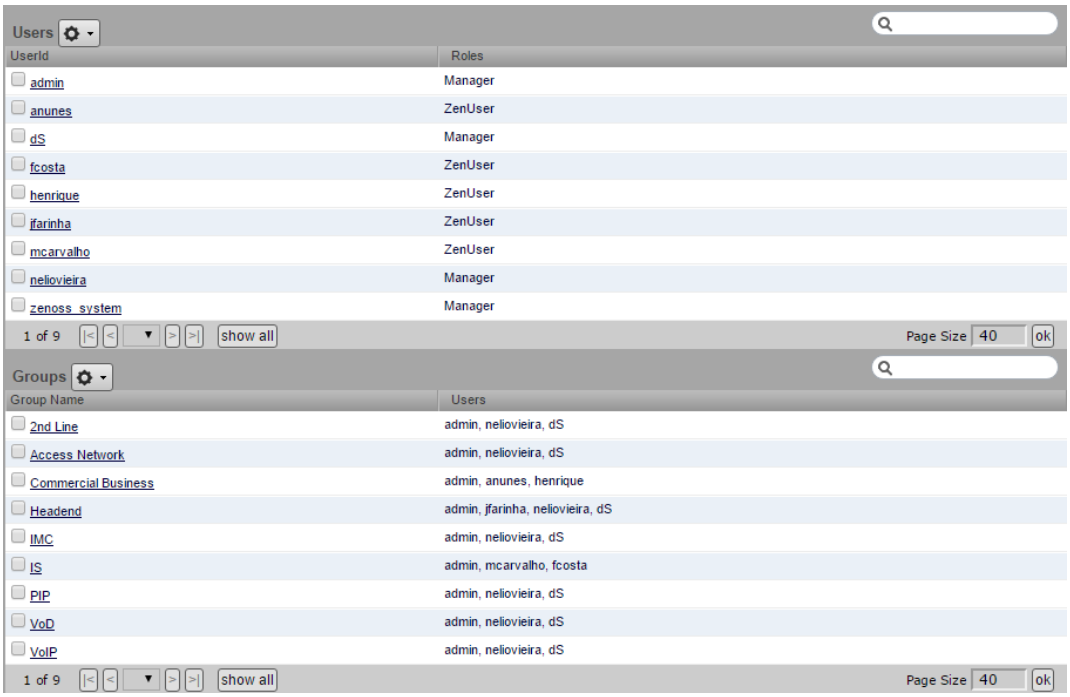


Figura 5.1: Utilizadores e grupos de utilizadores

O campo UserId reverte a favor da identificação unívoca de cada utilizador, e o campo Roles reverte a favor do papel (de utilizador) desempenhado por esse utilizador. Aos papéis de utilizadores são atribuídos controlo de acesso e permissões sobre qualquer secção e quaisquer objectos dentro da interface Web.

5.1.2 Dispositivos

Contrariamente ao SNMPc, o Zenoss Core permite organizar os dispositivos na secção Infrastructure através das árvores Devices, Groups, Systems e Locations como representa a Figura 5.2. Nestas árvores, estão criados tantos nós genéricos quanto nós específicos para agrupar os dispositivos de acordo com a Tabela 4.1.

Na árvore Devices, Environment e Mail são exemplos de nós genéricos criados. Em Mail, MX e WebMail são exemplos de nós específicos criados. Aqui, cada dispositivo pertence ao nó específico e aos nós genéricos que o supercedem (pelo que um dispositivo não pode estar em dois ou mais nós em simultâneo).

Na árvore Groups, PIP e SIs são exemplos de nós criados para agrupar logicamente os dispositivos conforme os grupos de utilizadores que os gerem. Aqui, e à semelhança do que acontece na realidade, um dispositivo pode pertencer a um

ou mais grupos de utilizadores (pelo que um dispositivo pode estar em dois ou mais nós em simultâneo).

Na árvore Systems, CMTS e SWDC são exemplos de nós criados para agrupar logicamente os dispositivos conforme os serviços que fornecem. Aqui, e à semelhança do que acontece na realidade, um dispositivo pode fornecer um ou mais serviços (pelo que pode um dispositivo pode estar em dois ou mais nós em simultâneo).

Na árvore Locations, FUN1 é um exemplo de nó criado para agrupar os dispositivos conforme a sua localização. Aqui, e à semelhança do que acontece na realidade, um dispositivo não pode pertencer a duas ou mais localizações (pelo que um dispositivo não pode estar em dois ou mais nós em simultâneo).

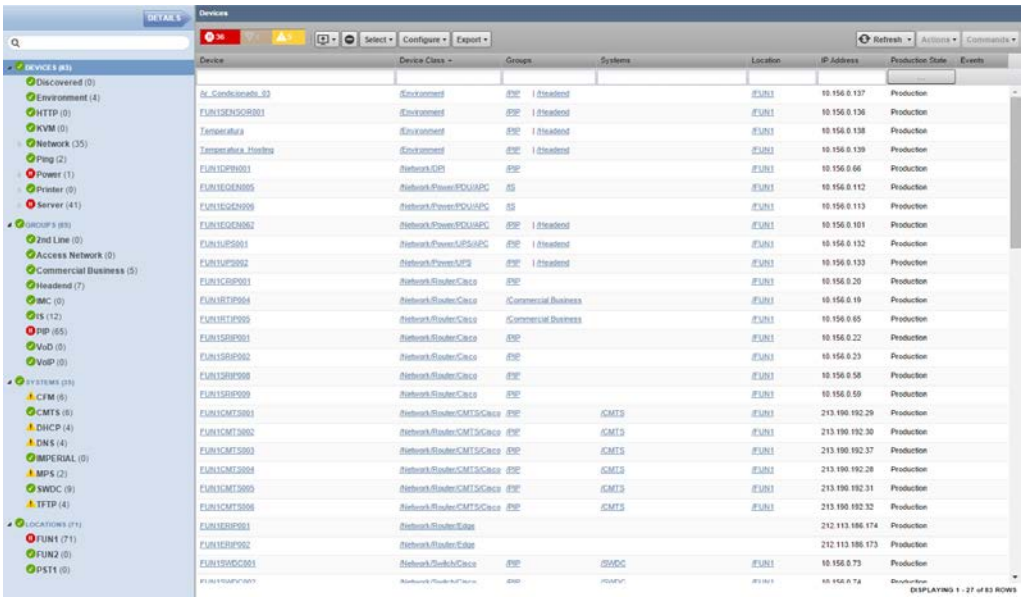


Figura 5.2: Dispositivos na secção Infrastructure do Zenoss

A secção Infrastructure da Figura 5.2 permite acrescentar e remover colunas, ajustá-las ao conteúdo automaticamente, redimensioná-las manualmente, reordená-las com *drag-and-drop*, ordenar e filtrar os dados por uma ou várias colunas em simultâneo. Permite exportar os dispositivos para *Comma-Separated Values* (CSV) ou XML, actualizar a janela automaticamente em intervalos de tempo configuráveis, executar acções e comandos em um ou mais dispositivos seleccionados em simultâneo. Possibilita, adicionalmente, sumarizar a quantidade de eventos de todos os dispositivos, adicionar um ou mais dispositivos em

simultâneo para monitorização, e descobrir redes/sub-redes automaticamente. Permite, entre outros, filtrar os dispositivos apresentados na tabela de dispositivos à medida que se clica em nós diferentes da mesma árvore, ou em nós diferentes de árvores diferentes.

5.1.3 Serviços

Contrariamente ao SNMPc, o Zenoss Core tira partido de *monitoring templates* para associar a verificação de serviços IP a dispositivos ou a grupos de dispositivos.

A Figura 5.3 mostra como é que está definida a verificação do serviço IP HTTPS com um *monitoring template*.

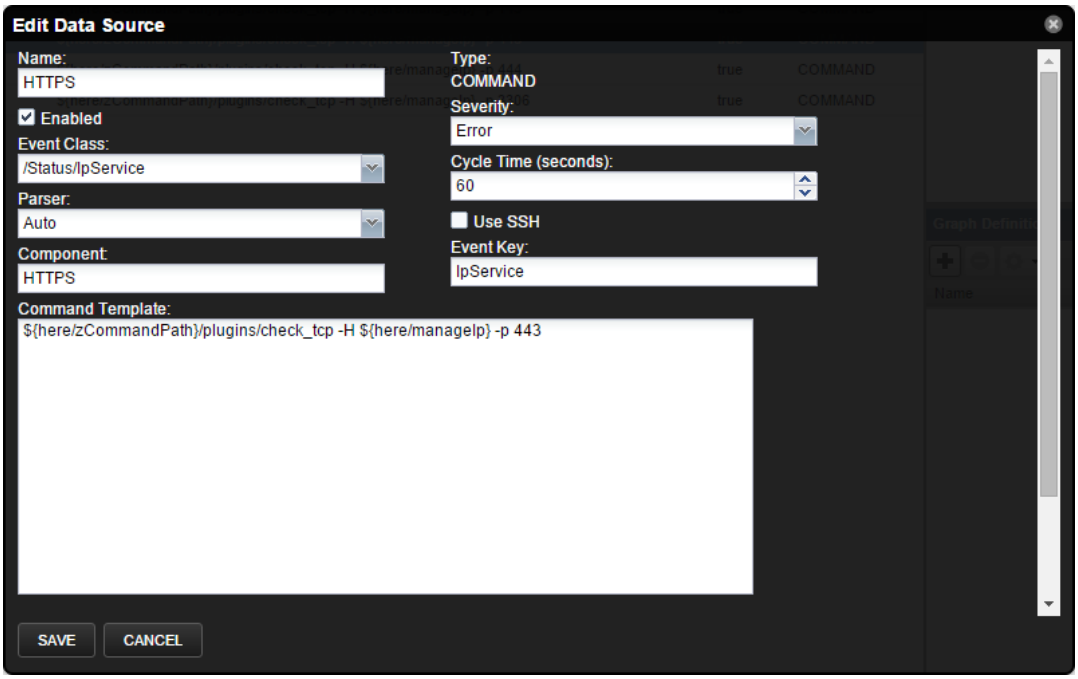


Figura 5.3: Serviço IP com *monitoring template*

Aqui, é configurada a verificação do serviço HTTPS através do *plug-in* check_tcp do Nagios. É possível activá-lo/desactivá-lo através da caixa de verificação Enabled, atribuir a classe de evento Event Class aos eventos gerados, indicar o Parser para interpretar o resultado obtido, e especificar o componente monitorizado. É também possível seleccionar a severidade dos eventos gerados, refinar o período de verificação, fazer uso de variáveis no comando a ser executado, e testá-lo num dispositivo real (não visível na Figura 5.3).

