Management der Konfigurationen und Zustände von Geräten in föderierten, verteilten Testbeds

Diplomarbeit

am Fachgebiet

Agententechnologien in betrieblichen Anwendungen und der Telekommunikation (AOT)

Prof. Dr.-Ing. habil. Sahin Albayrak

Fakultät IV – Elektrotechnik und Informatik

Technische Universität Berlin

Von Heiko Blume

Matrikelnummer 92507

Betreuer: Frank Steuer, Thomas Kaschwig

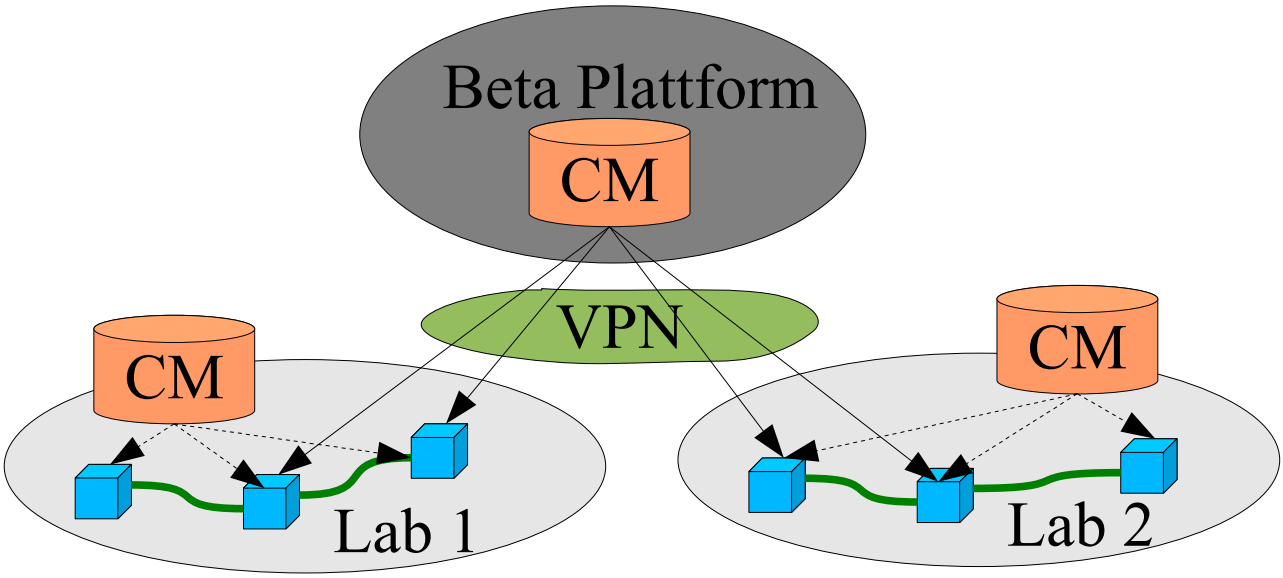
Thema zur Diplomarbeit „Management der Konfigurationen und Zustände von Geräten in föderierten, verteilten Testbeds“

i) Einleitung

Das Ziel der Diplomarbeit ist die Integrierung und Erweiterung von Open Source Programmen zu einer Lösung zur Versionskontrolle von Gerätekonfigurationen mit Schwerpunkt auf Routern und Switchen. Es sollen Konfigurationsdaten und Betriebszustände automatisiert und/oder manuell abgerufen, in einem Repository zur Versionskontrolle archiviert und mittels einer grafischen Weboberfläche visualisiert und dokumentiert werden. Durch ein Framework zum Übertragen von Konfigurationen aus einem Repository auf aktive Geräte sollen im Idealfall ganze Labore durch einen einzigen Aufruf komplett umgestellt werden können. Zusätzlich sollen mehrere Repositories zur Bearbeitung verschiedener Labore auf der selben Installation möglich sein. Da die Konfiguration oft vom Vorhandensein einer bestimmten Software auf den Geräten abhängt, soll außerdem ein Framework für eine einfache Backup Funktion erstellt werden. Die genannten Funktionen sollen für Cisco IOS und Juniper JunOS implementiert werden, für weitere sollen Templates erstellt werden. Das GUI soll für die Verwaltung der Lösung entsprechend erweitert werden.

ii) Szenario

Für das Projekt „Beta Plattform“ ist geplant, Geräte aus Laboren verschiedener Betreiber durch eine standortübergreifende Vernetzung für sekundäre Nutzung zugänglich zu machen. Über eine VPN Funktion sollen Dritte ausgewählte Geräte verbinden können, um damit zu arbeiten, ohne eigene Labore aufzubauen oder erweitern zu müssen.



Die Geräte sollen dazu online lokalisiert und gebucht werden können. Um diese Nutzung für die Laborbetreiber und die zukünftigen externen Nutzer nachhaltig zu ermöglichen, muss ein Management der Konfigurationen („Configuration Management – CM“) möglich sein, dass

kontinuierlich die Zustände der logischen Labore erfasst und einen gewünschten früheren Zustand leicht wiederherstellen kann.

Um den operativen Aspekt einfach zu gestalten und den Laborbetreibern einen direkten Nutzen der Teilnahme an der Beta Plattform zu bieten, soll innerhalb jedes physischen Labors eine lokale Installation des CM existieren können, die der Betreiber für seine eigenen, internen Zwecke einsetzen kann.

Die externen Nutzer benutzen dagegen eine übergeordnete Installation des CM bei der Beta Plattform. Diese arbeitet Betreiber übergreifend und greift über das VPN auf die Geräte in den Laboren zu um Zustände auszulesen und Konfigurationen einzuspielen.

iii) Anforderungen

Die Arbeit soll Open Source Programme integrieren und erweitern um sowohl die Dokumentationsfunktion als auch den Rollout von Konfigurationen im oben beschriebenen Szenario zu implementieren. Dazu werden verschiedene Programme in einer virtuellen Maschine für VMware x86 genutzt.

Die folgenden Funktionen sollen realisiert werden:

- Archivierung von Konfigurationen in mehreren Repositories
- Framework für die Beschickung von Geräten mit Konfigurationen aus den Repositories (Rollout), herstellerunabhängig, skriptbasiert, erweiterbar.
- Framework für eine einfache Sicherungsfunktion der Software, Herstellerunabhängig, skriptbasiert, erweiterbar
- Erweiterung des GUI zur Administration von Repositories und Gerätegruppen und zum Anstoßen der Funktionen für Rollout, Sicherung usw.

Die folgenden Voraussetzungen für den Wirkbetrieb müssen gegeben sein:

- Auf dem Zielrechner muss VMware installiert sein
- Die Informationen über die Geräte werden durch ein anderes Teilprojekt bereitgestellt

iv) Literatur

- Subversion (SVN) ist ein Versionskontrollsystem, http://subversion.tigris.org/
- RANCID ruft Zustände von Netzwerkgeräten ab, http://www.shrubbery.net/rancid/
- Insurrection ist ein SVN GUI, http://insurrection.tigris.org/

Eidesstattliche Versicherung

Die selbständige und eigenhändige Ausfertigung versichert an Eides statt:

_____

Heiko Blume Berlin, den 4.4.2009

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Zusammenfassung

Diese Arbeit implementiert eine virtuelle Linux Appliance zur zentralen Verwaltung der Konfigurationen der Software und Hardware sowie ausgewählter Zustände von Netzwerkkomponenten und UNIX Servern in verteilten, föderierten Testbeds und Laboren im Rahmen der Beta-Plattform, einem Projekt des Nationalen IT-Gipfels der Bundesregierung. Die Beta-Plattform ermöglicht die Nutzung von Geräten aus verschiedenen Testbeds in virtuellen Testbeds, die durch individuelle virtuell-private Netzwerke im Internet gebildet werden. Diese Implementierung ist aber generisch und kann auch zum lokalen, unabhängigen Einsatz verwendet werden, sodass jeder Testbedbetreiber auch direkt und beim Einsatz von Geräten profitieren kann, die nicht der Beta-Plattform zur Verfügung gestellt werden.

Das Configuration Management ermöglicht die komfortable Überwachung, Dokumentation, Archivierung und Revisionskontrolle der Konfigurationen und bestimmter Zustände. Darüber hinaus können die Software oder Dateien der Geräte archiviert werden, die Geräte können durch Fernsteuerung eingeschaltet, ausgeschaltet und neu gestartet werden. Für manche Gerätetypen können archivierte Konfigurationen auf die Geräte zurückgespielt werden (Deploy). Die Implementierung ist voll mandantenfähig.

Es wurden dazu die Open Source Programme Apache, Insurrection, Subversion, RANCID und selbst programmierte Erweiterungen zum Abruf von Software Images, zum Strommanagement und zur Integration aller Teile in die grafische Oberfläche zu einer gebrauchsfertigen Plattform vereinigt.

Die verwendeten Lösungen sind herstellerneutral. Sie können leicht auf weitere Geräte angepasst oder um neue Funktionen erweitert werden. In dieser Arbeit wurde das mit dem Abruf von Software Images, der Deploy Funktion und dem Strommanagement praktiziert. Dies ist bei kommerziellen Lösungen oft nicht ohne weiteres gegeben. Aus praktischen Gründen lag der Schwerpunkt der Entwicklung auf Geräten von Cisco Systems, Juniper Networks und auf Linux Systemen.

Zusätzlich kann ein privates Testbed durch ein virtuelles Router Netz auf Basis der Cisco Router Emulation DynaMIPS gebildet werden.

2 Einführung und Problemanalyse

In umfangreichen Test- und Entwicklungsumgebungen ist das Konfigurationsmanagement eine ständige Herausforderung. Gleichzeitiges, unabhängiges Arbeiten an einer Vielzahl von Geräten, deren Software und Konfigurationen, wird schnell intransparent. Durch den Einsatz des Fernzugriffs über Netzwerke finden Änderungen oft unbemerkt statt. Im Nachhinein ist es nur schwer zu ermitteln, wie es zu Änderungen gekommen ist. Erschwerend kommt häufig hinzu, dass der Zugriff auf die Geräte mit anonymen „Rollenzugängen" statt mit personenbezogenen Anmeldungen erfolgt.

Zu den klassischen Servern, Routern und Switches in Wirtschaft und Forschung sind in den letzten Jahren neue Geräteklassen hinzugekommen.

Durch preiswerte Hardware haben sich die integrierten Geräte sprunghaft vermehrt. Besonders zahlreich sind inzwischen Wireless LAN Access Points, DSL Router und mobile Geräte. Die Funktionalität und Komplexität steigt dabei durch den preiswerten Speicher, insb. Flash Speicher für umfangreiche Software, ständig weiter an. Zum Beispiel kann ein WLAN Access Point zusätzlich ein Vielzahl von hochwertigen Funktionen wie File Server und Voice-Over-IP Nebenstellenanlage erfüllen.

Gleichzeitig übernehmen zunehmend preiswerte Server mit Open Source UNIX Derivaten immer mehr Funktionen, die bisher dedizierte Netzwerkgeräte geleistet haben:

- Router, z.B. mit Quagga[1]
- WLAN Access Points und Meshing
- Firewalls
- Hochwertige VoIP Funktionen mit GNU Gatekeeper[2] und OpenSIPS[3]

Durch diese Entwicklungen wird die IT immer mächtiger und komplexer. Die Abhängigkeit von der IT und die Gefahr den Überblick zu verlieren wird somit beständig größer.

2.1 Konfigurationen und Zustände sind Werte

In der Arbeitswelt stellen die komplexen IT-Strukturen nicht nur Arbeitsmittel dar. Sie repräsentieren auch das Wissen der Mitarbeiter und erfordern erhebliche Investitionen für den Aufbau und den Betrieb.

Dabei sind eine Reihe von Faktoren besonders hervorzuheben.

Die Konfigurationen alleine sind schon zunehmend abstrakt, komplex, umfangreich und verteilt. Die Zustände der Hardware (Modultypen, Firmware, Hardware Revision, Seriennummern) sind in der eigentlichen Konfiguration jedoch nicht enthalten. Diese Informationen sind aber für viele Zusammenhänge relevant, z.B. Kompatibilität, Lizenzbedingungen und Wartungsverträge.

Der eigentliche Sinn der Konfigurationen ist oft das Erzeugen von "soft states", nicht-permanenten und dynamischen Zuständen. In Netzwerken werden zum Beispiel durch Routingprotokolle auf weit verteilten Geräten kooperative Systeme gebildet.

Durch Änderungen an der Hardware oder Umgebung, die sich in der Konfiguration nicht widerspiegeln, können betriebsrelevante Änderungen (z.B. Ausfälle) auftreten, die ohne Kenntnis des vorherigen Gesamtzustandes schwer zu analysieren sind.

Für Proofs of Concept, Reproduktion von Fehler-Situationen für Hersteller, Nachweis der Funktionstüchtigkeit für Auftraggeber und ähnliche Aufgaben kann die Dokumentation des Gesamtzustandes sehr aufwendig werden.

[1] http://www.quagga.net/
[2] http://www.gnugk.org/
[3] http://www.opensips.org/

Viele Unternehmen und deren Werte bestehen heute fast nur noch aus IT und Know-How. Will man solche Unternehmen bewerten, spielen also die Daten in den Köpfen und Geräten eine wesentliche Rolle.

Daher ist es eine kritische Aufgabe, für den langfristigen, geordneten Erhalt dieser Daten zu sorgen. Dazu wird auch ein umfassendes, erweiterbares Configuration Management benötigt.

2.2 Dokumentation

Eine permanente Herausforderung ist das Erstellen und Pflegen von aussagekräftiger und praktischer Dokumentation für die IT. Oft fehlt es an der Zeit der Mitarbeiter und der Entschlossenheit des Managements für eben diese Zeit zu sorgen.

In modernen, prozessorientierten Unternehmen sollte durch ein Change Management wie bei ITIL[4] eine ausreichende Dokumentation selbstverständlich sein. Diese erfolgt meist in Form von klassischen Dokumenten (Textverarbeitung, Tabellendokumente). Für komplexe Konfigurationen sind diese aber schwer zu strukurieren und umständlich, wenn die Konfigurationen oder Teile davon ebenfalls enthalten sein sollen. Für die praktische Arbeit an den IT-Systemen sind solche Dokumente problematisch.

Die Dokumentation von Konfigurationseinträgen und -änderungen ist oft nicht in der Konfiguration selbst möglich, oder nur eingeschränkt. Insbesondere sind die Möglichkeiten bei verschiedenen Geräten nicht einheitlich. Die Zuordnung der Dokumentation zu den Teilen der Konfigurationen ist daher schwierig. Welche Kommentare gehören zu welchen Revisionen, welche Zeilen der Konfiguration kommen aus welchem Change Request und umgekehrt?

Bei lange laufenden Installationen kann die Suche nach und in den relevanten Dokumenten sehr langwierig sein. Schnell kommen viele Dokumente zusammen.

Eine Form der Verknüpfung der Konfigurationen mit der Dokumentation, die über Webbrowser universell und unabhängig vom Betriebssystem des Benutzers und unabhängig von den Gerätetypen und den Dateiformaten der Dokumentation ist, wäre daher sinnvoll. Dabei sollte sowohl der Weg von der Revisionsinformation zum Konfigurationsteil als auch umgekehrt möglich sein.

2.3 Wirkbetrieb vs. Forschung und Entwicklung

Im Wirkbetrieb kommerzieller Netze stellt sich ein Life Cycle der Konfigurationen vielfach durch "lineare" Fortschreibung nach einer Aufbauphase dar. Es werden zum Beispiel sukzessive neue Kunden auf einer Plattform freigeschaltet, die stark schematisierten Massnahmen unterliegt. Die Dokumentation durch Change Request Prozesse auf Datenbanken kann gut strukturiert werden. Die Anzahl der Geräte kann sehr hoch werden.

Umfassende Änderungen und Tests finden eher auf einer kleinen Anzahl von Geräten statt, z.B. im Pre-Sales Bereich und im Life Cycle Management.

In der Forschung und Entwicklung stellt sich das oft anders dar. Die Nutzung eines Testbeds für verschiedene Vorhaben kann für häufige, umfassende Änderungen sorgen. Eine parallele Mehrfachnutzung bedeutet eine Überlappung der Gerätegruppen verschiedener Projekte und damit der Konfigurationen. Häufige Änderungen ohne Change Prozesse durch stark fluktuierende Personenkreise machen eine Dokumentation der Gesamtzustände schwierig. Die Anzahl der Geräte ist eher gering.

[4] „IT Infrastructure Library“ für Service Management, siehe http://www.itil.org/

Hilfreich wäre also die Möglichkeit, überlappende Gerätegruppen für verschiedene Szenarien auf einem Gerätepool und Mandantenfähigkeit auf den selben, verteilten Geräten der Testbeds definieren zu können.

2.4 Wiederverwendung

Ein besonders hervorzuhebender Aspekt ist die Möglichkeit einen Gerätepark routinemässig für andere Zwecke umnutzen zu können. Neben dem funktionalen Vorteil, mehrere Projekte sowohl nacheinander als auch alternierend auf den selben Geräten durchführen zu können, können finanzielle Einsparungen erzielt werden. Dies betrifft Investitionen und laufende Kosten, da zusätzliche Geräte auch erhebliche Folgekosten durch Raummiete, Wartungsverträge und Energieverbrauch nach sich ziehen.