Os *monitoring templates* são facilmente associados a grupos de dispositivos como mostra Figura 5.4.

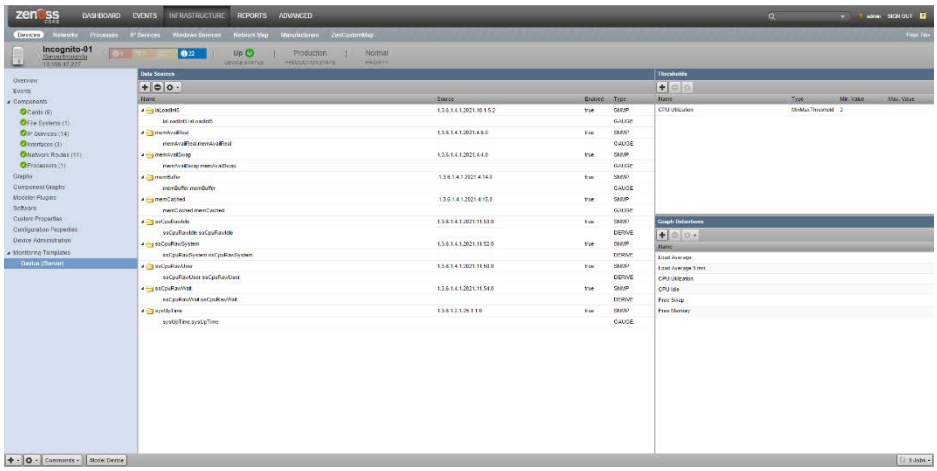


Figura 5.4: Dispositivo com *monitoring template*

Os *monitoring templates* permitem ao gestor definir que dados de monitorização o Zenoss Core deve recolher para cada dispositivo ou grupos de dispositivos, os *thresholds* e os gráficos respeitantes.

5.1.4 Thresholds

Contrariamente ao SNMPc, o Zenoss Core agrupa os *thresholds* por *monitoring template*. Com *monitoring templates* associados a dispositivos e/ou grupos de dispositivos, é inteiramente válido que o mesmo *threshold* assuma valores distintos para dispositivos e/ou grupos de dispositivos distintos.

A Figura 5.5 mostra como está definido o *threshold* para verificar o espaço disponível em disco.

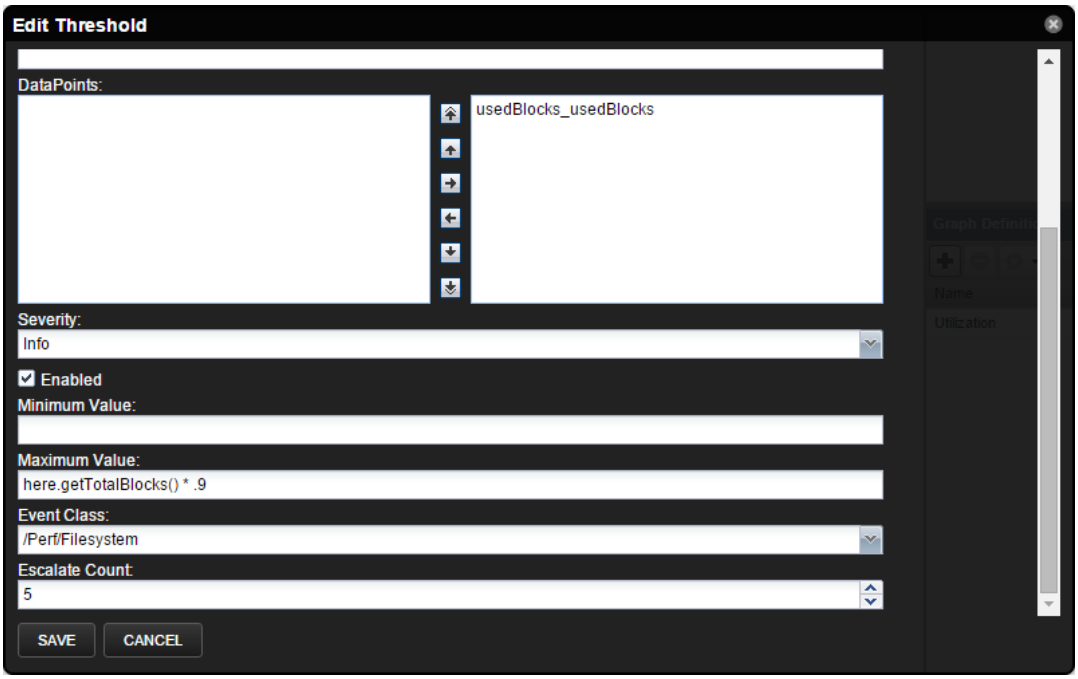


Figura 5.5: Edição de um *threshold*

Aqui, um *threshold* permite avaliar um ou mais dados de monitorização em simultâneo. É possível activá-lo/desactivá-lo através da caixa de verificação Enabled, atribuir a classe de evento Event Class aos eventos gerados, indicar o valor mínimo e o valor máximo para os resultados obtidos (no caso de um *threshold* MinMax). É também possível seleccionar a severidade dos eventos gerados, e escalar a severidade com base na propriedade Count dos eventos agrupados (idênticos, mas que ocorrem em períodos de tempo diferentes).

5.1.5 Triggers

Contrariamente ao SNMPc, o Zenoss Core permite associar um ou mais *triggers* a um evento só e/ou a vários eventos em simultâneo (classes ou tipos de evento diferentes). É com base nas propriedades do objecto evento que se retiram parâmetros de interesse distintos para utilizadores e/ou grupos de utilizadores distintos.

A Figura 5.6 mostra como está definido um dos *triggers* para os dispositivos do grupo de utilizadores PIP.

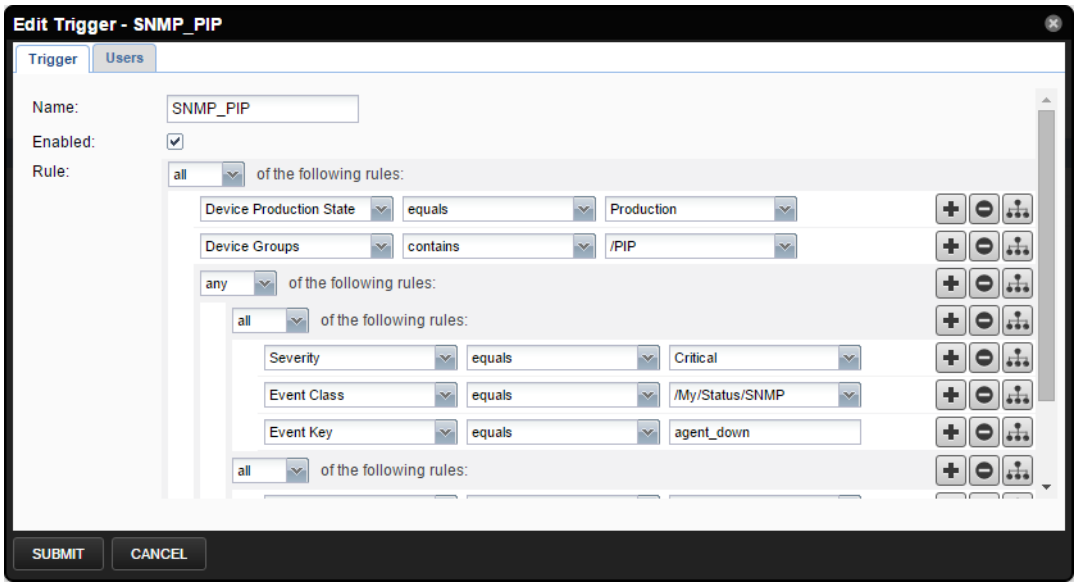


Figura 5.6: Edição de um *trigger*

Aqui, é possível activar/desactivar o *trigger* através da caixa de verificação Enabled. As regras Rule comportam condições lógicas que descrevem quando é que o *trigger* é, ou não, despoletado. O all representa a operação lógica E, e o any a operação lógica OU. É possível adicionar e remover tantas regras quanto necessário, e definir o utilizador ou grupo de utilizadores com permissões para ler ou ler e modificar o *trigger* em questão.

5.1.6 Notificações

Contrariamente ao SNMPc, o Zenoss Core permite associar uma ou mais notificações distintas sobre um ou mais *triggers* distintos. Isto faz com que a mesma notificação possa ser enviada por SMS para um utilizador ou grupo de utilizadores, e possa ser enviada por correio electrónico para outro utilizador ou grupo de utilizadores.

A Figura 5.7 mostra como está definida a notificação associada ao *trigger* da verificação dos serviços IP.

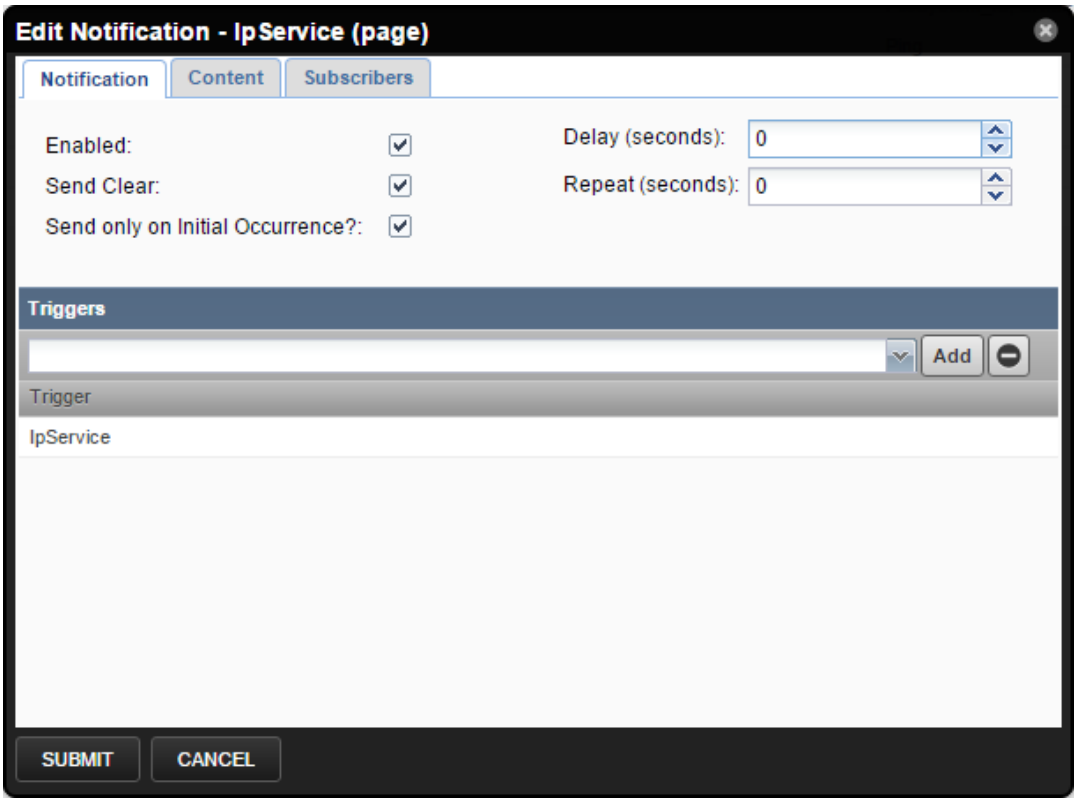


Figura 5.7: Edição de uma notificação

Aqui, page indica SMS. É possível activar/desactivar a SMS através da caixa de verificação Enabled, recebê-la uma vez só enquanto a situação estiver irregular através da caixa de verificação Send only on Initial Occurrence?, e receber uma outra SMS quando a situação estiver normalizada através da caixa de verificação Send Clear. É possível definir atraso no envio da SMS e o intervalo de repetição desta, em segundos. É também possível associar tantos *triggers* quanto possível para despoletar o envio da mesma. O conteúdo Content permite utilizar variáveis para formatar as mensagens tanto de situação irregular como de situação normalizada. Subscribers são, na verdade, os utilizadores ou grupos de utilizadores com permissões para ler ou ler e modificar a notificação, e os utilizadores ou grupos de utilizadores que vão receber a SMS em questão. É ainda possível agendar quando é que a SMS deve, ou não, ser enviada.

5.2 Conclusão

Neste capítulo, foi indicado onde e como é que o Zenoss Core se destaca das restantes plataformas de gestão *open-source* abordadas, para a gestão da infra-estrutura de telecomunicações da NOS Madeira. Foram apresentados exemplos reais que contrastam o antes e o depois respeitantes à definição e configuração de serviços, *thresholds*, *triggers* e notificações imprescindíveis para amenizar o impacto de situações imprevistas.

No capítulo seguinte, são abordadas as estratégias adoptadas para tornar o Zenoss Core o mais altamente disponível (possível). É feito um levantamento de requisitos, proposta uma arquitectura seguida do desenho e implementação de *software* respectivo, realizados testes e apresentados os resultados obtidos.

Capítulo VI

ZenDash

O projecto ZenDash surge para cobrir um ponto de ruptura na plataforma de gestão de redes Zenoss (Core): a alta disponibilidade do Control Center mestre. O Control Center mestre é o Control Center instalado no dispositivo onde o Zenoss reside. É o Control Center mestre quem controla e atribui as tarefas entre as *resource pools* e os dispositivos remotos que delas fazem parte. Cada um dos dispositivos remotos tem instalado, por sua vez, um Control Center agente para tornar a comunicação mestre-agente possível. É com base na comunicação mestre-agente que o Control Center mestre garante a distribuição de sobrecarga e a alta disponibilidade dos serviços envolvidos principalmente em situação de falha sendo, por isto, um ponto único de falha.

O presente capítulo descreve onde e como é que o ZenDash compensa as limitações da plataforma de gestão Zenoss nesta matéria. Descreve o desenho da solução concebida com um Control Center mestre designado mestre e um Control Center ou mais designados escravos. O mestre e os escravos, agentes redundantes Zenoss, automatizam tarefas como a verificação de serviços e a realização de cópias de segurança diárias e automáticas.

6.1 Requisitos Funcionais

As características, as funcionalidades e os serviços discutidos durante as reuniões com os gestores da infra-estrutura de telecomunicações respectiva sustentam o desenvolvimento de um *software* particular tal que, de acordo com os requisitos propostos, permita:

R1. Enviar uma cópia de cada *trap* recebida para os agentes redundantes
R2. Criar uma cópia de segurança diária e automática no mestre

Verificar:

R3. A disponibilidade dos agentes redundantes

R4. Os serviços do Zenoss nos agentes redundantes
R5. E redireccionar os gestores para a página Web do mestre

Receber e enviar:

R6. O correio electrónico do mestre apenas
R7. As SMS do mestre apenas

6.2 Arquitectura

O ZenDash é composto pelas camadas de *hardware*: Master/Slave, Gateway e UCARP (Figura 6.1).

A camada superior, designada Master/Slave, representa o *cluster* dos agentes redundantes Zenoss mestre e escravos do sistema.

A camada inferior, designada UCARP, representa o *cluster* dos dispositivos VIP que partilham um endereço IP virtual através do protocolo *Common Address Redundancy Protocol* (CARP).

A camada intermédia, designada Gateway, representa o *cluster* dos dispositivos *Simple Network Paging Protocol* (SNPP)[1] conjuntamente com o dispositivo SMTP[2] que estão a cargo de enviar as mensagens do agente redundante Zenoss mestre da camada superior Master/Slave.

[1]Cada dispositivo SNPP é, na verdade, um *gateway* de SMS.
[2]Servidor de correio electrónico.

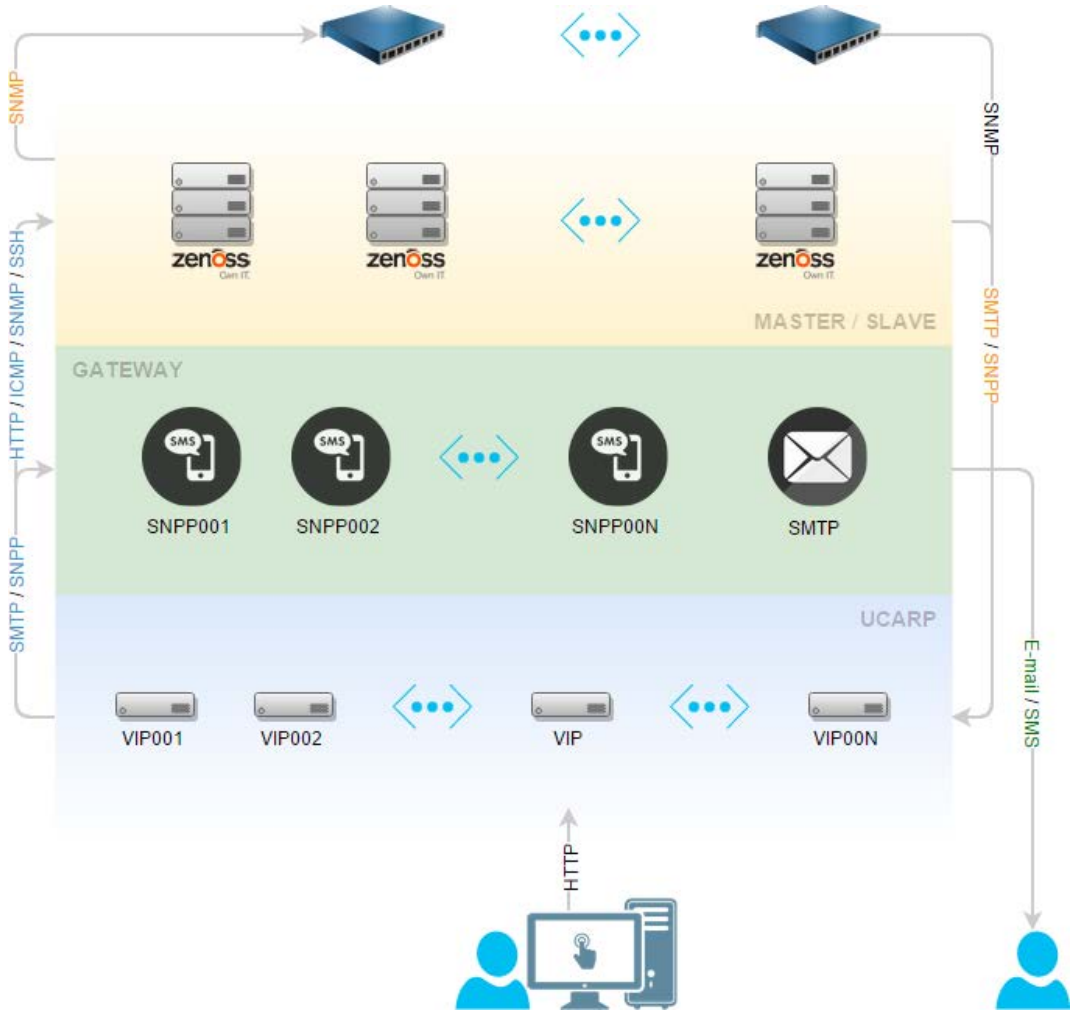


Figura 6.1: Arquitectura de *software* do ZenDash

É através do dispositivo VIP, ponto de entrada do sistema, que os gestores de rede da NOS Madeira acedem à interface Web de gestão do Zenoss sem conhecimento do agente redundante mestre sobre o qual operam.