Für die Umnutzung müssen in der Organisation geeignete Prozesse etabliert werden. Diese sollten durch Werkzeuge für automatisierte Umkonfiguration von Geräten und Gerätegruppen unterstützt werden. Eine Skriptfähigkeit der Werkzeuge würde komplexere Abläufe ermöglichen, etwa vorhergehende Prüfungen der Verfügbarkeit von Geräten und anderen Ressourcen.

2.5 Verteilte Teams, Web-Kollaboration

Netze und Testbeds werden weiter zunehmend von verteilt arbeitenden Gruppen in verschiedenen Instituten, Firmen und/oder Zeitzonen benutzt oder betreut. Dies trifft auch auf große Projekte zu. Die Kooperation und Koordinierung findet immer öfter über Wikis, webbasierte Archive, Mailing Listen und Telekonferenzen statt.

In diesen Situationen ist es von großem Vorteil, wenn direkte WWW-Links auf Konfigurationen einzelner Geräte und Gerätegruppen in einem Configuration Management Repository möglich sind, insbesondere auf spezifische Revisionen. Dann kann z.B. in einer Change Request Dokumentation, die in einem Wiki-Eintrag gepflegt wird, durch ein HTTP-URL direkt auf ganz bestimmte Zustände eines gesamten Testbeds verwiesen werden. Naheliegend wären hier natürlich Links auf die Vorher- und Nachher–Zustände und auf die zugehörigen Changesets, die die konkreten Änderungen darstellen.

2.6 Existierende Lösungsansätze

Für die ermittelten Aufgaben wurden existierende Lösungen für Netzwerkgeräte betrachtet. Diese wurden anhand der folgenden Kriterien bewertet:

- Archivierung von Konfigurationen und Zuständen
- Herstellerunabhängigkeit
- Erweiterbarkeit
- Mandantenfähigkeit
- Skriptfähigkeit
- Deployment/Rollback Funktion
- günstige Lizenzbedingungen für TestBed-Betreiber
- Benachrichtigung über Änderungen

Für den Testbedbetrieb sind gerade die Herstellerunabhängigkeit und Erweiterbarkeit ein kritischer Punkt. Die Testbeds müssen heterogene Netze verwalten können, speziell bei Interoperabilitätstests. Oft muss mit neuartigen Geräten oder neuer Software gearbeitet werden, die möglicherweise selbst von den herstellereigenen Management-Programmen noch nicht unterstützt werden. Dann muss der Testbed-Betreiber die benötigten Funktionen auch selbst implementieren können.

2.6.1 Juniper Network and Security Manager (NSM)

„Juniper NSM[5]" ist eine kommerzielle, proprietäre Lösung. Sie unterstützt Configuration Management, Policy Management, Software-Image Mangement, Logging, und Monitoring. Insbesondere können netzwerk-weite Aktionen ausgeführt werden.

Die Lösung ist für Juniper Router, Switche, Firewalls und IDS Geräte geeignet. Geräte anderer Hersteller werden nicht unterstützt und eigene Erweiterungen sind nicht möglich. CLI Clients sind in anderen Produkten implementiert. Eine Mandantenfähigkeit ist nicht gegeben.

2.6.2 CiscoWorks Ressource Manager Essentials (RME)

„CiscoWorks RME[6]" ist eine kommerzielle, proprietäre Lösung. Die Funktionen umfassen Inventar Management, Configuration Management, Software-Image Management, Change-audit Dienste und Syslog Analyse.

Die Lösung ist für Cisco Router, Switche, Firewalls, IDS und viele andere Geräte geeignet. Geräte anderer Hersteller werden nicht unterstützt und eigene Erweiterungen sind nicht möglich. Es gibt ein Web-GUI, aber keine CLI-Clients. Eine Mandantenfähigkeit scheint nicht gegeben zu sein.

2.6.3 Kiwi CatTools

„Kiwi CatTools[7]" ist eine kommerzielle Lösung, ist aber herstellerunabhängig. Es können Konfigurationen archiviert und verglichen werden. Änderungen werden per Email mitgeteilt. Massenänderungen können mit Kommando-Templates automatisiert werden.

Die Software ist nicht erweiterbar und läuft nur unter Windows. Ein Web-GUI ist nicht vorhanden. Eine Mandantenfähigkeit ist nicht gegeben.

2.6.4 Really Awesome New Cisco confIg Differ (RANCID)

„RANCID[8]" ist eine Open Source Lösung. Die Software ist herstellerunabhängig und unterstützt alle relevanten Netzwerkgeräte. Der Fokus liegt auf der Archivierung der Konfigurationen und von Zuständen der Geräte. Benachrichtigungen über Änderungen werden per Email verschickt. Als Repositories können CVS und Subversion eingesetzt werden.

Eigene Erweiterungen sind durch Skripte leicht möglich. Ein eigenes Web-GUI ist nicht implementiert, für die verwendeten Repositories gibt es aber eine Reihe von GUI Implementierungen. Im CLI kann RANCID direkt aufgerufen werden, für die Repositories gibt es eine Reihe von Clients. Eine Mandantenfähigkeit ist gegeben.

2.6.5 Network Device Change Control (NDCC)

„NDCC[9]"ist Open Source und herstellerunabhängig. Es speichert Konfigurationen in einer MySQL Datenbank und kann diese und Unterschiede über ein Web-GUI anzeigen.

---

[5] http://www.juniper.net/products_and_services/firewall_slash_ipsec_vpn/netscreen_security_manager/index.html

[6] http://www.cisco.com/en/US/products/sw/cscowork/ps2073/index.html

[7] http://www.kiwisyslog.com/kiwi-cattools-overview/

[8] http://www.shrubbery.net/rancid/

[9] http://ndcc.totallygeek.com/

Es werden keine Zustände gespeichert. Eine Erweiterung ist möglich. Eine Mandantenfähigkeit ist nicht gegeben.

2.6.6 The NOC Project

„The NOC Project[10]" ist ein Operation Support System nach dem FCAPS-Modell („Fault, Configuration, Accounting, Performance, Security").

Die Software ist herstellerunabhängig und unterstützt verschiedene Netzwerkgeräte. Eine Mandantenfähigkeit ist nicht gegeben. Das Projekt befindet sich noch in einem frühen Stadium.

2.6.7 Zusammenfassung

Die Hersteller von Netzwerkgeräten unterstützen nur ihre eigenen Produkte. Erweiterungen können nicht vom Benutzer selbst vorgenommen werden. Bei einigen Tools werden auch Informationen zur verwendeten Hardware gespeichert, darüber hinausgehende Zustände werden jedoch nicht berücksichtigt.

Fast alle herstellerunabhängigen Lösungen sind Open Source Projekte, die aufgrund des Mangels von kommerziellen Lösungen entwickelt wurden. Lediglich die „Kiwi CatTools" stellen eine Ausnahme dar, leider ist diese Lösung nicht mandantenfähig und hat keine Webschnittstelle.

Von den Open Source Lösungen hat RANCID den größten Funktionsumfang. Es verfügt zwar nicht über eine eigene Web-Schnittstelle. Durch die Verwendung von Subversion für die Versionskontrolle kann aber aus einer Reihe verschiedener OSS GUI gewählt werden, die erweiterbar sind. Die wesentlichen Vorteile von RANCID liegen in der universellen Methodik. Für jeden Gerätetyp können beliebige Befehle auf den Geräten ausgeführt werden, die alle Informationen abrufen, die von Interesse sind. Ausser den Konfigurationen und den informationen über die Hardware können insbesondere die damit erzeugten Zustände wie Routingtabellen und Schnittstellen-Stati mit archiviert werden. Alle Informationen werden dabei in einer Datei pro Gerät und Revision konsolidiert. Diese kann leicht verarbeitet werden und Änderungen sind gut zu erkennen. Neue Befehle können leicht hinzugefügt werden. Die Unterstützung neuer Gerätetypen ist mit vertretbarem Aufwand möglich.

[10] http://trac.nocproject.org/trac/

3 Aufgabenstellung

Viele Forschungs- und Entwicklungseinrichtungen betreiben Testbeds aus verschiedenen Netzwerkgeräten und Servern von Cisco Systems, Juniper Networks, Sun Microsystems und anderen Herstellern.

Das Ziel dieser Arbeit ist die Erstellung eines Systems zur zentralen Verwaltung der Konfigurationen und Zustände von Testbedkomponenten (Router, Switche, Server etc.) für verteilte, föderierte Testbeds. Es soll verteiltes, webbasiertes Arbeiten unterstützt werden. Das System soll voll mandantenfähig sein. Die Benutzung soll außer mit einem Web-GUI auch mit dem CLI und aus Skripten möglich sein.

3.1 Die Beta-Plattform

Diese Arbeit ist Teil des Projektes „Forschungsnahe Beta-Plattform für die Zukunft des Internets", einer Initiative für die effiziente und nachhaltige Entwicklung innovativer IP-Dienste. Die Motivation wird vom Projekt so beschrieben[11]:

> *„Der Übergang in die Informationsgesellschaft und der damit verbundene Eintritt in neue Märkte unter der Nutzung neuer Technologien erfordert neue Herangehensweisen und Methoden bei der Entwicklung von Kommunikationsdiensten und Lösungen. Unternehmen, Wissenschaftler und Entwickler mit guten Ideen in digitalen Geschäftsfeldern stehen vor komplexen Fragestellungen, wenn es an die Validierung der Ergebnisse und den Transfer von prototypischen Implementierungen hin zu Produkten geht. Folgende Fragestellungen fassen diese Punkte zusammen:*
>
> *Wie können Forschungsergebnisse und Prototypen in realitätsnahen Umgebungen über Simulationen hinaus validiert werden?*
>
> *Wie können Anreize zum Austausch und zur Wiederverwertung von vorliegenden Forschungs- und Entwicklungsergebnissen geschaffen werden?*
>
> *Wie kann die Reichweite von Forschungsergebnissen erhöht werden?*
>
> *Wie können existierende Testbed-Infrastrukturen effizienter genutzt und Doppelfinanzierungen vermieden werden?*"

Durch die föderierte Nutzung von existierenden, verteilten Testbeds soll eine dafür geeignete Infrastruktur ermöglicht werden. Ein zentrales Hardware Repository enthält darin Informationen über die Geräte in den verteilten Testbeds, aus denen die Nutzer geeignete Geräte auswählen können. Eine Zuteilung der Geräte zu den Nutzern wird durch ein zentrales Buchungssystem ermöglicht. Die Kopplung der zugeteilten Geräte wird in getrennten virtuell-privaten Netzwerken (VPN) pro Nutzer realisiert.

Es soll eine zentrale Verwaltung der Konfigurationen implementiert werden, um die Arbeiten, die die Nutzer an den Geräten durchführen, kontinuierlich zu archivieren und zu dokumentieren.

Diese zentrale Verwaltung der Konfigurationen und der Zustände ist Gegenstand dieser Arbeit. Zusätzlich soll die Verwaltung lokal in den Testbeds genutzt werden können.

[11] Aus http://www.bmwi.de/BMWi/Navigation/Service/publikationen,did=279734.html

3.2 Aufgaben im Detail

Die Aufgaben, die sich im laufenden Betrieb stellen, und welche Lösungsansätze verfolgt werden sollen, sind hier genauer dargestellt:

Wurde etwas geändert?

> Durch zyklischen Abruf und Vergleich der Konfigurationen und Zustände soll eine Revisionshistorie erzeugt werden. An der Revisionsnummer und deren Datum soll erkennbar sein ob sich etwas geändert hat.

Was hat sich geändert und wann?

> Für jedes Gerät sollen die Unterschiede zwischen den Zuständen zeitbezogen gespeichert werden und die Differenzen zwischen den Zuständen aufzeigen. Die verwendete Software soll Details zur Hardware, Software, Konfiguration und bestimmten dynamischen Zuständen (soft state) jedes Gerätes speichern. Es soll erkennbar sein, wenn ein Software Update durchgeführt, ein Hardware Modul entfernt oder ein VLAN inaktiv wurde.

Warum wurde etwas geändert und von wem?

> Durch Einträge in einer webbasierten Revisionshistorie sollen die Mitarbeiter Änderungen zentral dokumentieren können.

Was war der Gesamtzustand des Labors zu einem bestimmtem Zeitpunkt?

> Die Versionskontrolle soll nicht nur den Zustand eines Gerätes zu einem bestimmtem Zeitpunkt anzeigen, sondern auch den umgekehrten Weg ermöglichen. Man soll über die Revisionsnummer die Zustände aller Geräte zu einem bestimmten Zeitpunkt erhalten.

Referenzsysteme und Reproduktion von Szenarien

> Durch das Exportieren von Gesamtzuständen sollen Referenzinstallationen für Projekte und Fehlerszenarien zur Reproduktion dokumentiert werden können.

Was ist für einen Rollback notwendig?

> Um einen früheren Zustand wieder herzustellen, sollen die Differenzen aller Geräte zwischen zwei bestimmten Zuständen ermittelt werden können.

Welche Massnahmen sind für einen Rollout nötig?

> Ähnlich dem Vorgehen beim Rollback sollen Änderungen an einer Referenzinstallation oder einem Fehlerszenario gesammelt dargestellt werden können, um die notwendigen Massnahmen in einem Zielnetz aufzuzeigen.

Insbesondere sollen auf Basis dieser Funktionen Geräte für andere Zwecke wiederverwendbar werden, indem frühere Situationen wiederhergestellt werden können. Sie müssten dann für Referenzsysteme nicht mehr dediziert sein und unproduktive Kapitalbindungen könnten vermieden werden.

3.3 Strategie

Der Einsatz von kommerziellen Lösungen wurde nicht ausgeschlossen. Es hat sich aber gezeigt, das die kritischen Funktionen der Herstellerunabhängigkeit und der Erweiterbarkeit in den untersuchten Produkten nicht ausreichend sind. Änderungen und Anpassungen bei den Herstellern anzufragen ist schwierig und zeitaufwändig.

Im Rahmen des Beta-Plattform Projektes war eine Neuentwicklung nicht sinnvoll. Diese wäre zu langwierig und zu teuer. Eine Eingewöhnung in komplette neue GUI und CLI gegenüber dem Einsatz bewährter Lösungen könnte zudem die Akzeptanz bei den Nutzern verringern.

Open Source Lösungen haben hier deutliche Vorteile. Da sie von unabhängigen Entwicklern stammen, sind sie meist herstellerunabhängig. Vor allem sind eigene Anpassungen und Erweiterungen in der Regel gut möglich, da der Quellcode verfügbar ist. Unterstützung durch Autoren und Nutzerforen können sehr hilfreich sein. Ausserdem ist es von Vorteil die Kosten gering zu halten, Open Source Lösungen sind in der Regel nicht kostenpflichtig.

Open Source Lösungen zu erweitern ist aber nicht trivial. Es sind Umsicht und Disziplin notwendig, um die Wartbarkeit zu gewährleisten. Der Fokus muss beibehalten werden und das richtige Maß und die richtige Stelle müssen wohl überlegt sein. Dies gilt insbesondere dann, wenn die Änderungen in das Open Source Projekt einfliessen sollen um die Akzeptanz bei den Entwicklern zu gewährleisten.

Schliesslich soll der Betrieb der zentralen Lösung möglichst wenig Aufwand für die Testbed Betreiber bedeuten. Es sollen möglichst geringe Kosten für die Testbeds anfallen.

Daher wird für diese Arbeit die Integration und Erweiterung bewährter, existierender Open Source Lösungen nach dem Bausteinprinzip angestrebt. Datenschnittstellen sollen möglichst im XML Format ausgelegt werden, um die Interoperabilität zu gewährleisten, insbesondere zum Abruf der Gerätedaten vom Hardware Repository der Beta-Plattform.

Um den Testbed-Betreibern auch ohne positive Effekte der Beta-Plattform einen Anreiz zu bieten, soll die Anwendung sowohl lokal standalone für jedes Testbed als auch zentral als übergreifende Lösung für die Beta-Plattform geeignet sein. Die Inbetriebnahme der Standalone-Lösung soll einfach sein, z.B. in Form einer virtuellen Appliance.

Die Arbeit wird als Open Source Lösung in das Software Repository der Beta-Plattform aufgenommen und kann in den Testbeds und in anderen Projekten wiederverwendet werden. Es werden der Quellcode und lauffähige virtuelle Appliances bereitgestellt.

4 Architektur

Mittels Integration und Erweiterung von mehreren Open Source Lösungen zu einer virtuellen Appliance auf Basis von VMware[12]/XEN[13] soll eine gebrauchsfertige Plattform geschaffen werden.

Die Beta-Plattform betreibt ein zentrales Buchungssystem mit einem Hardware Repository und ein zentrales Configuration Management (CM). Das Hardware Repository enthält Informationen über die Geräte in den verteilten Testbeds. Jeder Nutzer bekommt über das Buchungssystem der Beta-Plattform die benötigten Geräte zugeordnet. Über das für ihn eingerichtete VPN kann der Benutzer auf diese Geräte zugreifen.