O dispositivo VIP, mestre da camada inferior UCARP ao qual está atribuído o endereço IP virtual partilhado, confere a disponibilidade e os serviços dos agentes redundantes Zenoss da camada superior Master/Slave através da relação de confiança entre eles, sobre os quais determina qual o mestre e quais os escravos no momento. Recebe as SMS e o correio electrónico quer do mestre quer dos escravos mas reencaminha para a camada intermédia apenas as mensagens do mestre, que as faz chegar aos destinatários especificados através dos dispositivos SNPP e SMTP.

O dispositivo VIP envia, para o mestre e para os escravos da camada superior Master/Slave, uma cópia de cada *trap* recebida dos dispositivos monitorizados. Cria uma cópia de segurança diária e automática no mestre, a qual repõe em cada um dos escravos.

6.3 Desenho e Implementação

O projecto ZenDash consiste num conjunto de *daemons*, escritos em Python, configuráveis através de uma classe Conf criada para o propósito (à semelhança do que acontece com o Zenoss). Correm no dispositivo VIP, estão em ciclo infinito e são sincronizados pelo *daemon* servidor zen, que escuta e envia mensagens por *sockets* na porta local 9999 (também configurável) do protocolo TCP.

As mensagens são armazenadas e trocadas segundo a notação JSON, para que os *daemons* partilhem estruturas de dados primitivas entre si. Cada mensagem é precedida por uma outra mensagem, que indica o número de *bytes* a ser enviado e lido posteriormente. O número de *bytes* é representado em binário, ocupa 4 *bytes* (comprimento fixo) e segue a ordem de *bytes* little-endian.

É registada a actividade de cada *daemon* numa estrutura de directórios ano/mês/dia/*daemon*, através da classe Log criada para o propósito. O zenlog e o zentail, escritos em Bash com base no comando tail –f, acedem aos últimos ficheiros de registo gerados. O zenlog apresenta o último ciclo do *daemon* indicado, e o zentail as últimas 10 linhas deste (ou de todos os *daemons* em simultâneo). Aqui, as mensagens têm cores (configuráveis) para níveis de severidade diferentes.

É com o *daemon* servidor zen que os *daemons* cliente zenbackup, zenping, zenstatus, zenproxy, zensmtp, zensnpp e zentrap estabelecem ligação enquanto estão em execução, como mostra o desenho de *software* do ZenDash da Figura 6.2.

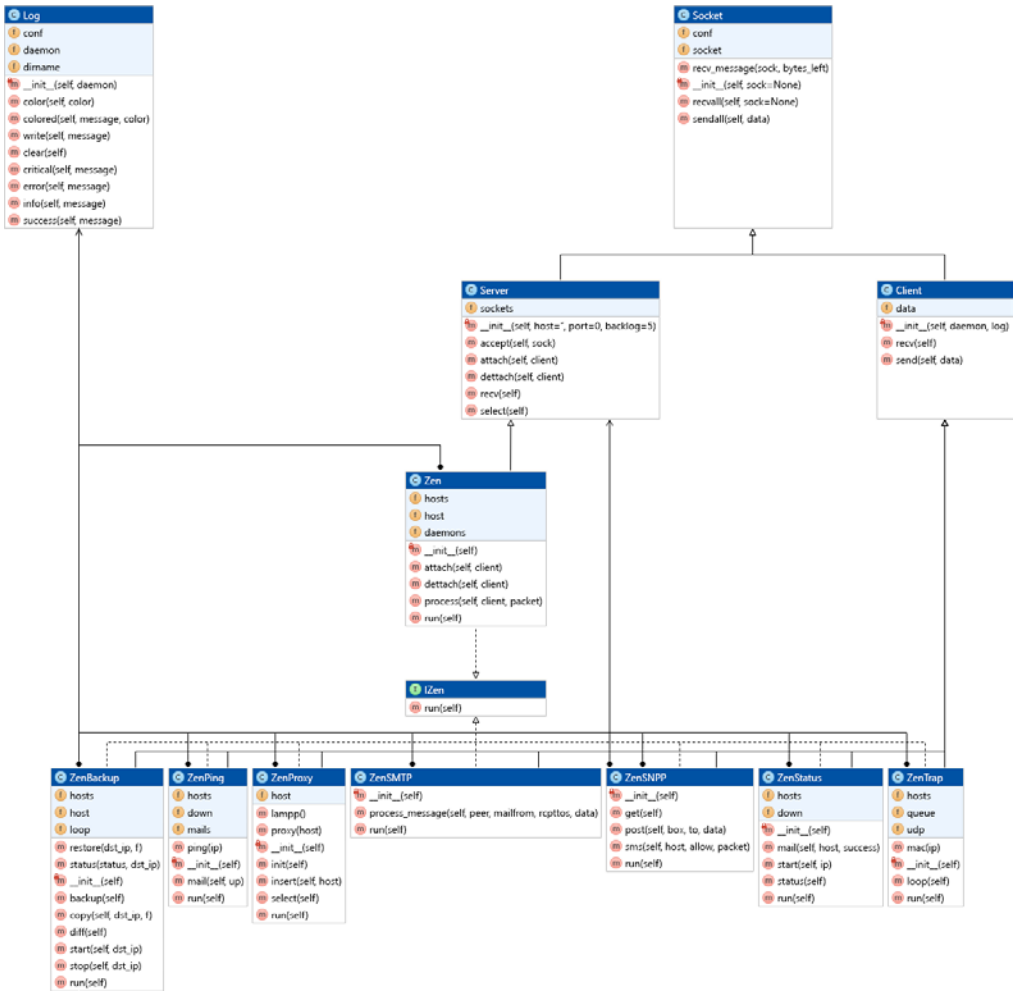


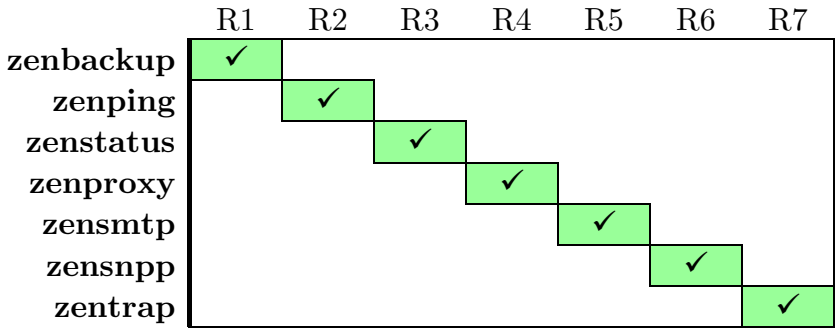
Figura 6.2: Desenho de *software* do ZenDash

O zen contém informações relevantes sobre os agentes redundantes disponíveis, configurados na classe de configuração Conf, e indica qual o mestre num determinado espaço de tempo.

Como servidor, comporta o estado de cada uma das ligações estabelecidas com os *daemons* cliente. Actualiza os dados locais envolvidos com base na informação recebida pelos clientes (ao longo do tempo), e notifica outros clientes que dela tiram proveito. Coordena o fluxo de comunicação do sistema, especialmente entre os *daemons* cliente zenping, zenstatus e zenproxy (da esquerda para a direita).

A Tabela 6.1 enfatiza a responsabilidade de cada um dos clientes desenvolvidos para o sistema, mediante o conjunto de requisitos funcionais estipulado na secção 6.1.

Tabela 6.1: Requisitos e funcionalidades por *daemons* cliente



O zenbackup é actualizado pelo zen sobre qual o mestre e quais os escravos no momento. Cria uma cópia de segurança no mestre, e repõe em cada um dos escravos. A distribuição entre o mestre e os escravos é directa, através do dispositivo VIP, com o comando *Secure CoPy* (SCP) -3. São parados e iniciados os serviços do Zenoss nos escravos automaticamente, antes e depois processo de reposição (ou restauro). Fá-lo em ciclo infinito, na hora (configurável) da cópia de segurança, para replicar as configurações do mestre para os escravos diária e automaticamente.

O zenping é actualizado pelo zen, no início, sobre quais os agentes redundantes configurados. Verifica a disponibilidade de cada um deles por ICMP, e devolve a informação actualizada ao zen. Envia uma mensagem de correio electrónico (para destinatários configuráveis) quando um ou mais agentes redundantes estão em baixo, e outra quando voltam à normalidade. Fá-lo em ciclo infinito, num intervalo de tempo constante (também configurável).

O zenstatus é actualizado pelo zen, quando actualizado pelo zenping, sobre os agentes redundantes ainda disponíveis. Verifica os serviços do Zenoss em cada um deles por SSH e devolve a informação ao zen, e tenta reiniciar serviços com problemas até ao número de tentativas com tempo de expiração constante (configuráveis). Envia uma mensagem de correio electrónico (para destinatários também configuráveis) quando um ou mais agentes redundantes persistem com problemas, e outra quando voltam à normalidade. Fá-lo em ciclo infinito, no intervalo de tempo coordenado pelo zenping.

O zenproxy é actualizado pelo zen, quando actualizado pelo zenstatus, sobre os agentes redundantes ainda disponíveis. Obtém, no início, qual o mestre registado na base de dados MySQL local. Caso não haja, ou esteja em baixo, é escolhido o agente mestre seguinte disponível (se algum), regista-o na base de dados local e devolve a informação ao zen. Redirecciona os gestores para a página Web do mestre através de um *proxy* reverso configurado no servidor Apache. Tenta reiniciar os serviços Apache e MySQL com problemas, também para garantir a replicação mestre-mestre configurada entre as bases de dados MySQL dos dispositivos VIP redundantes. Fá-lo em ciclo infinito, no intervalo de tempo coordenado pelo zenstatus.

O zensmtp escuta, bloqueia ou reencaminha o correio electrónico (SMTP) dos agentes redundantes. Apenas é reencaminhado, para o servidor SMTP que o faz chegar aos destinatários, o correio electrónico enviado pelo mestre. A validação é feita pelo zen, e evita o envio de mensagens duplicadas.

O zensnpp escuta, bloqueia ou reencaminha as SMS dos agentes redundantes via SNPP. Apenas são reencaminhadas, para um ou mais (configurável por dispositivo monitorizado) dispositivos SNPP disponíveis (no topo de preferências) que as fazem chegar aos destinatários, as SMS enviadas pelo mestre. A validação é feita pelo zen, e evita o envio de SMS duplicadas. Em caso de problema, a SMS é enviada para os endereços de correio electrónico dos destinatários respectivos.

O zentrap é actualizado pelo zen, no início, sobre quais os agentes redundantes configurados. Recebe as *traps* dos dispositivos monitorizados, retira os dados e o endereço IP de origem dos pacotes respectivos, cria pacotes com esses mesmos dados de interesse, e envia uma cópia dos pacotes criados para todos os agentes redundantes.

6.4 Teste e Depuração

Foram utilizados os dispositivos Zenoss-001, Zenoss-002, VIP001 e VIP002 para testes de sistema, monitorizados pelos dispositivos Zenoss-001 e Zenoss-002. Nos dispositivos VIP001 e VIP002, é verificado o serviço MySQL na porta TCP 3306.

O dispositivo VIP é, na verdade, o dispositivo VIP001 ou o dispositivo VIP002 com o endereço IP virtual atribuído no momento. Aqui, são verificados os serviços SMTP, HTTP, HTTPS e SNPP nas portas TCP 25, 80, 443 e 444 (respectivamente).

As notificações são, em qualquer um dos casos, enviadas para o grupo de utilizadores PIP.

São vários os resultados obtidos num período de testes de 5 meses. As situações irregulares detectadas foram causadas, na maior parte dos casos, por configurações indevidas e, portanto, corrigidas. Qualquer inconsistência de dados da dissertação para os resultados operacionais seguintes é justificada pela preferência de alguns termos em inglês ou outras situações de impacto semelhante quando configurada a solução final.

6.4.1 Eventos

Foram parados, iniciados e reiniciados serviços ao longo do tempo para auferir a fiabilidade da consola de eventos do Zenoss 5. Foram igualmente desligados, ligados e reiniciados dispositivos importantes para verificar o mesmo ou outro comportamento em contexto real. A sequência de eventos *down*/*up* e *up*/*down* produzida e o incremento do número de eventos vão de encontro com o esperado.

São já 1.981.751 os eventos arquivados e 82.433 os eventos por arquivar, dos quais apenas 115 estão visíveis e são verdadeiramente relevantes (Figura 6.3).

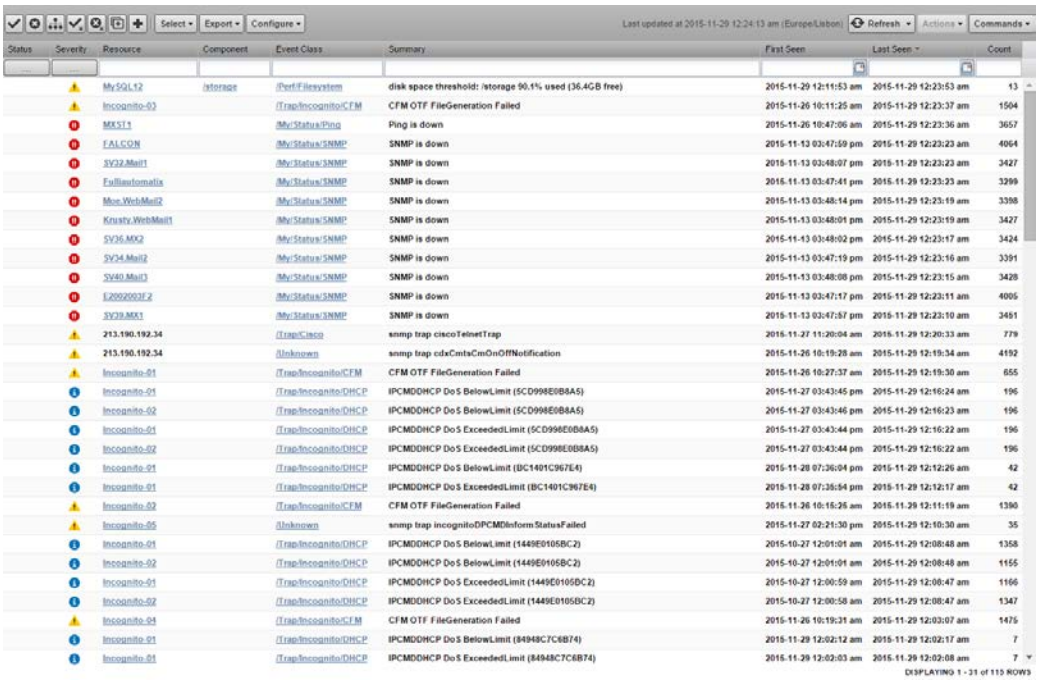


Figura 6.3: Consola de eventos do Zenoss

Na consola de eventos da Figura 6.3, é possível dar Acknowledgement e fechar (colocar como resolvido, e invisível) eventos. É possível reclassificar eventos (modificar a classe de evento), adicionar comentários aos eventos e criar eventos manuais configuráveis para situações de teste. É igualmente possível colorir as linhas da tabela de eventos de acordo com a severidade de cada um deles. É permitido, à semelhança da secção Infrastructure representada na Figura 5.2, acrescentar e remover colunas, ajustá-las ao conteúdo automaticamente, redimensioná-las manualmente, reordená-las com *drag-and-drop*, ordenar e filtrar os dados por uma ou várias colunas em simultâneo. É também possível guardar as acções tomadas, ou reverter tudo. É possível exportar os eventos para CSV ou XML, actualizar a janela automaticamente em intervalos de tempo configuráveis, executar acções e comandos no dispositivo ou dispositivos associados ao evento ou eventos seleccionados em simultâneo.

6.4.2 Thresholds, Triggers e Notificações

Para testes de alarmística foram utilizados *thresholds* definidos como representado na secção 5.1.4, *triggers* definidos como representado na secção 5.1.5 e notificações definidas como representado na secção 5.1.6.

A Figura 6.4 mostra um dos testes efectuados.

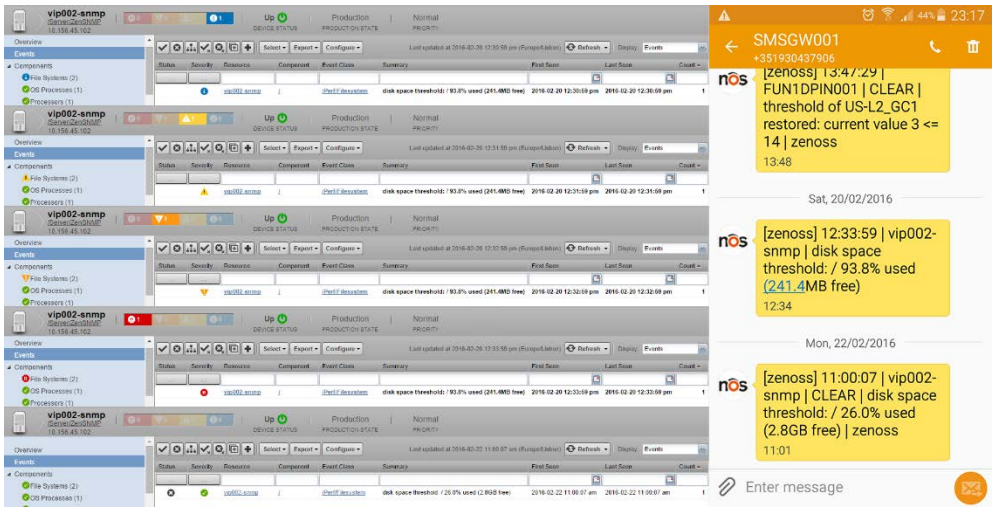


Figura 6.4: Notificações por SMS

Aqui, os resultados obtidos vão novamente de encontro com o esperado.

6.4.3 Mapas

Foram construídos mapas de rede (manualmente) à semelhança dos que estão definidos na plataforma de gestão de redes comercial em uso SNMPc. Aqui, foi explorada outra das características do Zenoss e verificado o comportamento resultante na simulação de situações problemáticas.

A simulação teve lugar num ecrã grande com os servidores Mail em baixo, seguido do envio de *traps* para verificar que alterações ocorriam no mapa de rede principal representado na Figura 6.5.