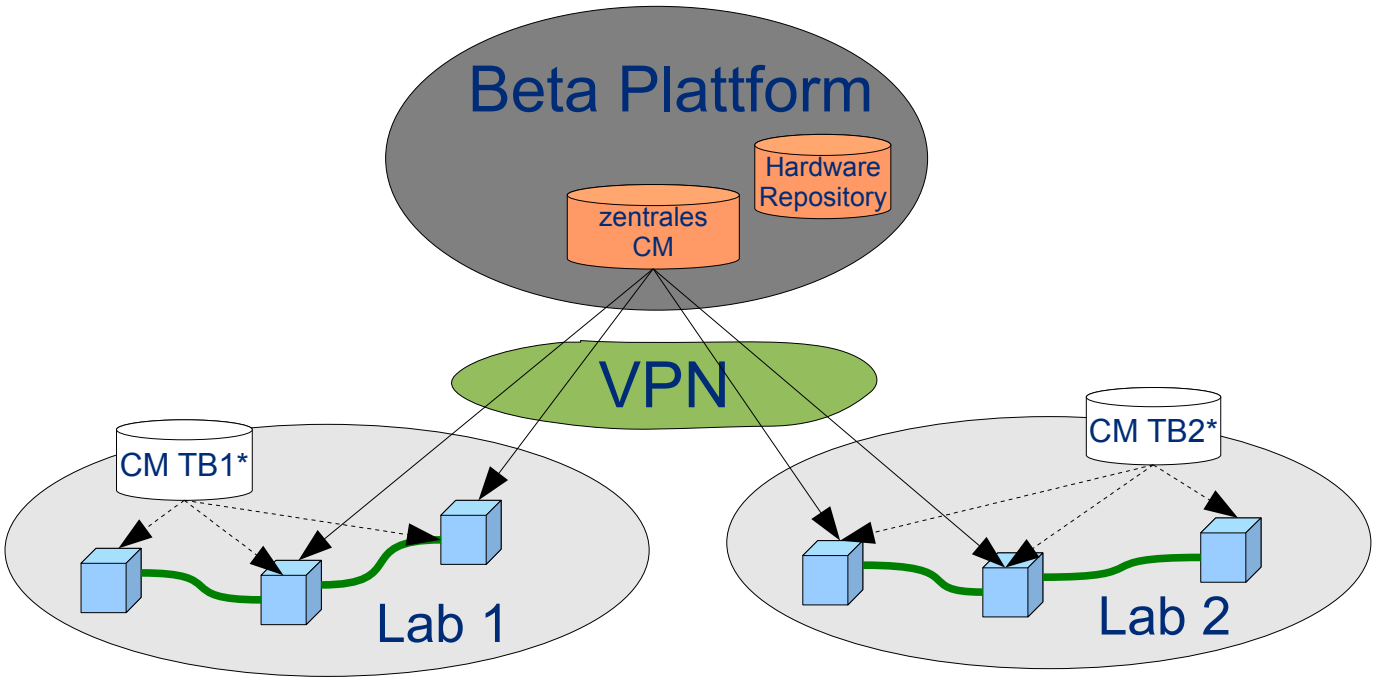


Abbildung 1: Architektur des Beta-Plattform Configuration Management

Das Configuration Management ruft vom Buchungssystem und dem Hardware Repository die Informationen über die zugeordneten Geräte ab und ermöglicht dem Nutzer die Verwaltung seiner Geräte und deren Konfigurationen und Zustände.

Unabhängig davon kann in jedem Testbed eine separate, lokale Instanz des Configuration Management betrieben werden, die auf den selben oder anderen Geräten arbeitet. Das Configuration Management wird dann unabhängig von der Beta-Plattform betrieben.

Bezüglich der CM-Repositories soll der Zugriff auch ohne spezielle Clients möglich sein, die an bestimmte Betriebssysteme gebunden sind. Insbesondere soll HTTP für den Zugriff aus embedded Devices und Browsern möglich sein. Für weitere Automatisierung sollen auf der Komandozeile (Command Line Interfaces - CLI) und in der Web-Schnittstelle (Graphical User Interface - GUI) möglichst gleiche Funktionen implementiert sein.

[12]x86 Virtualisierung, http://www.vmware.com/
[13]Virtualisierungshypervisor, http://www.xen.org/

5 Implementierung

Die Kernfunktionen der Implementierung bilden „RANCID" als Kollektor der Konfigurationen mit der Versionsverwaltung „Subversion" und dem GUI „Insurrection". Die Wahl fiel auf RANCID, da es herstellerneutral und sehr gut erweiterbar ist. Subversion ist flexibler als CVS und es gibt verschiedene umfangreiche Web-GUI Implementierungen.

Durch Umstellung der relevanten RANCID Konfigurationsdateien auf die eXtensible Markup Language (XML[14]) wurde eine Integration weiterer Softwarekomponenten und der Beta-Plattform ermöglicht. So konnten auch Erweiterungen gut implementiert werden, was mit dem alten Format umständlich gewesen wäre.

Mit dem Subversion Web-GUI „Insurrection" können mandantenfähig Repositories angelegt und benutzt werden. Auch Insurrection ist gut erweiterbar. Ausserdem sind damit zur Benachrichtigung über Änderungen an den Konfigurationen zusätzlich zu Email auch Web Feeds mit Really Simple Syndication (RSS) und Atom Syndication (ATOM) möglich. Insurrection wurde erweitert, um die RANCID Funktionen und andere Erweiterungen anzusteuern.

Alle Funktionen können auch aus dem CLI und in Skripten benutzt werden. Die Repositories können auch mit einer grossen Zahl von Clients benutzt werden. Einfache Abrufe sind über HTTP auch direkt aus dem CLI von Geräten möglich.

Die anderen Funktionen der kommerziellen Lösungen wie Inventarverwaltung, Policy Management und Monitoring werden in der Anwendung für die Beta-Plattform anderweitig implementiert oder nicht benötigt.

5.1 Redaktionelle Hinweise

Im Folgenden wird der Begriff „Lab" synonym für „Testbed" und „Labor" verwendet. Der Ausdruck „~labmgmt" steht für den Verzeichnispfad „/home/labmgmt". Zur sinnvollen Darstellung der Pfade gilt im Folgenden als Beispiel das Repository „bp". Es können aber beliebig viele Repositories mit Namen aus Buchstaben und Zahlen verwendet werden. Diese werden in der Regel mit dem GUI angelegt.

5.2 Rahmenbedingungen

Für die Realisierung wurden folgende Ressourcen benötigt, die nicht Teil der Arbeit sind:

- Eine virtuelle Maschine auf einem Intel Server
- Eine Linux Debian [DEBIAN] Installation darauf
- Ein Secure Socket Layer (SSL) Zertifikat für den Web Server
- Das Hardware Repository der Beta-Plattform mit Schnittstellen zum Abruf der Geräteinformationen.

Auf dieser Basis wird die Arbeit als virtuelle Appliance eingerichtet. Die formelle Abgabe erfolgt in Form eines Images (Abbild) der virtuellen Maschine auf einer DVD.

[14] http://www.w3.org/XML/

5.3 Beschreibung der Subsysteme

Zur Verdeutlichung der Zusammenhänge sind im folgenden Schaubild die wesentlichen Bestandteile des Systems grafisch dargestellt.

Das hellblaue Segment am oberen Rand stellt das GUI und den Web Server dar. Über das Common Gateway Interface (CGI) und Distributed Authoring and Versioning (DAV) Schnittstellen wird von hier auf das Subversion Repository, RANCID und das Strommanagement zugegriffen.

Die blauen Würfel stellen die Geräte dar. Auf sie wird mit den Protokollen Telnet, Secure Shell (SSH) und dem Simple Network Management Protocol (SNMP) zugegriffen.

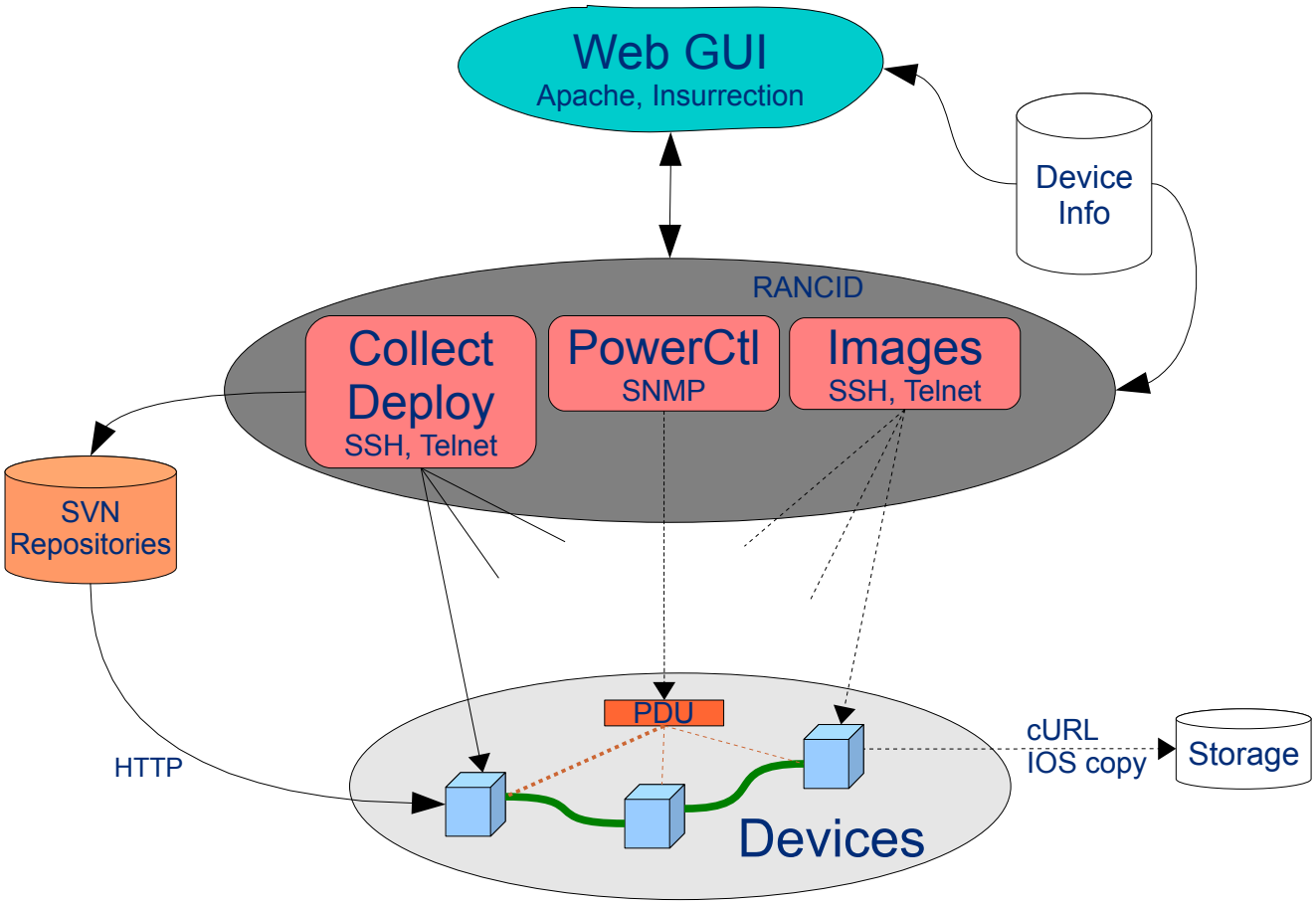


Abbildung 2: Integration der Subsysteme

Die Konfigurationen und Zustände werden nach dem Abruf aus den Geräten konsolidiert und in den Subversion Repositories archiviert.

Bei bestimmten Gerätetypen kann mit der Deploy Funktion eine archivierte Konfiguration auf Geräte übertragen werden.

Die Sicherung von Software Images und Dateien erfolgt auf je Gerät individuell einstellbaren Storage Servern. Dazu können eine ganze Reihe von Protokollen verwendet werden, z.B. HTTP, FTP, SCP usw.

Wenn geeignete Power Switche (Power Distribution Unit - PDU) vorhanden sind können mit SNMP die daran angeschlossenen Geräte ein- und ausgeschaltet werden.

5.4 Einbettung in die Beta-Plattform

Die Föderation der Testbeds wird durch die Beta-Plattform technisch und organisatorisch koordiniert. Für die Nutzer ist die Beta-Plattform der alleinige Anlaufpunkt.

Der grundsätzliche Ablauf ist im Folgenden dargestellt:

1. Die Betreiber der Testbeds geben der Beta-Plattform Informationen über ihre Geräte und bereiten diese für die Nutzung vor
2. Die Beta-Plattform prüft und sammelt diese Informationen im zentralen Hardware Repository und vergibt global eindeutige Geräte-IDs und -Namen für alle Geräte
3. Der Nutzer wählt daraus die benötigten Geräte für seine Anwendung
4. Der Nutzer bestimmt Anzahl und Namen von Gerätegruppen
5. Die Beta-Plattform erzeugt entsprechende Geräte-Konfigurationsdateien je Nutzer
6. Die Beta-Plattform richtet ein IPSEC VPN für den Nutzer ein, in dem alle benötigten Geräte erreichbar sind
7. Die Beta-Plattform erzeugt die Configuration Management Repositories, die Einstellungen werden in der jeweiligen „.conf.xml" Datei abgelegt. Danach werden die Geräte-Konfigurationsdateien vom Hardware Repository abgerufen und in die lokalen Konfigurationsdateien „devices.xml" in jeder Gerätegruppe abgelegt
8. Die Beta-Plattform teilt dem Nutzer seinen Login für das Configuration Management und die Credentials (Username/Passwort) für die Geräte mit
9. Der Nutzer meldet sich beim Configuration Management GUI an und stösst den initialen Abruf und die Archivierung der Konfigurationen der Geräte an
10. Der Nutzer kann nun sein Testbed und das Configuration Management benutzen

Die Geräte in den Testbeds müssen den Zugriff mit den konfigurierten Credentials zulassen, z.B. mit RADIUS, TACACS+ oder lokalen Einträge auf Geräten. Dies muss von den Testbedbetreibern selbst realisiert werden, die Beta-Plattform und diese Arbeit stellen diese Funktionen nicht zur Verfügung.

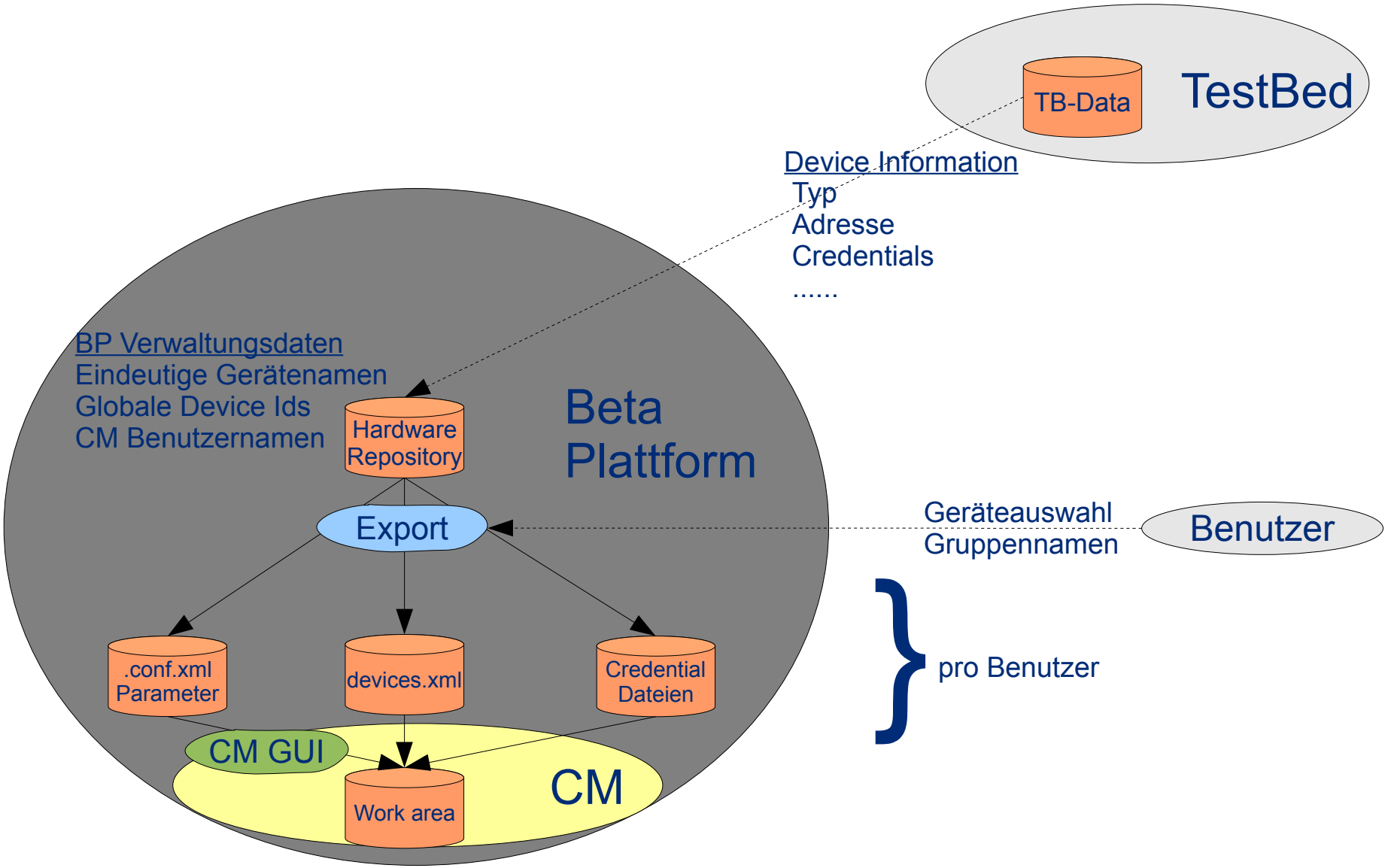


Abbildung 3: Fluss der Konfigurationsdaten

5.5 Kontroll- und Datenflussmodell

Für jeden Nutzer (Beispiel: „bp“) gibt es drei Sätze an Konfigurationsdaten

1. Eine Datei „devices.xml“ mit den Geräteinformationen
2. Je Gerät eine Datei mit den Login Credentials
3. Die Parameter für das Einrichten des CM mit dem GUI (.conf.xml)
    - Name des Repositories, dieser entspricht dem Nutzername im GUI
    - Namen der Gerätegruppen
    - Betriebsmodus Standalone oder Beta-Plattform

Die Parameter für das Einrichten des CM mit dem GUI werden auf die Defaultwerte aus „/home/labmgmt/rancid/_template.conf.xml“ angewendet und ergänzen oder überschreiben diese. Das Ergebnis wird als „bp.conf.xml“ gespeichert.

Der Ablauf im Detail ist wie folgt:

1. Auf dem CM liegt „_template.conf.xml“ mit den CM-lokalen Default Pfaden und Pfad-Prefixen für RANCID Arbeitsbereiche, Subversion Repositories usw.
2. Mit dem CM Admin GUI oder CLI wird RANCID aufgerufen
3. Anhand der Argumente wird „_template.conf.xml“ für den jeweiligen Benutzer zu „bp.conf.xml“ ergänzt / modifiziert
4. Die Arbeitsbereiche für den Nutzer und das Repository werden angelegt
5. Per <DEVICES_URL> aus „bp.conf.xml“ wird das „devices.xml“ für den Nutzer abgerufen. Im Wirkbetrieb der Beta-Plattform wird dies vom Hardware Repository geladen. Sie wird als „devices.xml“ für die interne Nutzung im Workspace des CM abgelegt
6. Für jedes Gerät ist in „devices.xml“ eine Information <loginrcURL> enthalten, mit der die Credential-Dateien abgerufen werden können. Im Wirkbetrieb der Beta-Plattform werden sie vom Hardware Repository geladen.
7. Alle Credentials je Nutzer werden zu einer Datei „.cloginrc“ zusammengefasst

Der Abruf der Geräteinformationen und Credentials kann später vom Nutzer über das GUI wiederholt werden, falls sich Änderungen an seinen Geräten ergeben haben.

Die Installation enthält das Repository „bp“ als Beispiel mit folgenden Default-Werten:

- CVSROOT/BaseURL „/home/labmgmt/repositories“
- Gerätegruppen „g1 g2“
- Per <DEVICES_URL> „http://localhost/BP„ werden abgerufen
    - „/home/labmgmt/BP/bp/devices.xml“ mit den Geräteinformationen
    - „/home/labmgmt/BP/bp/d?“ Login Credentials je Gerät

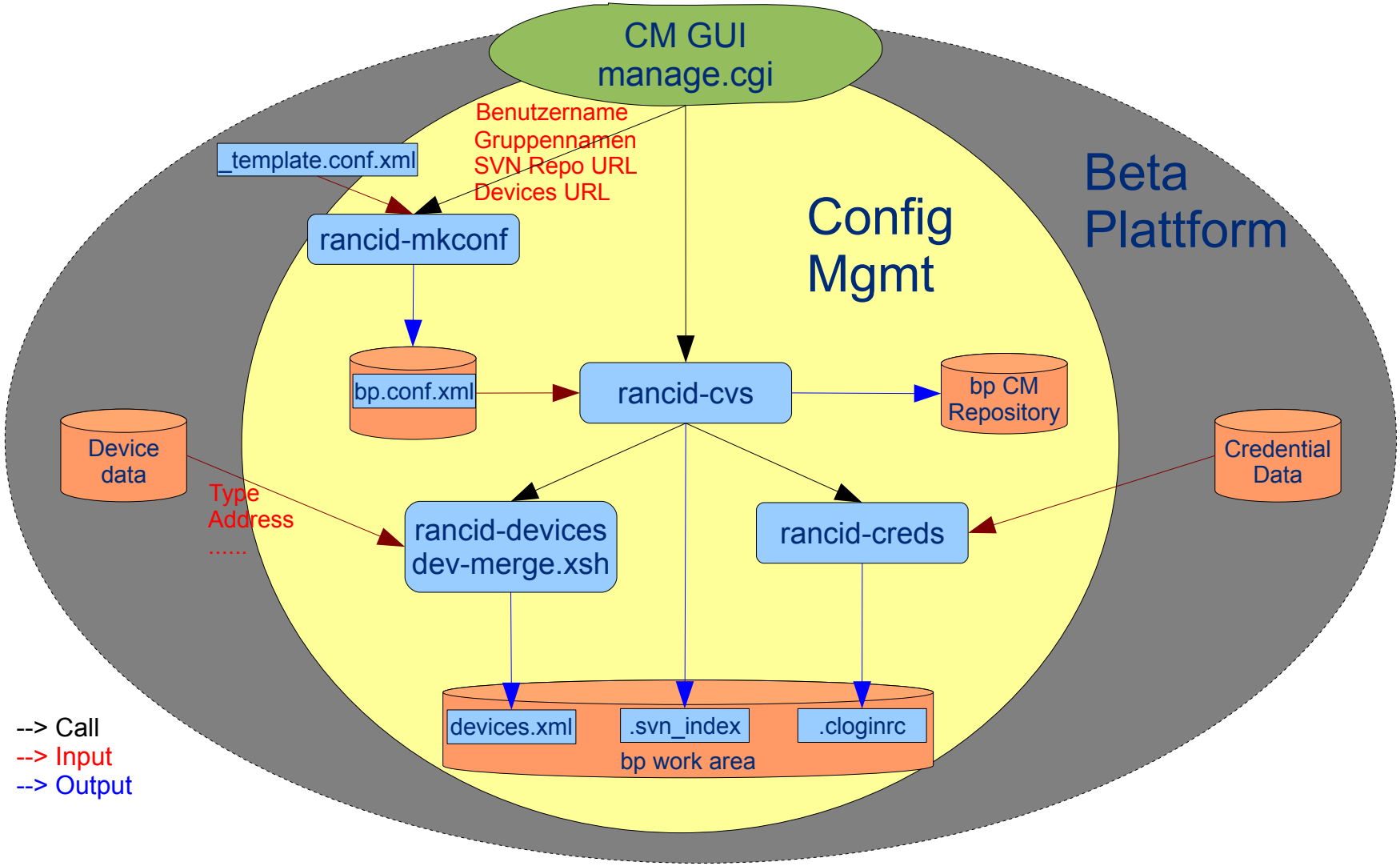


Abbildung 4: Kontroll- und Datenflussmodell

5.6 Funktionen

Die Funktionen der Installation werden in diesem Kapitel getrennt nach Aufgabengebiet dargestellt. Die genaue Beschreibung der Installation, Konfiguration und der Nutzung wird in den darauf folgenden Kapiteln beschrieben.

5.6.1 Abrufen, Konsolidieren der Konfigurationen

Mit Hilfe der Software [RANCID] werden umfangreiche Informationen der Geräte abgerufen:

- Hardware Modell und Versionen des Chassis und der installierten Module, der Typ angeschlossener Kabel, Netzteile, Lüfter und deren Zustände
- Die aktuell laufende Software-Version
- Eine Auflistung angeschlossener Speichermedien und deren Inhaltsverzeichnisse
- Statische und dynamische Zustände wie konfigurierte und gelernte virtuelle LAN (VLAN)
- die eigentliche Konfiguration

Die Software führt dazu einen Login per Telnet oder SSH[15] auf alle Geräte durch, ruft die notwendigen Daten ab und konsolidiert sie in einer einzigen Datei je Gerät.

Diese Dateien werden dann in einem revisionskontrollierten Repository gespeichert.

Der Abruf von den Geräten kann auf verschiedene Arten erfolgen:

- Zyklisch per CRON[16]
- Zielzeitbezogen per AT
- Manuell mit der grafischen Oberfläche
- Manuell in der UNIX-Kommandozeile
- Aus Skripten oder Programmen heraus

Prinzipiell unterstützt die Software Cisco Router, Switches, WLAN Access Points mit dem Betriebssystem IOS, Cisco Switches mit dem Betriebssystem CatOS, Juniper Router, Foundry Switches, HP ProCurve, UNIX Systeme und diverse andere

Hier ein gekürztes Beispiel, das vollständige Beispiel ist im Anhang zu finden.

[15] „Secure Shell“ zum kryptierten Zugriff auf Geräte anstelle von Telnet
[16] CRON und AT sind Programme der Jobsteuerung von UNIX Derivaten

`!RANCID-CONTENT-TYPE: cisco`
`!`
`!Chassis type: 3640 - a 3600 router`
`!CPU: R4700, R4700 CPU at 100MHz, impl 33, Rev 1.2`
`!`
`!Memory: main 124928K/6144K`
`!Memory: nvram 125K`
`!`
`!Processor ID: 00000000`
`!`
`!Power: Redundant Power System is  present.`
`!`
`!Image: Software: C3640-IS-M, 12.4(13b), RELEASE SOFTWARE (fc3)`
`!Image: Compiled: Tue 24-Apr-07 20:31 by prod_rel_team`
`!Image: tftp://255.255.255.255/unknown`
`!`
`!ROM Image: Version 12.4(13b), RELEASE SOFTWARE (fc3)`
`!`
`!`
`!`
`!Flash: System flash directory:`
`!Flash: File  Length   Name/status`
`!Flash:   1   840      vlan.dat`
`!Flash: [8388604 bytes used, 0 available, 8388604 total]`
`!Flash: 8192K bytes of processor board System flash (Read/Write)`
`!`
`!Flash: nvram: Directory of nvram:/`
`!Flash: nvram:   123  -rw-        1484         <no date>  startup-config`
`!Flash: nvram:   124  ----           5         <no date>  private-config`
`!Flash: nvram:     1  -rw-           0         <no date>  ifIndex-table`
`!Flash: nvram: 129016 bytes total (126451 bytes free)`
`!`
`! <gekürzt>`
`!`
	Informationen über Hardware, Software, Firmware, Dateisysteme und andere Zustandsdaten

`config-register 0x2102`
`version 12.4`
`service timestamps debug datetime msec`
`service timestamps log datetime msec`
`no service password-encryption`
`!`
`hostname Cisco_ID_03`
`!`
`boot-start-marker`
`boot-end-marker`
`!`
`enable secret 5 $1$IJy2$hxIzPICwwgvuwXolgt5tq1`
`!`
`no aaa new-model`
`memory-size iomem 5`
`!`
`ip cef`
`!`
`interface Loopback0`
` description Yeah.`
` ip address 192.168.0.100 255.255.255.255`
`!`
`interface FastEthernet0/0`
` ip address dhcp`
` speed auto`
` half-duplex`
`!`
`interface FastEthernet1/0`
` no ip address`
` shutdown`
` duplex auto`
` speed auto`
`!`
`router eigrp 1`
` network 192.168.1.0`
` network 192.168.0.0`
` no auto-summary`
`!`
`ip http server`
`!`
`control-plane`
`!`
`line con 0`
`line aux 0`
`line vty 0 4`
` password vlab`
` login`
`!`
`end`
	Eigentliche Konfiguration des Gerätes

Tabelle 1: Beispiel für Gerätezustände und Konfiguration

5.6.2 Archivierung und Versionskontrolle

Die Konfigurationen und Zustandsdaten werden in einem Subversion [SVN] Repository gespeichert. Subversion ist eine alternative Software zu CVS [CVS].

Die gespeicherten Zustände zu einem bestimmtem Zeitpunkt können als Webseite und als Textdatei per Download abgerufen werden. Auf der Kommandozeile können sie als normale Textdateien abgerufen werden.

Auf dem Repository können folgende Operationen ausgeführt werden:

- Die Unterschiede zwischen zwei beliebigen Zuständen einzelner Geräte können als „Diff" (für „Differences") ermittelt werden
- Die Unterschiede zwischen zwei beliebigen Zuständen einer ganzen Gruppe können als „Change Sets", einer Aneinanderreihung von „Diffs" ermittelt werden
- Welche Änderungen in welcher Revision an einem Gerät vorgenommen wurden, können mittels einer Darstellung mit „Annotations" angezeigt werden. Dort sieht man welche Zeilen zu welchem Zeitpunkt zustande gekommen sind.
- Welche Geräte verändert wurden, hinzugekommen sind oder entfernt wurden kann in einer Revisionshistorie („Log") eingesehen werden. Hier können von den Nutzern Kommentare eingegeben werden, was geändert wurde und warum.
- Ein Abzug des gesamten Repositories („Dump") im Subversion Format ist für Backupzwecke möglich
- Eine Extraktion aller Konfigurationsdateien aus dem Repository („Checkout"), ist lokal und über das Netzwerk möglich

Ausserdem kann mit einer Vielzahl von anderen Subversion Clients[17] auf das Repository zugegriffen werden.

5.6.3 Überwachung von Änderungen

Um zu erfahren, ob Änderungen durchgeführt wurden, will man nicht regelmäßig in das Repository schauen müssen. Eine automatische Benachrichtigung ist also sinnvoll. Dazu muss die zyklische Ausführung des Abrufs mittels CRON eingerichtet sein.

Sobald sich Änderungen an der Konfiguration, der Software, Hardware oder Zuständen wie den VLANs ergeben, wird man optional auf zwei Arten benachrichtigt. Mit einem Web Browser kann man RSS und ATOM Feeds pro Repository oder Gerätegruppe abonnieren. Diese können dann zum Beispiel über „Live Bookmarks" abgerufen werden. Man kann für jeden Mandanten die Adresse eines E-Mail Verteilers angeben. Darüber werden die Änderungen als „Diffs" nach jedem Abruf an alle Empfänger des Verteilers versendet.

5.6.4 Abruf und Archivierung von Software Images oder Dateien

Zum Zustand eines Gerätes gehört nicht nur die Konfiguration, sondern auch die installierte Software. Daher wurde eine Erweiterung programmiert, die es erlaubt von einzelnen Geräten oder Gerätegruppen die Software oder Dateien von den eingebauten Speichermedien zu sichern.

[17] http://subversion.tigris.org/links.html#clients

Der Abruf von Software Images ist derzeit nur für Cisco Geräte mit monolithischem IOS implementiert. Für UNIX Systeme ist eine allgemeine Sicherung mit „tar", „cpio" und beliebigen Host-Skripten möglich. Eine Anpassung auf weitere Geräte ist ohne weiteres möglich. Die vorhandenen Skripte können dazu als Basis dienen und sind leicht in die Struktur zu integrieren.

Die Sicherungen werden aber nicht in das Subversion Repository eingestellt, da eine Versionierung von grossen Binärdateien sinnlos ist und dieselben Images vielfach gespeichert würden. Stattdessen wird bei Cisco IOS mit dem „copy" Befehl und bei UNIX mit cURL [CURL] ein generischer Mechanismus realisiert. So kann mittels einer Vielzahl von Protokollen gemäß einem je Gerät konfigurierbaren URL auf geeignete Server gesichert werden.

Der Ablauf für Cisco IOS ist :

1. Das Programm meldet sich per Telnet oder SSH auf dem Gerät an
2. Es ermittelt, welchen Pfad das laufende Image im Dateisystem (z.B. Flash) hat
3. Es legt die Image Datei per Cisco IOS [IOS] „copy" Befehl vom Gerät aus auf einem Server ab

Auf der Kommandozeile kann außer dem gerade beschriebenen Weg auch der Transfer beliebiger Dateien angestoßen werden, die auf einem Gerät existieren. Die notwendigen Pfade müssen dann schon bekannt sein.

Es gibt folgende Einschränkungen:

Vom Netzwerk z.B. per TFTP geladene Images können nicht gesichert werden, da die zugehörigen Dateien nicht auf dem Gerät vorliegen. Die unterstützen Protokolle zum Sichern werden durch die genutzte Cisco IOS Version vorgegeben. Unter anderem sind dies TFTP, FTP, RCP, SCP, HTTP, HTTPS. Bei Sichern über TFTP werden protokollbedingt maximal 32MB große Dateien unterstützt.

Für UNIX Systeme wurde ein etwas anderer Weg gewählt. Die Methode, deren Argumente und der Storage URL können je Gerät separat definiert werden.

- Per Aufruf von „tar" auf dem Zielsystem, die Listen der zu sichernden und die der nicht zu sichernden („exclude") Dateien sind in Dateien auf dem Zielsystem abgelegt. Die Namen werden als Argumente übergeben.
- Per Aufruf von „cpio", die Liste der zu sichernden Dateien ist auf dem Zielsystem abgelegt. Der Name wird als Argument übergeben.
- Per Aufruf eines beliebigen Skripts auf dem Zielsystem, alle Argumente werden übergeben.