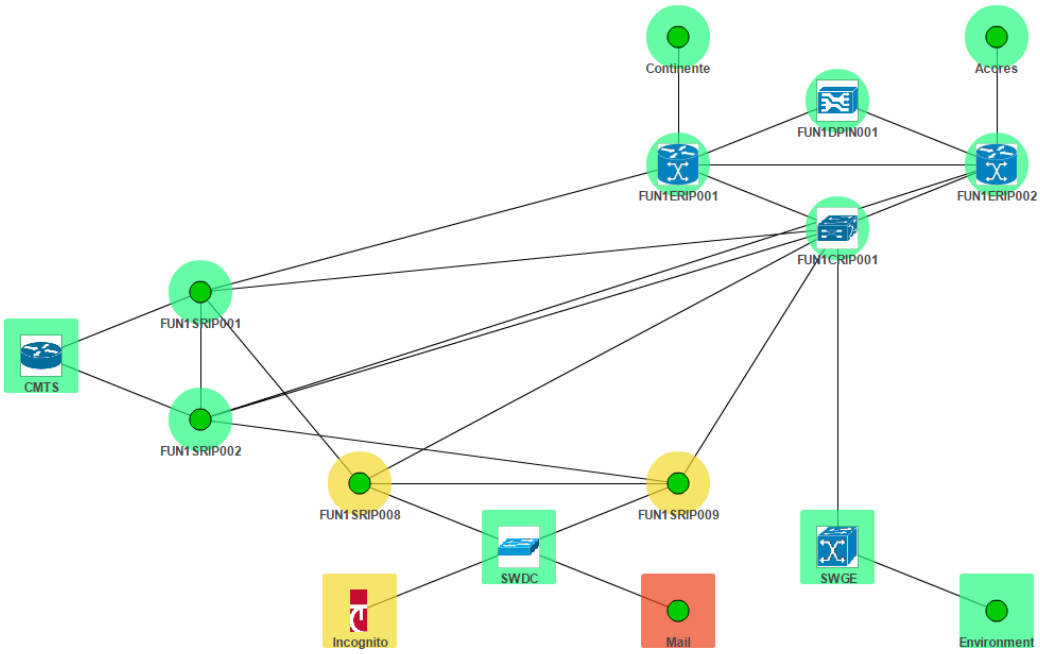


Figura 6.5: Mapa de rede principal no Zenoss

Na Figura 6.5, os círculos representam dispositivos e os quadrados representam sub-mapas. O amarelo indica a presença de eventos com severidade de alerta Warning, e o laranja (avermelhado) indica a presença de eventos com severidade crítica Critical. Um quadrado amarelo ou laranja (avermelhado) indica a presença de eventos com as severidades respectivas, em um ou mais dispositivos desse sub-mapa.

Aqui, os resultados obtidos vão de encontro com o esperado na medida em que o mapa revela o estado real dos dispositivos sempre que este se altera pelas várias razões possíveis.

6.4.4 Relatórios

Os relatórios[3] pré-definidos, os relatórios personalizados e os relatórios gerados automaticamente são outra das vantagens que o Zenoss tem para oferecer. É com base nos indicadores destes relatórios que se recolhem estatísticas sobre aspectos de um ou mais dispositivos agrupados conforme necessário, com ou sem filtro de dados como representa a Figura 6.6.

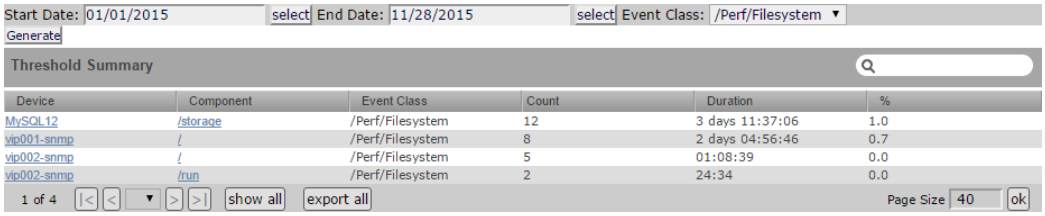


Figura 6.6: Relatório com *thresholds* sobre partições de disco

Aqui, nem sempre se consegue validar a informação gerada (mesmo que os resultados obtidos tenham sido positivos nos casos mais específicos).

6.4.5 Gráficos

É com base nos gráficos[4] pré-definidos e nos gráficos personalizados (filtrados e actualizados sob qualquer unidade de tempo automaticamente) que se testa outro potencial do Zenoss. A possibilidade de ver os gráficos em ecrã completo, e com a qualidade destes, colocou o Zenoss frente a frente com o Cacti.

São vários os gráficos por dispositivo e podem ser vários os gráficos incluídos nos relatórios da secção 6.4.4 de um ou mais dispositivos em simultâneo.

O gráfico da Figura 6.7 mostra a utilização de CPU num dispositivo específico.

[3]Podem ser exportados para CSV.
[4]Podem igualmente ser exportados para CSV.

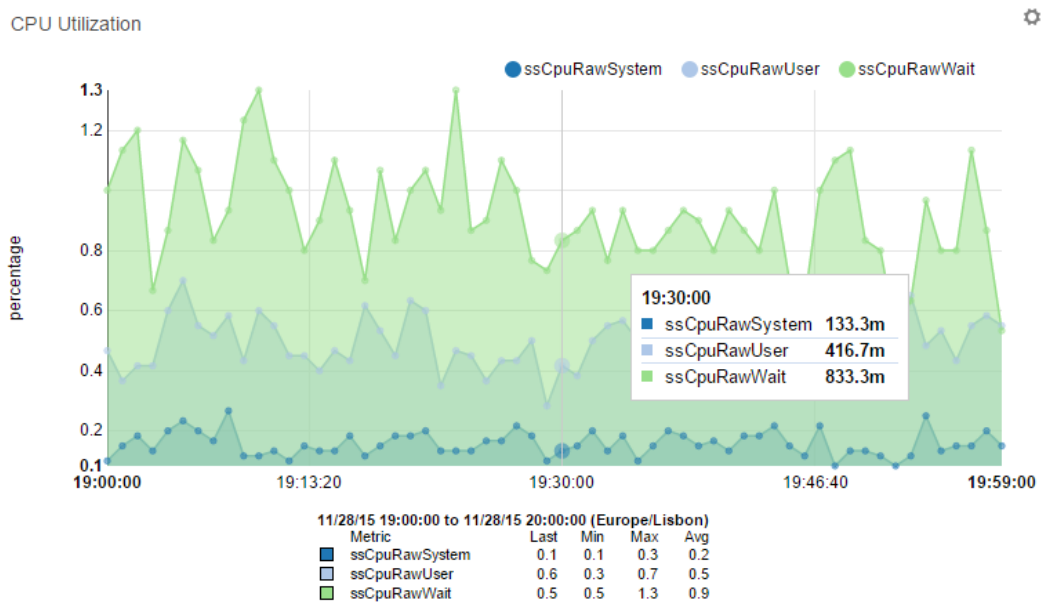


Figura 6.7: Gráfico com utilização de CPU

Os gráficos definidos permaneceram estáveis ao longo do tempo (ao contrário do que se pensou inicialmente). Não comportam, para já, algumas facilidades de configuração do Cacti incluídas em *plug-ins* comerciais (de propósito específico) igualmente disponíveis em *plug-ins* comunitários brevemente (para o Zenoss Core).

6.4.6 Outros

Os outros testes, sobre o ZenDash, permitem verificar o funcionamento básico do sistema de acordo com o que foi estipulado e implementado. Aqui, foi essencialmente testado o ficheiro cron.py e testados os *daemons* cliente zenbackup, zenping, zenproxy, zensmtp, zensnpp, zenstatus e zentrap.

A Figura 6.8 mostra alguns dos ficheiros de registo dinâmicos de *daemons* cliente em execução no dispositivo VIP.