Nach der jeweiligen Aktion wird das erzeugte Archiv mit cURL auf das pro Gerät einstellbare Storage-URL kopiert. Auf dem Zielsystem muss daher cURL installiert sein.

Es gibt folgende Einschränkung:

Die unterstützen Protokolle zum Sichern werden durch die genutzte cURL Version auf dem Zielserver vorgegeben. In der aktuellen Version sind unter anderem TFTP, FTP, SFTP, FTPS, HTTP, HTTPS, cp, SCP möglich.

5.6.5 Strommanagement

Oft ist es von Vorteil, wenn man einzelne Geräte oder Gruppen von Geräten nur bei Bedarf einschaltet. Dies kann zum Beispiel bei hohem Stromverbrauch, seltener Nutzung oder ungenügender Klimatisierung sinnvoll sein. Sollte die Software eines Geräte abstürzen oder Fehlfunktionen der Hardware auftreten, kann so auch ein Kaltstart erzwungen werden und den Weg in eine möglicherweise weit entfernte Lokation vermeiden.

Zur Schaltung der Stromzufuhr gibt es Geräte von verschiedenen Herstellern. Um Geräte verschiedener Hersteller anzusteuern wurde ein Skript für SNMP basierte PDU-Geräte programmiert. Zum Beispiel ist der APC MasterSwitch [APC] zum Ein- und Ausschalten von Geräten geeignet. Er kann per Telnet, über eine Webschnittstelle oder per SNMP bedient werden. Im folgenden werden solche Geräte als „Powerswitch" bezeichnet.

Das Skript nutzt SNMP und kann aus dem GUI, zeitgesteuert (AT, CRON), aus anderen Skripten oder von der Kommandozeile aus aufgerufen werden.

Für einzelne Geräte oder ganze Gerätegruppen kann so

- Der Status der Stromzufuhr jedes Anschlusses ermittelt werden
- Die für die Anschlüsse vergebenen Namen in der SNMP MIB des Powerswitch angezeigt werden
- Der Strom dauerhaft ein- oder ausgeschaltet werden
- Ein Reboot durch kurzzeitige Stromabschaltung erzwungen werden

Der zugehörige Powerswitch, der Anschluss und die SNMP Parameter werden individuell für jedes Gerät in der RANCID Konfiguration angegeben.

5.6.6 Grafische Oberfläche

Die beschriebenen Funktionen der Archivierung und Versionskontrolle von Konfigurationen können mit der grafischen Oberfläche Insurrection [INSUR] genutzt werden. Dies sind generische Funktionen die mit jedem Subversion Repository möglich sind. Mit den Mitteln von HTML und JavaScript ergibt sich eine dynamische Ansicht des Repository.

Für die Administration der Repositories stehen mehrere Funktionen zur Verfügung. Es gibt eine Verwaltung der Benutzer des GUI, eine Vergabe von Zugangsberechtigungen für die Benutzer und eine Backupfunktion für die Daten in Subversion.

Durch Verlinkungen der Inhalte der Konfigurationen auf die jeweilige Revisionen kann unmittelbar zwischen den Zuständen gesprungen werden. In einer zweispaltigen Darstellung („Annotations") wird gezeigt, aus welcher Revision die jeweiligen Zeilen stammen, wenn man auf die entsprechenden Bereiche klickt gelangt man direkt zu der betreffenden Revisionshistorie in der die Änderungen dokumentiert sind.

Für die Überwachung von Änderungen an den Konfigurationen und Zuständen dienen Email-Verteiler und RSS- und ATOM-Feeds, die über Buttons in den Webseiten aktiviert werden können.

Im Rahmen dieser Arbeit wurden zahlreiche Erweiterungen vorgenommen um die Oberfläche mit den anderen Subsystemen zu integrieren und die Funktionen für die Beta-Plattform zu realisieren. Es können neue Mandanten angelegt werden, die Gerätedaten

aus dem Hardware Repository abgerufen werden und alle Funktionen von RANCID und dem Strommanagement genutzt werden.

Durch farbliche Hinterlegung der Unterschiede bei „Diff" und „ChangeSets" kann sehr schnell erkannt werden, wo Änderungen erfolgt sind.

Diese sind durch ein einzelnes URL adressierbar, sodass die Änderungen von allen Geräten, die in einer Revision vorgenommen wurden, in einem Change Request Dokument oder einem Wiki mit einem einzigen Link referenziert werden können.

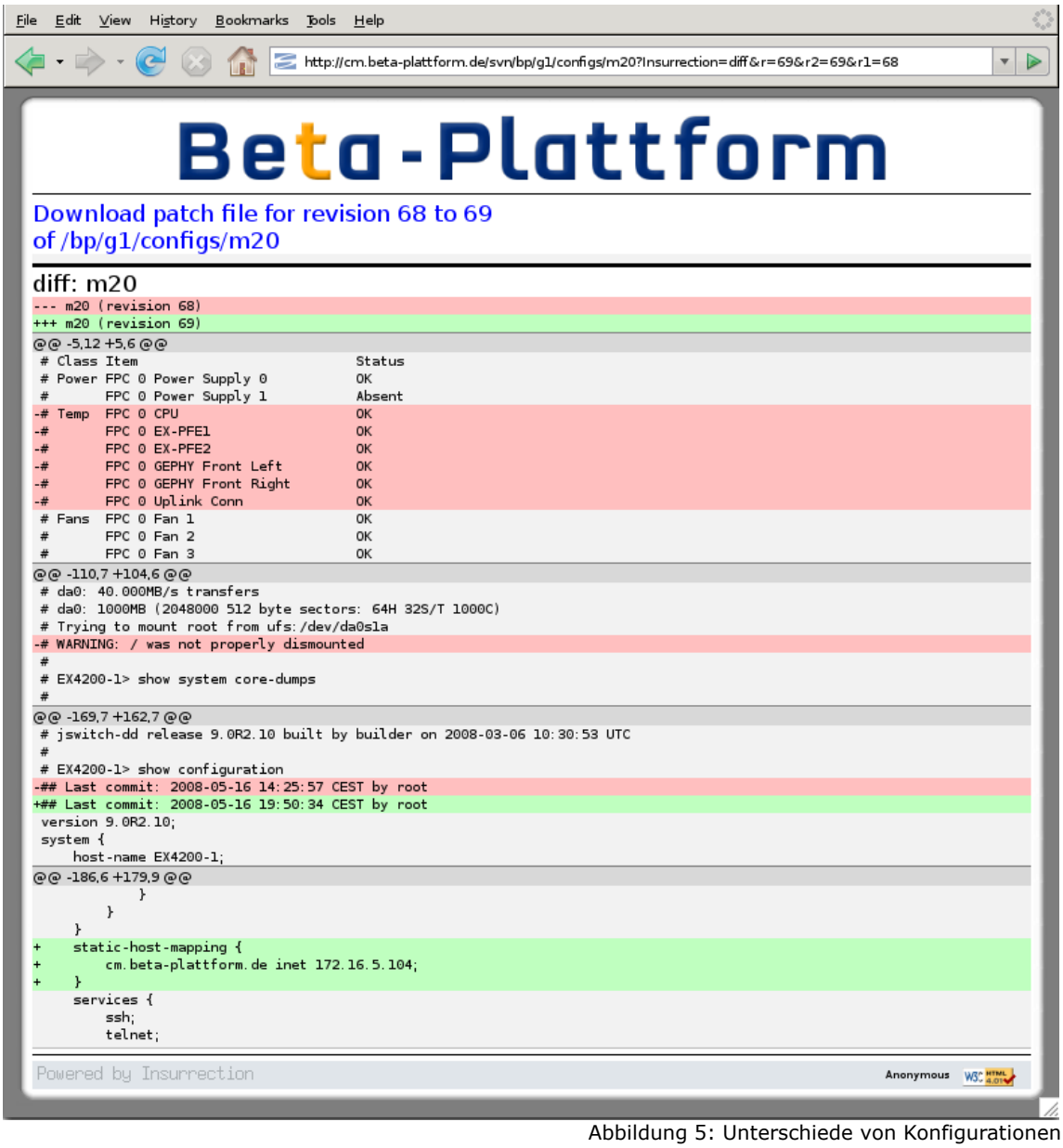


Abbildung 5: Unterschiede von Konfigurationen

Die Änderungen können zur Verarbeitung in anderen Dokumenten oder zur Speicherung in separaten Dateien mit den Funktionen „Patch" und „Patchset" auch als reine Textdaten heruntergeladen werden.

In der Revisionshistorie wird aufgelistet, welche Geräte von einer Revision betroffen waren, hinzugekommen oder weggefallen sind.

Ein Editor zur Pflege der Revisionshistorie ermöglicht die zentrale Dokumentierung von Änderungen durch die Benutzer.

Insbesondere können hier URL mit Links zu anderen Dokumenten wie Change Requests, Ticket Systemen und Wiki Einträgen vermerkt werden.

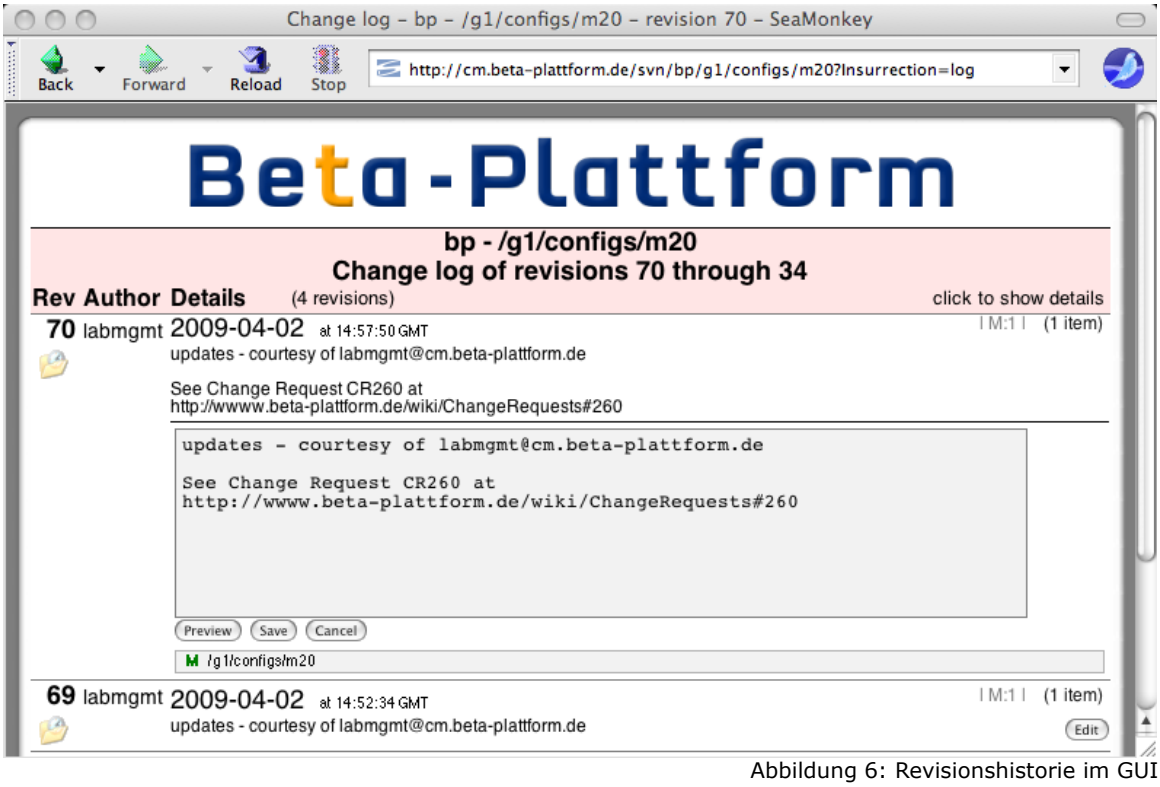


Abbildung 6: Revisionshistorie im GUI

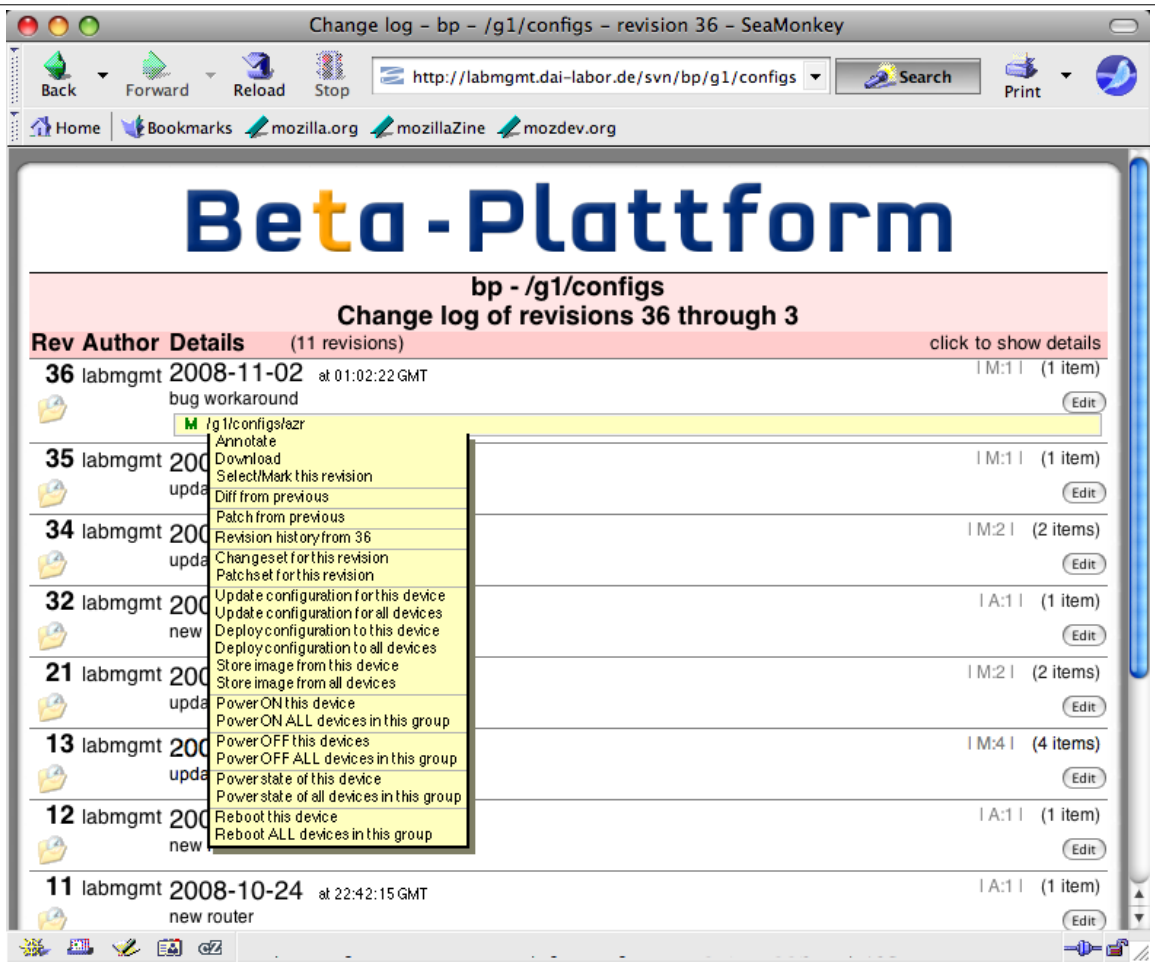
Die Revisionshistorie einer Gruppe oder eines einzelnen Gerätes zu einer bestimmten Revision können durch ein einziges URL adressiert werden. So kann in Change Request Dokumenten gezielt auf den relevanten Teil der Historie referenziert werden.

Erweiterungen

Das GUI wurde um mehrere Funktionen erweitert. Ein zusätzlicher Reiter auf der Startseite enthält eine Kurzhilfe für die Nutzung des GUI und der Kommandozeile und die wichtigsten Dateien.

Die Einrichtung neuer Repositories wurde um neue Parameter für die Beta-Plattform erweitert. So wurde der Abruf der Gerätedaten aus dem Beta-Plattform Hardware Repository implementiert, es kann der Betriebsmodus pro Mandant als standalone und zentral eingestellt werden und die Pfade der CM Repositories sind einstellbar.

Es wurde eine Webschnittstelle zu den Funktionen von RANCID und dem Strommanagement programmiert. Das Kontext-Menü der Revisionshistorie wurde erweitert, um die neuen Funktionen im GUI auszulösen zu können.



Hier kann der Abruf der Konfigurationen und deren Archivierung einzelner Geräte oder von ganzen Gerätegruppen direkt angestossen werden. Ebenso kann die Sicherung von Software Images bzw. ausgewählter Dateien einzelner Geräte oder von Gerätegruppen angestossen werden.

Es kann eine ausgewählte Konfiguration aus dem Repository auf das jeweilige Gerät oder alle Konfigurationen einer Gerätegruppe auf die jeweiligen Geräte übertragen werden (Deploy, Rollback, Rollout)

Hier können auch die Funktionen des Strommanagements benutzt werden.

Für die Verwaltung der Geräte und Gerätegruppen wurde ein eigener Reiter implementiert.

Hier wird eingestellt, welches Gerät in welcher Gruppe aktiv ist. Jedes Gerät kann gleichzeitig in mehreren Gruppen aktiv sein, um verschiedene Szenarien darstellen zu können oder überlappende Nutzung zu modellieren. Der Abruf der Konfigurationen und die Sicherung von Images und Dateien ist hier ebenfalls möglich. Die Funktionen sind aber nur dann möglich, wenn das Gerät vorher für die Gruppe aktiviert wurde. Ist eine Funktion durch den Betreiber nicht erlaubt, oder wird vom Gerät nicht unterstützt, wird „n/a" für „not available" angezeigt.

Für jede Gerätegruppe wird eine eigene Box dargestellt, hier ist exemplarisch nur eine zu sehen:

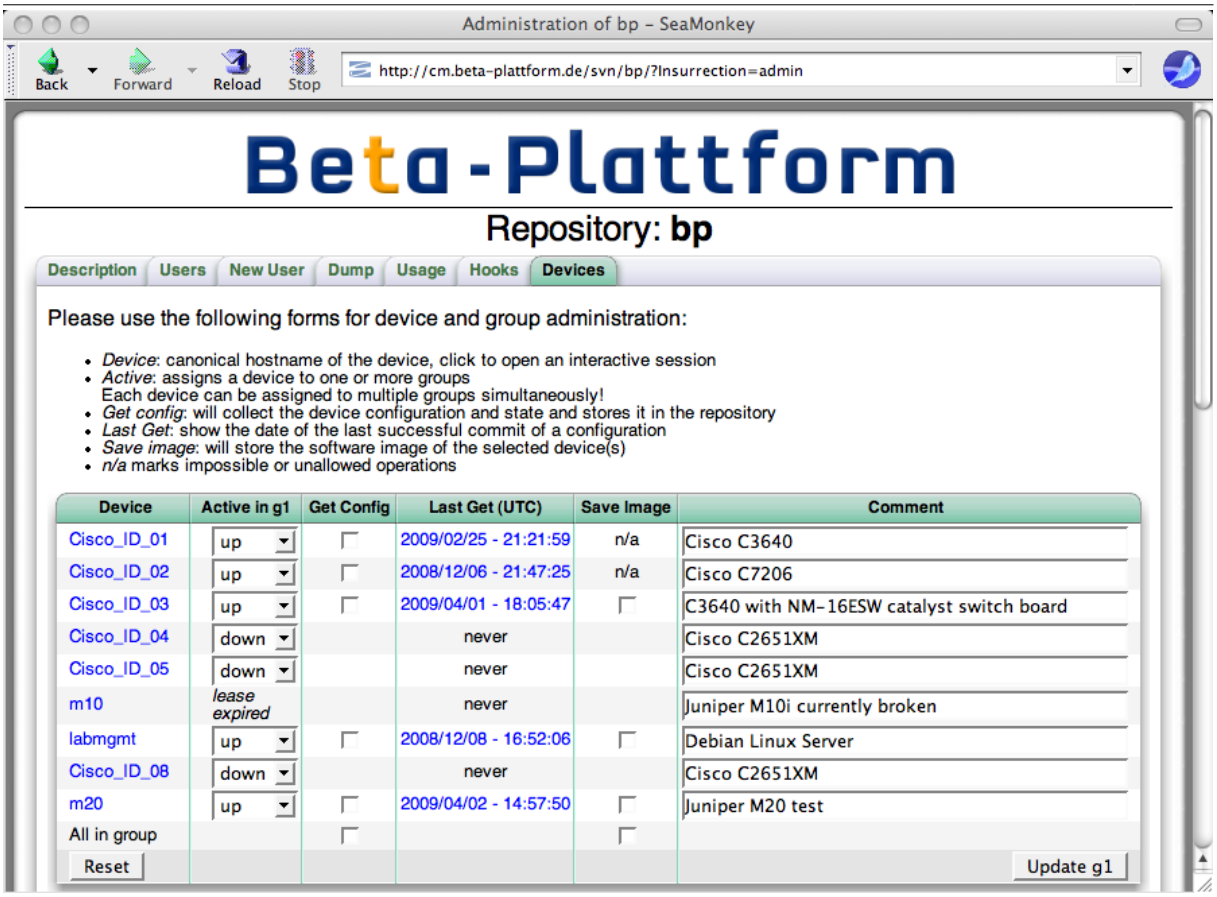


Abbildung 7: Geräteverwaltung im GUI

Zu den Geräten können Kommentare eingegeben werden. Dies geht auch, wenn sie nicht aktiv sind, z.B. um Hinweise auf die Nutzung in anderen Gruppen machen zu können.

Durch Klicken auf den Namen des Gerätes kann eine interaktive Session mit Telnet oder SSH gestartet werden. Die Anzeige der zuletzt abgerufenen Konfigurationen ist durch Klicken auf den Zeitstempel in der Spalte „Last Get" möglich. Wurde die Konfiguration noch nie abgerufen wird dies durch „never" angezeigt.

Geräte deren Buchungen abgelaufen sind werden nicht aus den Repositories entfernt, um die langfristige Archivierung zu gewährleisten. Sie können lediglich in den Gerätegruppen nicht mehr aktiviert werden. Stattdessen wird „lease expired" angezeigt. Wird die Buchung erneuert, können nach einem Update der Gerätedaten die Geräte wieder aktiviert werden.

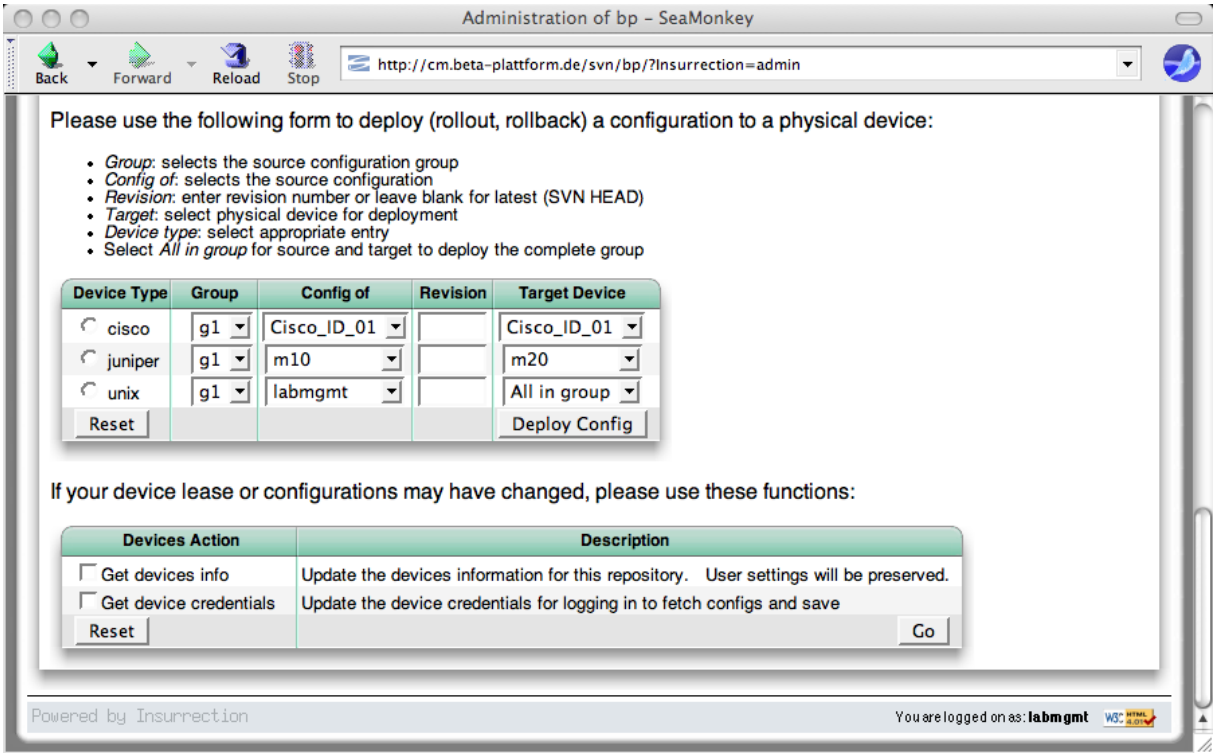


Abbildung 8: Deploy- und Update-Funktion im GUI

Die Deploy-Funktion ist auf dieser Seite flexibler als im Kontextmenü, die Konfigurationen können aus verschiedenen Gruppen und Geräten auf beliebige Geräte übertragen werden. Hier kann auch die gewünschte Revision direkt angegeben werden.

Die letzte Box auf der Seite dient dem erneuten Abrufen der Gerätedaten aus dem Beta-Plattform Hardware Repository, falls sich für den Benutzer etwas an dessen Buchungen geändert hat.

5.6.7 Virtuelles Cisco Labor

Zum Testen und zur potentiellen Erweiterung jedes Labors wurden zusätzlich die Softwarepakete DynaMIPS [DYNAMIPS] und DynaGEN [DYNAGEN] installiert.

DynaMIPS kann bestimmte Cisco Router emulieren und darauf original Cisco IOS Software Images ablaufen lassen. Mit einem Hypervisor-Modus können recht effektiv mehrere Router parallel emuliert und zu virtuellen Netzen verbunden werden. Mittels einer Bridging Funktion können die virtuellen Router über die Netzwerkkarte des Rechners mit dem Host Betriebssystem, dem LAN und den dort angeschlossenen Geräten verbunden werden, um ein gemeinsames Netz zu bilden.

Das Programm emuliert aktuell verschiedene Modelle der Cisco Router Serien 7200, 3600, 3700, und 2600. In diese können nach Bedarf emulierte Module für ATM, POS, Ethernet, Serielle, E1, und Catalyst Ethernet Switching konfiguriert werden. Diese Module können zum Verbinden der virtuellen Router untereinander genutzt werden. Dazu werden virtuelle Ethernet Switches, Frame Relay Switches, ATM Switches oder Back-to-Back Kabel-Verbindungen („Null Modem“) eingesetzt. Die Emulation benötigt viel Rechenleistung, und soll bis zu 1.000 Pakete pro Sekunde erreichen.

DynaGEN ergänzt DynaMIPS und ermöglicht eine komfortable Verwaltung der virtuellen Router in zusammenhängenden Szenarien, sodass ein komplettes virtuelles Netz mit

einem Befehl aktiviert werden kann. So konnte ein virtuelles Testbed realisiert werden, mit dem die meisten Funktionen dieser Arbeit getestet werden konnten.

5.7 Integration der Subsysteme

5.7.1 Das Betriebssystem

Die Arbeit wurde unter dem UNIX-Dialekt „Debian Linux" implementiert. Prinzipiell wären die meisten UNIX-Dialekte geeignet.

Installation

Es wurde eine virtuelle Maschine mit einem Debian Linux vorinstalliert. Die Pakete Apache 2 nebst verschiedenen Modulen, Subversion, SNMP, XSH2, XMLstarlet und libXML sowie eine Vielzahl von Bibliotheken wurden im Rahmen der Arbeit darauf installiert.

Konfiguration

Die Tabelle gibt einen Überblick über die durchgeführten Änderungen an der Systemkonfiguration:

Verzeichnis	***Dateien***	***Nutzung***
/etc	passwd	Benutzer
	group	Gruppenzugehörigkeit
	hosts	Hostnamen und Adressen
	sudoers	Privilegienvergabe
	at.deny	Berechtigung Batchbetrieb
/home/labmgmt/Archives	(Tar Dateien)	Software Archive
/home/labmgmt/src	(verschiedene)	Staging Area

Tabelle 2: Betriebssystem-Konfiguration

- Diese Benutzer wurden mit „useradd" in „/etc/passwd" angelegt
    - „labmgmt" ist der Hauptbenutzer für RANCID, Subversion und das Strommanagement. Das Verzeichnis ist „/home/labmgmt".
    - „www-data" ist der Benutzer, unter dem der Apache Web Server und Insurrection läuft
    - „images" wird zum Zugriff auf den Storage Server verwendet. Das Verzeichnis ist „/home/images".
- „/etc/group" wird für den gemeinsamen Dateizugriff in „/home/labmgmt" um diese Zuordnung ergänzt:

  ```
  www-data:x:33:labmgmt
  ```

- „/etc/sudoers" benötigt einige Einträge, damit die Apache Prozesse von Insurrection die Skripte von RANCID unter dem Benutzer „labmgmt" ausführen können

```
# User alias specification
Runas_Alias LABUSER = labmgmt

# Cmnd alias specification
Cmnd_Alias  LABCMDS = /usr/local/bin/rancid-run,\
                      /usr/local/bin/rancid-cvs,\
                      /usr/local/bin/rancid-creds,\
                      /usr/local/bin/rancid-deploy,\
                      /usr/local/bin/rancid-devices,\
                      /usr/local/bin/rancid-mkconf,\
                      /usr/bin/svn,\
                      /usr/sbin/svnadmin
# User privilege specification
root  ALL=(ALL) ALL
Defaults:labmgmt always_set_home
www-data ALL = (LABUSER) NOPASSWD: LABCMDS
```

- „/etc/at.deny" darf den Benutzer „www-data" nicht enthalten, damit dieser mit CRON und dem Befehl „batch" zeitversetzt Programme starten kann. Dies wird für die CGI Schnittstelle von Insurrection zu RANCID und den zyklischen Abruf genutzt.

Zu den Routern im LAN, von denen Konfigurationen abgerufen werden sollen, müssen entsprechende Routen eingetragen werden, sofern diese nicht über den Default Gateway erreichbar sind.

5.7.2 Die Middleware „Apache"

Apache [Apache] ist ein HTTP Server, der aus Dateien gelesene oder von Programmen generierte Daten als Webseiten zum Browser der Benutzer überträgt.

Installation und Konfiguration

Apache 2 und die Module SSL, DAV, SVN und Deflate wurden mit APT installiert. Die folgenden Verzeichnisse und Dateien sind relevant:

Verzeichnis	***Datei***	***Nutzung***
/etc/apache2	ports.conf	Liste der aktiven Ports
	apache2.conf	Globale Einstellungen
/etc/apache2/mods-enabled	*.conf, *.load	Aktive Module
/etc/apache2/sites-enabled	labmgmt	Virtueller Web Server
	labmgmt-443	Virtueller SSL Web Server
~labmgmt/authentication/ssl	server.crt, server.key	Schlüssel, Zertifikat für SSL

Tabelle 3: Middleware-Konfiguration

Die folgenden Einstellungen wurden vorgenommen:

- „apache2.conf" muss definieren, unter welcher User-ID und Group-ID die Apache Insurrection Prozesse laufen sollen:

  ```
  User www-data
  Group www-data
  ```

- „ports.conf" definiert auf welchen TCP Ports die Web Server antworten sollen. Port 80 ist für HTTP, Port 443 ist für HTTPS mit SSL:

```
Listen 80
Listen 443
```

- „mods-enabled" ist ein Verzeichnis, in dem symbolische Links zu den Apache Modul-Dateien im Verzeichnis „mods-available" angelegt werden müssen, die von der Installation verwendet werden sollen. Es werden die Module CGI, DAV, DAV-FS, DAV-SVN, Deflate, Rewrite, SSL und Proxy verwendet.

  In „proxy.load" müssen zusätzliche Module geladen werden:

```
LoadModule cache_module /usr/lib/apache2/modules/mod_cache.so
LoadModule disk_cache_module
/usr/lib/apache2/modules/mod_disk_cache.so
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_http_module
/usr/lib/apache2/modules/mod_proxy_http.so
```

5.7.3 Abruf der Konfigurationen mit „RANCID"

RANCID ist das Tool, mit dem die Konfigurationen von den Geräten abgerufen, konsolidiert und in einem System zur Revisionskontrolle gespeichert werden.

Installation und Konfiguration

RANCID wurde von [RANCID] in der Version 2.3.2a7 heruntergeladen, mit dem Skript „configure" für die Installation in „/usr/local" vorbereitet und mit „make install" dort installiert.

In der folgenden Tabelle sind die wichtigsten Dateien für das Beispiel-Repository „bp" dargestellt:

Verzeichnis	***Datei***	***Nutzung***
~labmgmt/rancid	_template.conf.xml	Template für globale Einstellungen
~labmgmt/rancid/bp/g1	devices.xml	Geräteliste der Gruppe g1
~labmgmt/rancid/bp/configs	(Namen der Geräte)	Konfigurationen
~labmgmt/rancid/bp/g2	devices.xml	Geräteliste der Gruppe g2
~labmgmt/rancid/bp/configs	(Namen der Geräte)	Konfigurationen
~labmgmt/rancid	bp.conf.xml	Globale Einstellungen für „bp"
~labmgmt/rancid/bp	.cloginrc	Geräte Usernamen/Passwörter

Tabelle 4: RANCID Konfiguration

Die globalen Parameter werden in „`_template.conf.xml`" voreingestellt und bei der Einrichtung des Repositories in „`bp.conf.xml`" übertragen und durch die GUI Parameter ergänzt/ersetzt.

Die globale Konfigurationsdatei in „`/usr/local/etc/rancid.conf`" wird nicht benutzt.