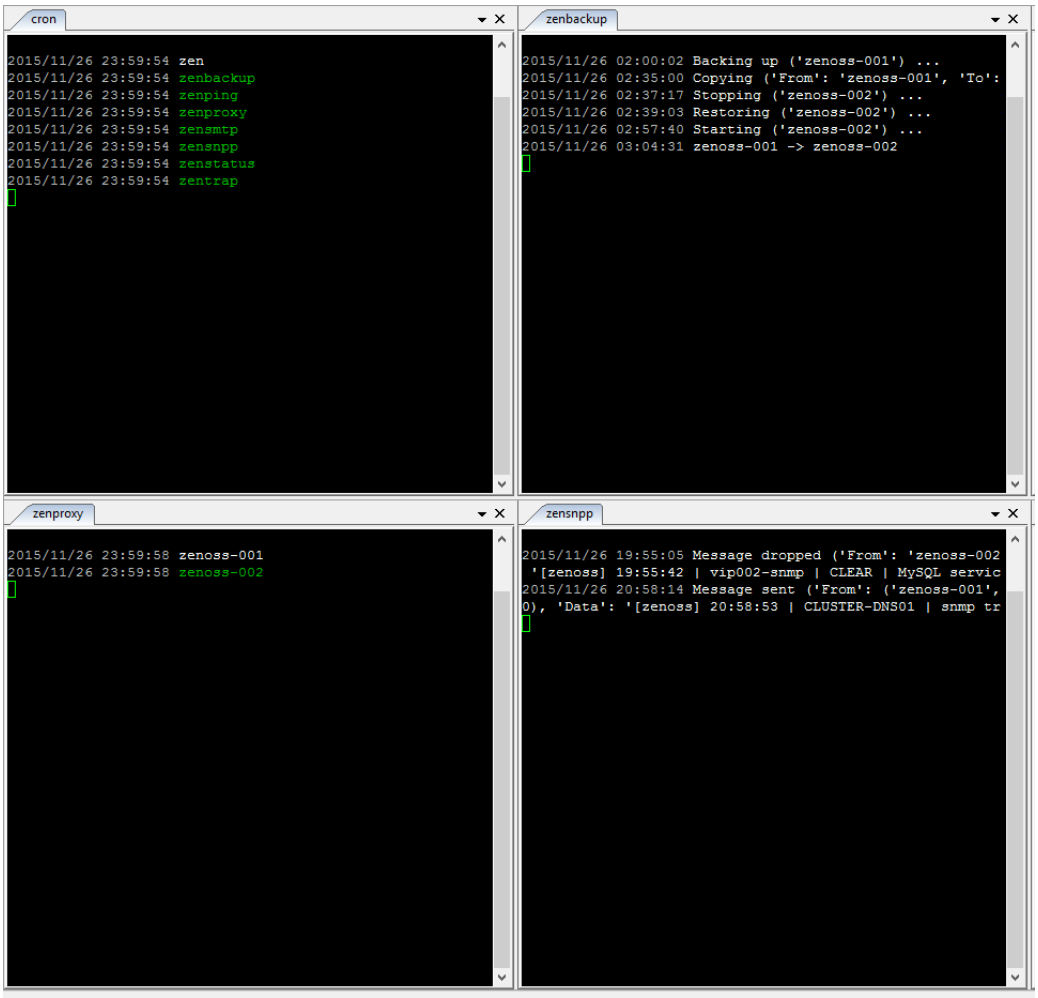


Figura 6.8: cron, zenproxy, zenbackup e zensnpp

No cron, o branco indica que o *daemon* servidor zen continua em execução desde o último ciclo de execução. O verde indica as ligações estabelecidas entre o zen e os *daemons* cliente, no momento.

No zenproxy, o branco indica qual o mestre no momento. O verde indica os escravos preparados para assumir o cargo de mestre nesse preciso momento.

No zenbackup, as mensagens a branco são meramente informativas. Estão a cobrir os passos realizados durante uma cópia de segurança efectuada com sucesso.

No zensnpp, as mensagens a branco são meramente informativas. Estão a cobrir duas situações distintas: a SMS do mestre (reencaminhada), e a SMS de um escravo (bloqueada/descartada para evitar o envio de SMS duplicadas).

Aqui, os resultados obtidos nem sempre foram de encontro com o esperado (pois as configurações nem sempre estiveram adequadas para a rede em questão). Já no fim, a situação acabou por se confirmar estável a ponto de entrar e estar em produção.

6.5 Distribuição e Manutenção

O ficheiro cron.py detecta o mestre e os escravos na camada inferior UCARP através da atribuição do endereço IP virtual partilhado entre eles. É executado e reexecutado automaticamente pelo Cron num intervalo de tempo configurável para garantir que uma instância deste processo verifique, inicie e pare os *daemons* pela ordem natural destes (especialmente na presença de problemas).

A distribuição do *software* fica assim reduzida aos *scripts* disponíveis no repositório GitHub, acessível em https://github.com/rubenhenriques/ZenDash. O responsável pelo processo apenas tem de saber que procedimento adoptar, segundo o tipo de dispositivo redundante a colocar em produção: se uma nova instância Zenoss, se um novo VIP.

Foram entregues guias de instalação, configuração e actualização do Zenoss e transmitida informação relevante sobre o ZenDash para a formação dos gestores de rede da NOS Madeira. É pretendida, com isto, a médio e longo prazos, a partilha de um nível de conhecimento comum entre toda a comitiva alheia às fases de planeamento, desenvolvimento e teste da solução implementada.

6.6 Conclusão

A realização do projecto ZenDash contribuiu fortemente para a revisão e aplicação de conceitos teórico-práticos de redes e sistemas operativos adquiridos ao longo dos anos. Exigiu o estudo e a compreensão das arquitecturas genéricas das plataformas de gestão, e seguiu um conjunto de directrizes respectivas fulcrais para o seu desenvolvimento.

Os requisitos e as funcionalidades destacados com a NOS Madeira fizeram com que desenhasse uma arquitectura multi-camada, com dispositivos redundantes em todas elas. Os requisitos fizeram com que desenvolvesse uma solução de *software* com entidades dedicadas para determinado tipo de funcionalidades, interligadas e sincronizadas entre si como indicado.

Aqui, foi configurado um *proxy* reverso no servidor Apache. Foram utilizados *threads* e *sockets*, e analisadores de pacotes desenvolvidos em Python. Foram criados *scripts*, em Bash, para auxiliar o teste e depuração do sistema pelo que a familiarização com sistemas operativos Debian e CentOS e a utilização de ferramentas tradicionais como TCPDUMP e IPTABLES foram uma mais-valia.

Capítulo VI

Conclusão

A realização desta dissertação contribuiu para estudar, compreender e aplicar conceitos substanciais da gestão de redes em contexto real. A complexidade da arquitectura de gestão OSI torna clara a necessidade e o aparecimento da arquitectura de gestão TCP/IP com sub-modelos simplificados que fazem dela norma, de facto, adoptada e desenvolvida pela maior parte dos fabricantes.

É com base na arquitectura TCP/IP que são desenvolvidas plataformas de gestão integradas, capazes de substituir ferramentas de gestão tradicionais limitadas e obsoletas no tempo. Estas plataformas são, cada vez mais, importantes para os grandes centros de operações de rede modernos por facilitarem e automatizarem a tarefa de gestão do modo que o fazem.

O estudo do NOC da NOS Madeira permitiu o contacto directo com as dificuldades que a tarefa de gestão acarreta na vida real. Aqui, a exigência é alta e faz-se sentir em cada ponto de estrangulamento de rede ou de serviços fornecidos. O cliente é a preocupação máxima, e a qualidade de serviço tem de estar no topo da lista de prioridades.

O levantamento de requisitos, e o cruzamento destes com as funcionalidades das plataformas de gestão *open-source* abordadas, permitiu uma visão geral sobre a oferta deste tipo de soluções mediante o que fazem a médio e o que podem fazer a longo prazo. O conceito Gestão de Redes Integrada e Automatizada parece estar, mais do que nunca, presente (e bem presente).

O desenho e a implementação de *software* adicional para tornar uma plataforma desta natureza o mais altamente disponível foi, sem dúvida, o elevar da fasquia nos desafios propostos. Exigiu o contacto com novas tecnologias, o cuidado redobrado e os testes de sistema para prevenir e detectar erros que comprometessem o correcto funcionamento da infra-estrutura considerada.

A satisfação da NOS Madeira com a plataforma de gestão seleccionada, e o sucesso da implementação e integração de alta disponibilidade com o projecto ZenDash foi e é motivo para pensar mais além.

7.1 Trabalho futuro

São várias as ideias e ainda mais as possibilidades de crescimento oferecidas pelo Zenoss. Aqui, são facilmente desenvolvidos *plug-ins* para auxiliar a modelação de dispositivos heterogéneos de marcas como a Cisco (e outras). A criação de *plug-ins* ZenPack, pela interface Web ou com YAML, é outro ponto de partida para a integração de componentes disponíveis apenas na versão comercial.

A integração transparente com qualquer *plug-in* do Nagios permite migrar de uma solução *agentless*, ou sem agentes, para uma solução com suporte para estes. São várias as plataformas e ainda mais os clientes desenvolvidos para consolidar o princípio de funcionamento do Zenoss ao nível da monitorização unificada Unified Monitoring.

O desenvolvimento do *daemon* zencommand para o ZenDash permitiria ao mestre executar comandos SSH remotamente, com um Zenoss pró-activo. A integração deste com outros *daemons* específicos é facilitada através do ponto central zen, que indica qual o mestre e quais os escravos no momento. O zen permite, na verdade, a comunicação entre qualquer *daemon* do sistema.

Referências

[1] V. Mohan, "5 Tips for Creating Your Own Network Operations Center (NOC)," 2014. [Online]. Available: http://web.swcdn.net/creative/pdf/whitepapers/5_tips_for_creating_your_own_network_operations_center.pdf. [Acedido em 2015].

[2] J. Oksanen, "Organizing a Network Operation Centre on Campus," 2013. [Online]. Available: http://services.geant.net/cbp/Knowledge_Base/Network_Monitoring/Documents/gn3-na3-t4-organizing-noc.pdf. [Acedido em 2015].

[3] Cisco Systems, Inc, "Internetworking Technology Handbook," 2003. [Online]. Available: http://docstore.mik.ua/cisco/pdf/routing/Cisco.Press.Internetworking.Technologies.Handbook.Fourth.Edition.eBook-kB.pdf. [Acedido em 2015].

[4] A. Balchunas, "OSI Reference Model v1.31," 2012. [Online]. Available: http://routeralley.com/guides/osi.pdf. [Acedido em 2015].

[5] Y. Yemini, "The OSI Network Management Model," 1993. [Online]. Available: http://researchgate.net/profile/Yechiam_Yemini/publication/3195162_The_OSI_network_management_model/links/00b495329c48caa3b8000000.pdf?disableCoverPage=true. [Acedido em 2015].

[6] R. White e D. Donohue, The Art of Network Architecture: Business-Driven Design, Cisco Press, 2014.

[7] R. Hassan, R. Razali, S. Mohseni, O. Mohamad e Z. Ismail, "Architecture of Network Management Tools for Heterogeneous System," 2009. [Online]. Available: http://arxiv.org/ftp/arxiv/papers/1001/1001.1967.pdf. [Acedido em 2015].

[8] Intel Corporation, "Network Management Systems Today," 2005. [Online]. Available:

http://marco.uminho.pt/~dias/MIECOM/GR/Projs/P2/fcaps-today-intel.pdf. [Acedido em 2015].