Beispiel für eine Repository Konfiguration im „.conf.xml“ Format:

```
<rancidConf id="bp">
  <commands>
    <command>umask 022</command>
  </commands>
  <environment>
    <variable name="MODE">standalone</variable>
    <variable name="BASEDIR">/home/labmgmt/rancid/bp</variable>
    <variable name="CVSROOT">file:///home/labmgmt/repositories/bp</variable>
    <variable name="LOGDIR">/home/labmgmt/rancid/bp/logs</variable>
    <variable name="LIST_OF_GROUPS">"g1 g2"</variable>
    <variable name="DEVICES_URL">http://localhost/BP/bp/devices.xml</variable>
    <variable name="IMAGE_URL">ftp://images:XXX@cm.beta-plattform.de/</variable>
    <variable name="TERM">network</variable>
    <variable name="PATH">/usr/local/bin:/usr/bin:/usr/local/bin:/usr/sbin:/bin</variable>
    <variable name="TMPDIR">/tmp</variable>
    <variable name="RCSSYS">svn</variable>
    <variable name="FILTER_PWDS">NO</variable>
    <variable name="NOCOMMSTR">NO</variable>
    <variable name="MAX_ROUNDS">1</variable>
    <variable name="OLDTIME">4</variable>
    <variable name="PAR_COUNT">5</variable>
    <variable name="DEBUG">on</variable>
  </environment>
</rancidConf>
```

Die wichtigsten Variablen in „bp.conf.xml“ haben folgende Einstellungen erhalten:

- „BASEDIR“ gibt das Arbeitsverzeichnis für RANCID an:

  `/home/labmgmt/rancid/bp`

- „CVSDIR“ heißt aus historischen Gründen so, bezeichnet aber hier den Pfad zum Subversion Repository für RANCID:

  `/home/labmgmt/repositories/bp`

- „LIST_OF_GROUPS“ (mit Anführungszeichen) gibt die Gerätegruppen an:

  „`g1 g2`“

- „LOGDIR“ gibt das Verzeichnis der Protokolldateien an:

  `/home/labmgmt/rancid/bp/logs`

Erweiterungen

Die Konfigurationsdateien wurden auf XML umgestellt und für das GUI und die Zwecke der Beta-Plattform erweitert.

Es wurden folgende Erweiterungen der globalen Variablen in „.conf.xml“ (ehemals „rancid.conf“) notwendig:

- „MODE“ gibt an, ob das Repository über die Beta-Plattform konfiguriert wird („`BP`“) oder ob es eine lokale Instanz für das Testbed sein soll („`standalone`“).

- „DEVICES_URL“ gibt an, wo die XML Datei „devices.xml“ mit den Informationen über die Geräte bezogen werden kann. Das Format ist im Kapitel „Nutzung mit der Kommandozeile“ beschrieben, die formale Definition ist als „devices.xsd“ im Anhang enthalten.

- „IMAGE_URL“ gibt an wie und wo die Software Images von den Geräten gespeichert werden sollen, sofern für ein Gerät kein spezifisches URL in „devices.xml“ angegeben ist (s.u.). Der jeweilige Gruppenname wird automatisch an den URL angehängt. Es müssen daher auf dem Server Unterverzeichnisse mit den Namen der Gruppen vorhanden sein. Hier sind das „g1“ und „g2“.

  Das Format ist ein URL, z.B.:

  `ftp://username:password@ftp.server.de/bp`

- „DEBUG“ kann zur detaillierten Protokollierung auf „on“ oder „off“ gesetzt werden. Die Protokolldateien sind unter „/home/labmgmt/rancid/bp/logs“ zu finden.

Die folgenden Dateien und Skripte wurden neu erstellt oder geändert:

- Das neue Skript „urancid“ sichert Teile der UNIX Konfiguration und Zustände.

- Das neue Skript „rancid-image“ für Cisco IOS meldet sich auf den Geräten an und kopiert das laufende Software Image auf den per URL angegebenen Server.

- „ccopy“ ist ein Expect Skript, dass von „rancid-image“ zum Kopieren der Dateien auf den Cisco Geräten benutzt wird.

- Die neuen Skripte „urancid-image" und „ulogin" meldet sich auf den UNIX Geräten an und rufen „tar", „cpio" oder ein Skript auf und kopieren das so erzeugte Archiv auf den per URL angegebenen Server.

- Zum Übertragen von Konfigurationen aus dem Repository auf Cisco Geräte dienen die Skripte „rancid-deploy" und „cdeploy". Wegen eines Bugs in IOS (nicht konform zu RFC2616[18]) musste ein Workaround in der Konfiguration implementiert werden, siehe dazu „Nutzung auf der Kommandozeile". Ausserdem wird das Feature in Cisco IOS noch weiterentwickelt und ist daher nur eingeschränkt nutzbar. Z.B. bleibt in älteren Releases die Reihenfolge von Route-Maps nicht erhalten. Es sollten daher die entsprechenden Release Notes für das Image auf dem Gerät beachtet werden.

- Zum Übertragen von Konfigurationen aus dem Repository auf Juniper Geräte dienen die Skripte „jrancid-deploy" und „jdeploy".

- „rancid-run", „rancid-fe" und „control_rancid" wurden erweitert, um den Aufruf von „rancid-image" und den o.g. Scripten für die Deploy Funktion zu ermöglichen. Dazu wurden die Optionen „-d" und „-i" implementiert und die Möglichkeit geschaffen, später auch Geräte anderer Hersteller zu unterstützen.

- Zum Holen und Verarbeiten der XML Konfigurationsdateien wurden die Skripte „rancid-mkconf", „rancid-devices", „rancid-creds" und „dev-merge.xsh" erstellt.

- „devices.xml" (früher „router.db") hat viele zusätzliche Felder für die Verwaltung der Geräte und das Strommanagement etc. erhalten. Hier können z.B. zu jedem Gerät der zugehörige Hostname des Powerswitch, die SNMP Parameter und der Stromanschluss angegeben werden. Das Format ist im Anhang beschrieben.

- Beim automatischen Login auf die Geräte kann jetzt auch bei bei SSH der zu benutzende Port in „.cloginrc" angegeben werden. Dies war vorher nur bei Telnet möglich. Dazu wurden alle betroffenen login-Skripte geändert.

- Bei der Nutzung von Geräten der Beta-Plattform wird vor dem automatischen Login geprüft, ob der Zugriff noch erlaubt ist oder die Nutzungsdauer vorüber ist.

5.7.4 Archivierung und Versionskontrolle mit „Subversion"

Subversion ist ein System zur Revisionskontrolle. Es wird insbesondere in der Softwareentwicklung eingesetzt.

Installation und Konfiguration

Subversion wurde mit APT installiert. Eine allgemeine Konfiguration ist nicht notwendig.

5.7.5 Strommanagement mit SNMP

Installation

Im Rahmen der Installation des Betriebssystems wurde das SNMP Package mit APT installiert.

[18]http://blog.ioshints.info/2006/12/cisco-ios-violates-rfc-2616-http11.html

Programmierung

Es wurde das Shell Skript „/usr/local/bin/apc-power" erstellt. Dieses kann von der Kommandozeile, aus anderen Programmen und aus dem GUI aufgerufen werden.

Das Skript arbeitet generisch mit SNMP Aufrufen und ist nicht auf APC-Geräte beschränkt.

Konfiguration

Zum Test standen nur Geräte von APC zur Verfügung. Um diese anzusteuern, wurde die MIB Definitionsdatei „PowerNet-MIB.txt" von [APC] heruntergeladen und im Verzeichnis „/usr/share/snmp/mibs/" abgelegt. Sollen andere Geräte benutzt werden, so müssen die entsprechenden MIB Dateien dort abgelegt werden.

Aus der Datei „bp.conf.xml" für das Repository wird die globale Variable „BASEDIR" ausgelesen, um den Pfad zur RANCID Konfigurationsdatei „devices.xml" zu ermitteln.

Das Format der Datei „devices.xml" wurde um Elemente für den Hostname des Powerswitch und die SNMP Parameter erweitert, die das Schalten der Anschlüsse ermöglichen. Für jedes Gerät gibt es einen eigenen Satz von Parametern.

Hier ein Beispiel:

```
<deviceControl>
    <hostname>pwrctrl.beta-plattform.de</hostname>
    <outlet>3</outlet>
    <readCommunity>secret</readCommunity>
    <writeCommunity>secret</writeCommunity>
    <powerStateOID>sPDUOutletCtl</powerStateOID>
    <powerCtlOID>sPDUOutletCtl</powerCtlOID>
    <powerOnOP>outletOn</powerOnOP>
    <powerOffOP>outletOff</powerOffOP>
    <rebootOP>outletReboot</rebootOP>
</deviceControl>
```

Siehe dazu das Kapitel „Nutzung mit der Kommandozeile" und den Anhang.

5.7.6 Die grafische Oberfläche „Insurrection"

Insurrection bildet die Webschnittstelle zu Subversion und dessen Repositories, RANCID und dem Strommanagement.

Installation und Konfiguration

Insurrection wurde von [INSUR] heruntergeladen und in „/home/labmgmt" entpackt. Die wesentlichen Bestandteile sind:

Verzeichnis	***Datei***	***Nutzung***
~labmgmt/www	insurrection.pl, admin.pl	Hauptskripte und Konfiguration
	.htaccess	Pfade, Rewrites, Access Control
	*.cgi, *.js, *.js	CGI, JavaScript, XSLT Dateien
	log.js	Historienmenü mit Erweiterung
	rancid.cgi	Schnittstelle zu RANCID
	beta-plattform.template	Hilfstext für das GUI
~labmgmt/repositories	bp/	Repository für „bp“
~labmgmt/authentication	access	Repository- & Zugangskontrolle
	passwords	Nutzerdaten für Web-GUI
~labmgmt/logs	access.log, error.log	Web Server Protokolldateien

Tabelle 5: Konfiguration Insurrection

- „.htaccess“ kontrolliert den Zugriff auf die Programmteile durch die Abbildung der URL über Apache auf die CGI Skripte. Dazu mussten hier zwei Pfade ins Dateisystem angepasst werden.
- „access“ definiert, welche Repositories vom GUI aus genutzt werden dürfen, welche Benutzer darauf zugreifen dürfen und mit welchen Rechten. Diese Datei wird mit dem GUI verwaltet.
- „passwords“ ist eine Apache Datei mit den Namen und Passwörtern der Benutzer des GUI. Diese Datei kann mit „htpasswd2“ oder dem GUI verwaltet werden.
- Das Logo wurde in „/home/labmgmt/www“ als cm.beta-plattform.de.png“ abgelegt. Der Name (ohne die Dateiendung) muss dem URL des virtuellen Web Server entsprechen.

Um die Revisionshistorie ändern zu können, muss für jedes Repository im GUI im Reiter „Hooks“ die entsprechende Auswahlmöglichkeit aktiviert werden.

Erweiterungen

Das Verzeichnis „~labmgmt/www“ enthält die Programmteile von Insurrection. Hier wurden die folgenden Maßnahmen durchgeführt:

- Insurrection wurde auf die Nutzung der neuen XML Dateien von RANCID angepasst.
- Ein neuer Reiter mit einer Hilfsfunktion wurde in „index.cgi“ und „index.template“ eingerichtet. Der Inhalt des Reiters befindet sich in „beta-plattform.template“.
- Die Funktionen zum Anlegen neuer Repositories wurde auf die Beta-Plattform angepasst. Durch zusätzliche Parameter werden vom Hardware Repository die Geräteinformationen und Credentials abgerufen.

Es wurde eine mandantenfähige Schnittstelle zu „RANCID“, Subversion und zum Strommanagement programmiert:

- Es wurde eine Erweiterung des Kontextmenüs der Historien in „log.js“ um die neuen Funktionen durchgeführt.

- Es wurde das Skript „rancid.cgi" programmiert, das den Abruf von Konfigurationen, von Software Images, die Deploy Funktion und die Aktionen im Strommanagement aus dem Kontextmenü der Revisionshistorie ermöglicht.

- In „.htaccess" erfolgt die Abbildung der neuen URL auf das neue CGI Skript

- Für die Repositoryverwaltung wurde ein neuer „Devices" Tab für die Gruppenzuordnung und Verwaltung der Geräte in „admin.cgi" eingerichtet. Auch hier sind die oben genannten RANCID Funktionen ausser dem Strommanagement integriert. Die Deploy Funktion ist hier wesentlich flexibler als im Kontextmenü zu handhaben. Zusätzlich kann der erneute Abruf der Geräteinformationen und Credentials veranlasst werden.

5.7.7 Das virtuelle Cisco Labor mit „DynaMIPS"

DynaMIPS ist ein Emulator, der die Hardware von verschiedenen Cisco Routern so nachbildet, dass das Betriebssystem IOS von Cisco Systems unverändert darauf ablaufen kann. Mehrere emulierte Router können untereinander und mit dem LAN verbunden werden.

DynaGen steuert die Konfiguration des DynaMIPS Hypervisors. Die Konfiguration der Router Hardware und deren Vernetzung untereinander wird in „.net" Dateien angegeben. Ausserdem gibt es ein CLI für die Kontrolle der emulierten Router.

Installation und Konfiguration

DynaMIPS wurde von [DYNAMIPS] als Binärversion heruntergeladen und in „/home/labmgmt/DynaMips" und „/usr/local/bin" abgelegt.

DynaGen besteht aus Python Skripten, wurde von [DYNAGEN] heruntergeladen und in „/home/labmgmt/DynaLab/dynagen-0.9.2" abgelegt. Ein symbolischer Link zu „dynagen" wurde in „/usr/local/bin" angelegt.

Die relevanten Teile sind:

Verzeichnis	***Datei***	***Nutzung***
~labmgmt/DynaLab	dynagenidledb.ini	Daten zu virtuellem CPU Leerlauf
~labmgmt/DynaLab/images	(IOS Images)	Software für virtuelle Router
~labmgmt/DynaLab/vlab	vlab.net	Konfiguration virtuelle Router
	vlab.run	Startskript für Hypervisor
	vlab.routes	Skript für IP Routen
	*flash, *nvram, *sram	virtualisierte Routerteile
/usr/local/etc	dynagen.ini	Konfiguration von DynaGen
/usr/local/bin	get-vgw-ip.pl	IP Adress Sniffer
/etc/init.d	vlab	Boot Skript

Tabelle 6: Konfiguration DynaLab

- In „dynagen.ini" sind nur Einstellungen für Telnet und der Pfad zu „dynagenidledb.ini"

- Die „.net" und „.routes" Dateien werden im Kapitel „Nutzung mit der Kommandozeile" erklärt.

Um die virtuellen Router mit dem Host und dem LAN zu verbinden, müssen folgende Maßnahmen durchgeführt werden

- Es muss ein Bridging Interface zu „eth0" erzeugt werden, z.B. über eine virtuelle Schnittstelle von VMware/XEN oder ein logisches Interface von Linux. Hier wurde der erste Weg gewählt, um „eth1" anzulegen.

- Ein Interface eines Routers muss in der DynaGen Konfiguration auf den Host gebrückt werden, hier über das „eth1" Interface

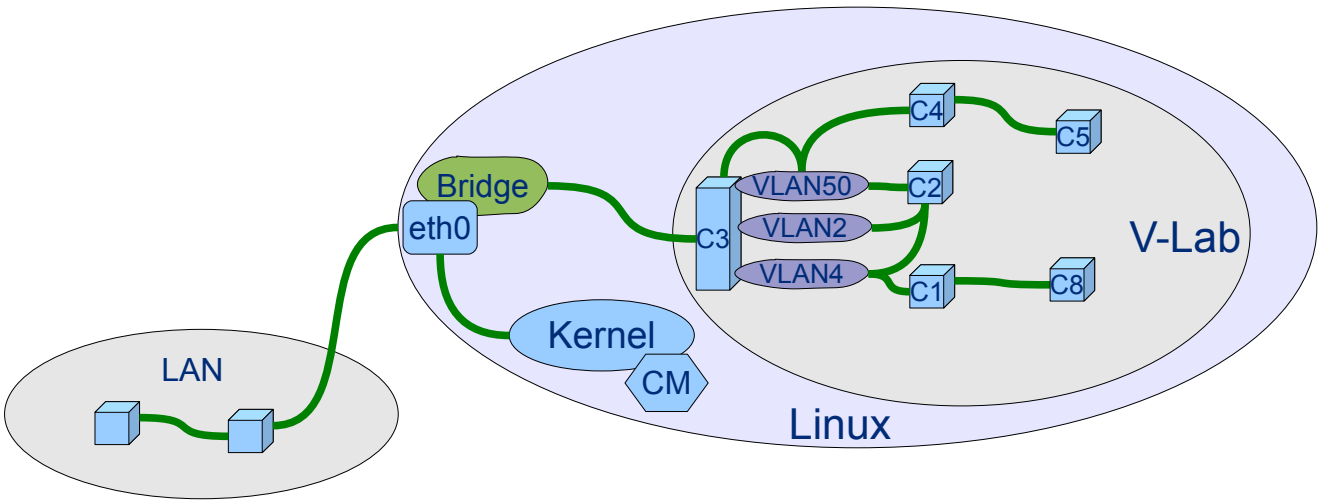
```
[[Router Cisco_ID_03]]
model = 3640
console = 2003
slot1 = NM-1FE-TX
slot2 = NM-16ESW
f0/0 = NIO_linux_eth:eth1
```

Erweiterungen

Beim Systemstart müssen die virtuellen Router gestartet werden und die virtuellen Netze erreichbar gemacht werden. Dazu wurden die Skripte „vlab", „vlab.routes" und „get-vgw.ip.pl" erstellt.