[9] Flextronics, "FCAPS," 2005. [Online]. Available: http://marco.uminho.pt/~dias/MIECOM/GR/Projs/P2/fcaps-wp.pdf. [Acedido em 2015].

[10] E. Monteiro e F. Boavida, Engenharia de Redes Informáticas, FCA, 2011.

[11] P. Marcu e W. Hommel, "Inter-organizational fault management: Functional and organizational core aspects of management architectures," 2011. [Online]. Available: http://arxiv.org/pdf/1101.3891.pdf. [Acedido em 2015].

[12] A. Gorod, R. Gove, D. B. Sauser e D. J. Boardman, "System of Systems Management: A Network Management Approach," 2007. [Online]. Available: http://worldsofsystems.com/downloads/2007GorodGoveSauserBoardman IEEE.pdf. [Acedido em 2015].

[13] L. A. Speaker, "A Toolkit for Developing TMN Manager/Agent Applications," 1996. [Online]. Available: http://hpl.hp.com/hpjournal/96oct/oct96a6.pdf. [Acedido em 2015].

[14] S. Al-Fedaghi, A. Alsaqa e Z. Fadel, "Conceptual Model for Communication," 2009. [Online]. Available: http://arxiv.org/ftp/arxiv/papers/0912/0912.0599.pdf. [Acedido em 2015].

[15] U. Warrier e L. Besaw, "The Common Management Information Services and Protocol over TCP/IP (CMOT)," 1989. [Online]. Available: https://tools.ietf.org/pdf/rfc1095.pdf. [Acedido em 2015].

[16] Contemporary Controls, "The ABCs of SNMP," 2006. [Online]. Available: http://ccontrols.com/pdf/abc11.pdf. [Acedido em 2015].

[17] H.-G. Hegering, S. Abeck e B. Neumair, Integrated Management of Networked Systems, Morgan Kaufmann, 1999.

[18] Microsoft, "Resources and Tools for IT Professionals | TechNet," 2003. [Online]. Available: https://i-technet.sec.s-msft.com/en-

us/library/cc783142.89f6ecac-cfe5-4e01-aab4-eac93a7c8e09(v=ws.10).gif. [Acedido em 2015].

[19] P. Murray, "SNMP: Simplified," 2008. [Online]. Available: https://f5.com/Portals/1/Cache/Pdfs/2421/snmp-simplified-.pdf. [Acedido em 2015].

[20] SolarWinds, "Introduction to SNMP Management," 2010. [Online]. Available: http://solarwinds.com/documentation/ref/IntroductionToSNMP.pdf. [Acedido em 2015].

[21] B. S. Kaliski Jr., "A Layman's Guide to a Subset of ASN.1, BER, and DER," 1993. [Online]. Available: http://zytrax.com/books/ldap/apb/asn1.pdf. [Acedido em 2015].

[22] E. J. Birrane e D. R. Cole, "Management of Disruption-Tolerant Networks: A Systems Engineering Approach," 2010. [Online]. Available: http://enu.kz/repository/2010/AIAA-2010-2206.pdf. [Acedido em 2015].

[23] J. Case, R. Mundy, D. Partain e B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework," 1999. [Online]. Available: http://getrfc.ru/files/pdf/rfc2570.txt.pdf. [Acedido em 2015].

[24] M.-J. Choi, Y.-J. Oh, H.-T. Ju e W.-K. Hong, "Towards XML-based Network Management for IP Networks," 2002. [Online]. Available: http://dpe.postech.ac.kr/knom/knom-review/v5n2/1.pdf. [Acedido em 2015].

[25] M. A. Miller, Internet Technology Handbook: Optimizing the IP Network, John Wiley & Sons, 2004.

[26] A. S. Godbole, Data Communications and Networks, Tata McGraw-Hill, 2002.

[27] E. Wong, "Network Monitoring Fundamentals and Standards," 1997. [Online]. Available: http://cse.wustl.edu/~jain/cis788-97/ftp/net_monitoring.pdf. [Acedido em 2015].

[28] N. J. Muller, "RMON, the New SNMP Remote Monitoring Standard," 1999. [Online]. Available: http://ittoday.info/AIMS/DCM/52-20-15.PDF. [Acedido em 2015].

[29] J. Ding, Advances in Network Management, CRC Press, 2009.

[30] M. H. Sherif, Handbook of Enterprise Integration, CRC Press, 2009.

[31] A. Arora et al., "Web Services for Management," 2005. [Online]. Available: http://specs.xmlsoap.org/ws/2005/06/management/ws-management.pdf. [Acedido em 2015].

[32] A. Sahai e C. Morin, "The Mobile Agent Enhanced Thin Client Approach to Network Management," 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.23.8793&rep=rep1&type=pdf. [Acedido em 2015].

[33] R. Khan, S. U. Khan, R. Zaheer e M. I. Babar, "An Efficient Network Monitoring and Management System," 2013. [Online]. Available: http://ijiee.org/papers/280-N011.pdf. [Acedido em 2015].

[34] V. Mohan, "A Guide to Enterprise Network Monitoring," 2013. [Online]. Available: http://web.swcdn.net/creative/pdf/Whitepapers/a_guide_to_enterprise_network_monitoring.pdf. [Acedido em 2015].

[35] H. Schwichtenberg e U. Trottenberg, "Interoperability and Openness in Today's Heterogeneous IT Environments," 2011. [Online]. Available: http://scai.fraunhofer.de/fileadmin/download/publikationen/Study-Interoperability-Openness-FraunhoferSCAI.pdf. [Acedido em 2015].

[36] R. Jenkins, "Why Web-based Network Monitoring? Leveraging the Platform," 1999. [Online]. Available: http://dpnm.postech.ac.kr/research/02/xgems/refpaper/WBNMpaper/Whyweb-basednetworkmonitoringLeveragingtheplatform.pdf. [Acedido em 2015].

[37] V. Mehta, Icinga Network Monitoring, Packt Publishing, 2013.

[38] SAS EDEATION, "edeation, l'expert en solutions open source," 2015. [Online]. Available: http://edeation.fr/wp-content/uploads/2015/03/Icinga_Architecture.png. [Acedido em 2015].

[39] Icinga, "Icinga | Open Source Monitoring," 2009. [Online]. Available: https://icinga.org. [Acedido em 2015].

[40] T. Ryder, Nagios Core Administration Cookbook, Packt Publishing, 2013.

[41] W. Kocjan, Learning Nagios 4, Packt Publishing, 2014.

[42] DigLinks GmbH, "Addon - Nagios to RRD - N2RRD," 2006. [Online]. Available: http://n2rrd-wiki.diglinks.com/download/attachments/557060/nagios-n2rrd.png. [Acedido em 2015].

[43] Nagios Enterprises, LLC, "NRDP - Overview," 2012. [Online]. Available: https://assets.nagios.com/downloads/nrdp/docs/NRDP_Overview.pdf. [Acedido em 2015].

[44] E. Galstad, "NRPE Documentation," 2007. [Online]. Available: https://assets.nagios.com/downloads/nagioscore/docs/nrpe/NRPE.pdf. [Acedido em 2015].

[45] Nagios Enterprises, LLC, "Nagios XI - Configuring The Windows Agent: NSClient++," 2014. [Online]. Available: https://assets.nagios.com/downloads/nagiosxi/docs/Configuring-The-Windows-Agent-NSClient++-for-Nagios-XI.pdf. [Acedido em 2015].

[46] Nagios Enterprises, LLC, "Nagios XI - Installing The Windows Agent: NSClient++," 2014. [Online]. Available: https://assets.nagios.com/downloads/nagiosxi/docs/Installing-The-Windows-Agent-NSClient++-for-Nagios-XI.pdf. [Acedido em 2015].

[47] Zabbix LLC, "Zabbix 1.6 Manual," 2008. [Online]. Available: http://zabbix.com/downloads/ZABBIX Manual v1.6.pdf. [Acedido em 2015].

[48] Zabbix LLC, "Zabbix.org," 2001. [Online]. Available: http://zabbix.org/mw/images/4/4b/ZabbixArchitecture-Generalized.png. [Acedido em 2015].

[49] R. Olups, Zabbix 1.8 Network Monitoring, Packt Publishing, 2010.

[50] Zabbix LLC, "Enterprise-Class Monitoring Solution for Everyone: All-in-One Open-Source Distributed Monitoring," 2014. [Online]. Available:

http://zabbix.com/files/Brochures/Zabbix_product_brochure_web.pdf. [Acedido em 2015].

[51] J. Curry, "Event Management for Zenoss Core 4," 2013. [Online]. Available: http://www.skills-1st.co.uk/papers/jane/zenoss4-events/zenoss_Core4_event_management_paper.pdf. [Acedido em 2015].

[52] Zenoss, Inc, "Zenoss Service Dynamics 5.0: Architecture Overview," 2015. [Online]. Available: http://zenoss.com/documents/wp_zsd_architecture_overview.pdf. [Acedido em 2015].

[53] Zenoss, Inc, "Redefining Monitoring for Today's Modern IT Infrastructures," 2015. [Online]. Available: http://zenoss.com/documents/Redefining-Monitoring-5.pdf. [Acedido em 2015].

[54] R. Cattell, "Scalable SQL and NoSQL Data Stores," 2011. [Online]. Available: http://cattell.net/datastores/Datastores.pdf. [Acedido em 2015].

[55] Wikipedia, "Comparison of network monitoring systems - Wikipedia, the free encyclopedia," 2001. [Online]. Available: https://wikipedia.org/wiki/Comparison_of_network_monitoring_systems. [Acedido em 2015].

[56] Castle Rock Computing, "Network Management - Castle Rock Computing - SNMPc," 1987. [Online]. Available: https://www.castlerock.com/company/. [Acedido em 2016].