Das Skript „/etc/init.d/vlab" startet Screen Sessions mit DynaMIPS und Dynagen um die virtuellen Router zu starten und ruft das Shell Skript „vlab.routes" auf. Es verbindet den Host und das physische LAN mit den virtuellen Routern und deren Netze durch Linux Routen.

Defaultmässig erhält der virtuelle Gateway Router „Cisco_ID_03" seine Adresse für sein Interface „FastEthernet 0/0" an der Bridge per DHCP. Das Skript „get-vwg-ip.pl" erkennt diese Adresse, in dem es den CDP[19] Verkehr auf Layer 2 mitliest. Mit dieser Zieladresse werden die Linux Routen eingetragen. Auf dem Interface des virtuellen Gateway Router darf daher die CDP Funktion nicht abgeschaltet werden.



C3: C3640 Router mit Catalyst-Modul

Abbildung 9: Netz-Topologie Virtuelle Router

[19]Cisco Discovery Protocol

6 Conclusio und Ausblick

Es ist gelungen, mehrere Open Source Tools der Aufgabenstellung entsprechend zu einem mandantenfähigen, herstellerunabhängigen Configuration Management zu integrieren. Mit dem angepassten GUI ist komfortables Arbeiten auf den Repositories und mit den Geräten in verteilten Testbeds über das Web möglich. Mit dem CLI und in Skripten können komplexe Abläufe automatisiert werden, insbesondere die zyklische Archivierung der Konfigurationen und Zustände der Geräte in versionierten Repositories. Damit kann eine Wiederverwendung von Testbeds implementiert werden. Die Erweiterungen zum Strommanagement und dem Sichern von Images und Dateien der Geräte erleichtern die Arbeit in den Testbeds.

Das Configuration Management kann durch die konsequente Verwendung von URL zur Adressierung von Dateien sowohl unabhängig in jedem Testbed als auch zentral mit dem Hardware Repository der Beta-Plattform für föderierte, verteilte Testbeds eingesetzt werden. Beide Betriebsmodi können für jeden Mandanten separat eingestellt werden, können also in einer Installation koexistieren. Es wurden zwei virtuelle Appliances erstellt, eine zentrale für die Beta-Plattform in einer XEN Instanz beim European Center for Information and Communication Technologies (EICT) und eine generische für den Einsatz in Testbeds in einer VMware Instanz. Mit Hilfe der virtuellen Router ist es auch ohne eigene Hardware möglich auf hohem Niveau Netzwerktechnik zu erlernen und zu erforschen[20], alles in einer einzigen virtuellen Appliance.

Durch die Möglichkeit in der Revisionshistorie URL zu anderen Dokumenten im Web abzulegen und in diesen Dokumenten URL zu bestimmten Revisionen von Konfigurationen, zu Changesets zwischen zwei Revisionen und zu den betreffenden Historien abzulegen, kann die Dokumentation von Changes gut unterstützt werden.

Mit der Umstellung der Konfigurationsdateien der verwendeten Tools auf XML wurde es wesentlich leichter diese zu integrieren und neue Funktionen zu implementieren. Die Tools „XSH“ und „xmlstarlet“ waren besonders hilfreich, um Operationen auf XML Dateien aus Perl- und Shell-Skripten heraus durchzuführen. Anhand der Erweiterungen zum Archivieren von UNIX Konfigurationen und Zuständen sowie dem Sichern von Dateien konnte gezeigt werden, dass die Tools gut für generische Server Betriebsysteme nutzbar gemacht werden können. Anpassungen auf individuelle Dienste und Applikationen auf solchen Server sind mit wenigen Zeilen Skript möglich.

Das RANCID Tool hat sich seit einiger Zeit nicht mehr viel verändert. Die in dieser Arbeit implementierten Änderungen und Erweiterungen sollen an die Maintainer übergeben werden, in der Hoffnung dass sie in eine neue Release einfliessen werden. Dies soll dem Tool eine neue Basis für zukünftige Anwendungen und Erweiterungen geben. In jedem Fall wird die Arbeit im Software Repository der Beta-Plattform zur Verwendung durch Testbeds und andere Projekte bereitgestellt.

Dass es weiterhin Grenzen beim Configuration Management gibt, hat sich bei der Entwicklung der Erweiterung zur Übertragung von Konfigurationen aus den Repositories auf Geräte gezeigt. So haben auch grosse Hersteller von Netzwerktechnik noch Probleme zwischen zwei Konfigurationen umzuschalten. Der Betrieb von heterogenen Netzwerken bleibt ohne herstellerunabhängige Managementtools auch weiterhin eine Herausforderung. Die zunehmende XML-Fähigkeit der Geräte gibt aber eine gute Perspektive für zukünftige, integrierte Lösungen.

[20] In der Arbeit sind nur virtuelle Cisco Router implementiert, es gibt aber auch die Möglichkeit Juniper Router Software in VMware laufen zu lassen, siehe dazu http://www.netemu.cn/juniperSimulation/20080413/536-1.html

7 Anhang A: Configuration Mangement Handbuch

7.1 Die grafische Oberfläche

Das GUI ist mit einem Web Browser unter zwei verschiedenen URL zu erreichen:

- http://cm.beta-plattform.de auf Port 80
- https://cm.beta-plattform.de mit SSL auf Port 443. Hier ist immer eine Anmeldung erforderlich.

7.1.1 Startseite mit der Auswahl der Repositories

Unter beiden genannten URL gelangt man auf die Startseite von Insurrection:

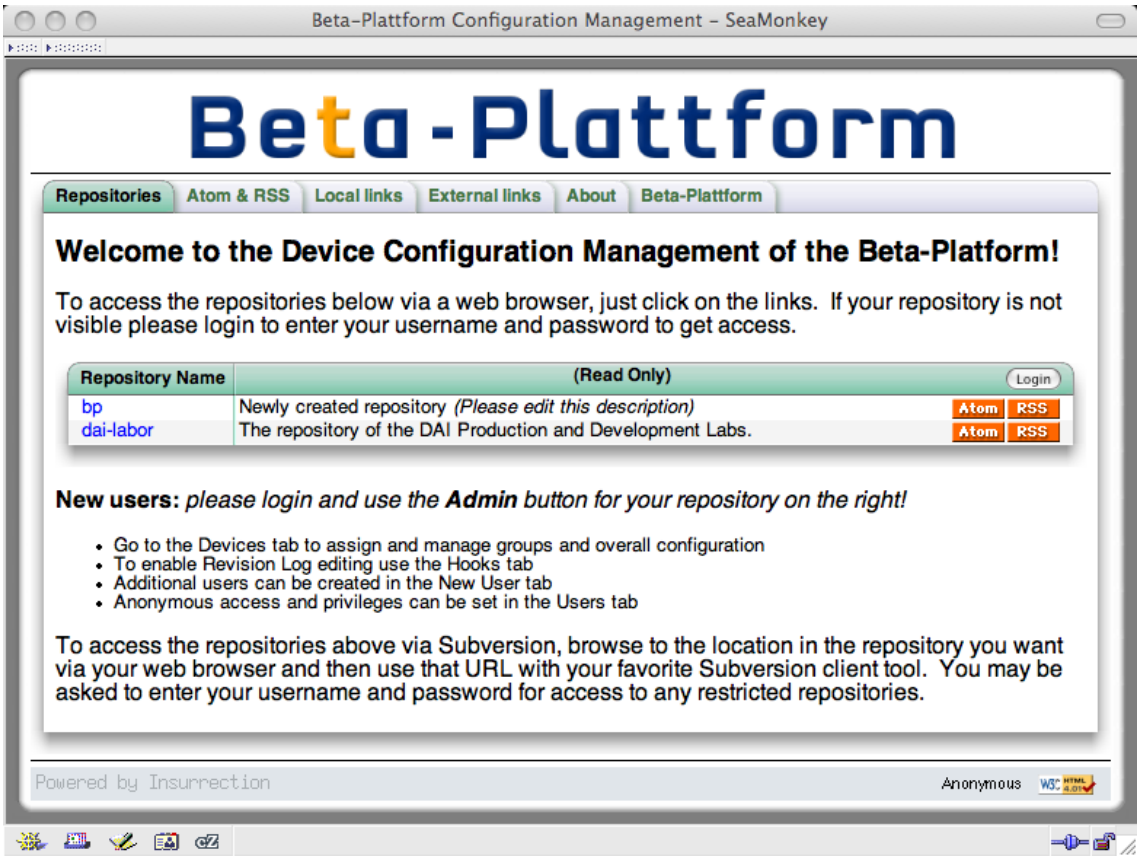


Abbildung 10: GUI Seite Repositories

Zum Repository mit den Konfigurationen gelangt man über den Link „bp".

Will man am GUI oder Repository Änderungen vornehmen oder die Funktionen zur Administration nutzen, muß man sich mit „Login" anmelden und dann die neuen Buttons „Admin" wählen. Der Benutzer muß über die notwendigen Privilegien verfügen.

Die „RSS" und „ATOM" Buttons aktivieren Web Feeds, mit denen man Aktualisierungen im Repository abonnieren kann. Dies kann zum Beispiel über „Live Bookmarks" erfolgen. Die Feeds werden alle zwei Stunden aktualisiert.

Im Reiter „Local Links" sind zwei interessante Links zu finden:

- Testseiten, um zu testen ob ein Browser für Insurrection geeignet ist
- Ein „Quick Start Guide" für Subversion

Im Reiter „Beta-Plattform" befindet sich eine Kurzhilfe für die Benutzer des Testbed. Dort werden die wichtigsten Verzeichnisse, Dateien und Befehle erklärt.

7.1.2 Auswahl der Gerätegruppen

Von der Auswahl des Repository „bp" gelangt man zur Auswahl der Gerätegruppen.

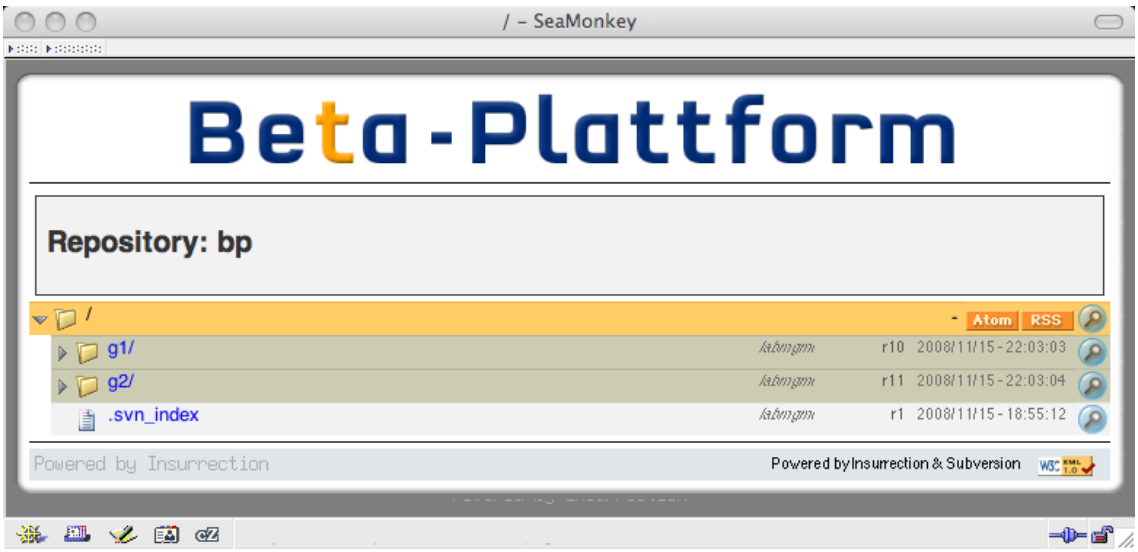


Abbildung 11: GUI Seite Gruppen

Das Repository „bp" enthält hier zwei Gerätegruppen.

- „g1" steht für Gerätegruppe 1
- „g2" steht für Gerätegruppe 2

Für jedes Repository sind die folgenden Elemente zu sehen

- „r10", das Datum und die Uhrzeit dahinter stehen für die Revision 10 der gesamten Gruppe .
- „r11", das Datum und die Uhrzeit dahinter stehen für die Revision 11 der gesamten Gruppe.
- Die kleinen Lupen auf der rechten Seite führen zur Revisionshistorie der jeweiligen Gruppe.

Das Logo führt stets zur Startseite zurück.

7.1.3 Anzeige der Geräteliste

Wählt man eine der Gerätegruppen aus, gelangt man auf die Gruppenseite:

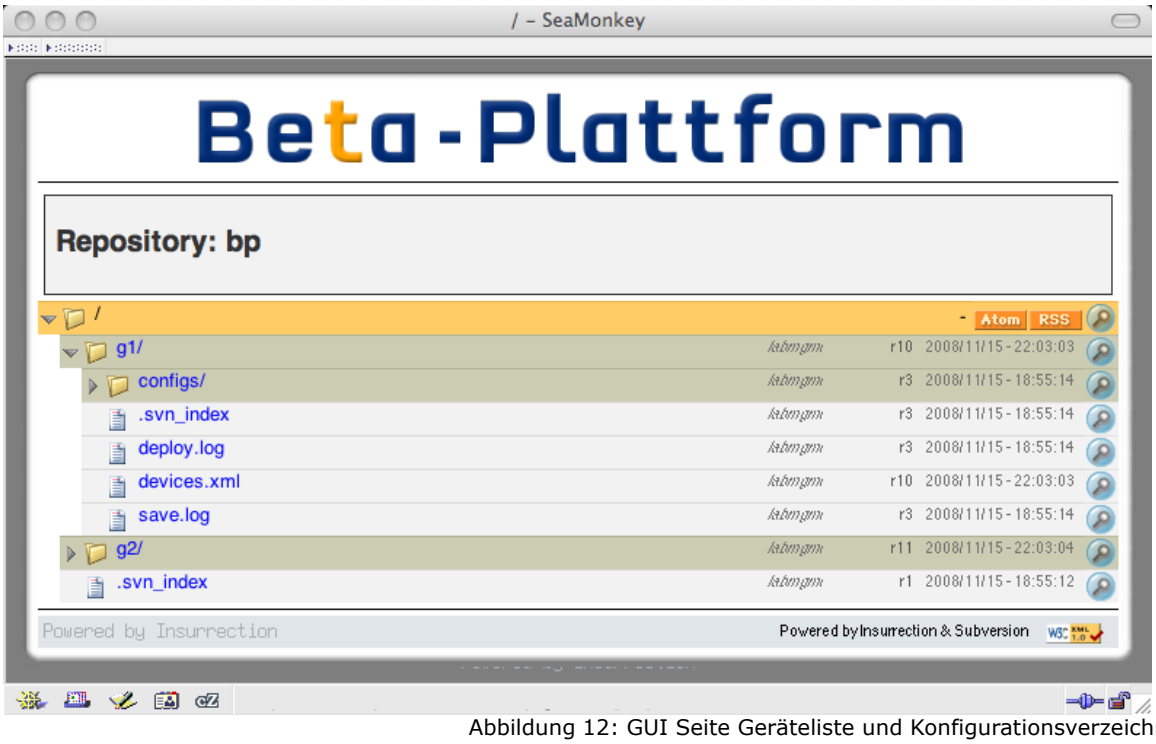


Abbildung 12: GUI Seite Geräteliste und Konfigurationsverzeichnis

Das Format entspricht der vorhergehenden Seite. Die folgenden Funktionen sind möglich:

- „devices.xml" ist die in das Subversion Repository eingebrachte Datei „/home/labmgmt/rancid/bp/g1/devices.xml", in der die Gerätegruppe definiert ist. „r10" ist die Revision der Datei selbst. Wählt man die Datei an, wird sie als Textdatei angezeigt.
- „r3" ist die Revision des Verzeichnis „configs" und damit auch die Revision der Konfigurationen der gesamten Gerätegruppe.
- Die Datei „.svn_index" enthält den HTML Text für den Begrüssungskasten oben auf der Seite. Änderungen werden mit dem Befehl „svn commit" aktiviert.
- Die kleinen Lupen auf der rechten Seite führen zur Revisionshistorie des jeweiligen Verzeichnis oder der jeweiligen Datei
- Die Datei „save.log" enthält die Kurzlogs der Sicherungsvorgänge
- Die Datei „deploy.log" enthält die Kurzlogs der Deploy Funktion
- Besucht man die Order nicht durch Aufklappen sondern durch Doppelklick, kann man in der Hierarchie zurückgehen, indem man den gewünschten Teil des Pfades „<root>/g1" in der Kopfzeile anwählt.

Das Format von „devices.xml" ist im Kapitel „Nutzung mit der Kommandozeile" und im Anhang beschrieben.

7.1.4 Konfigurationen der Geräte

Nach Auswahl des Verzeichnis „configs" gelangt man zu den gespeicherten Konfigurationen aller Geräte der Gruppe.

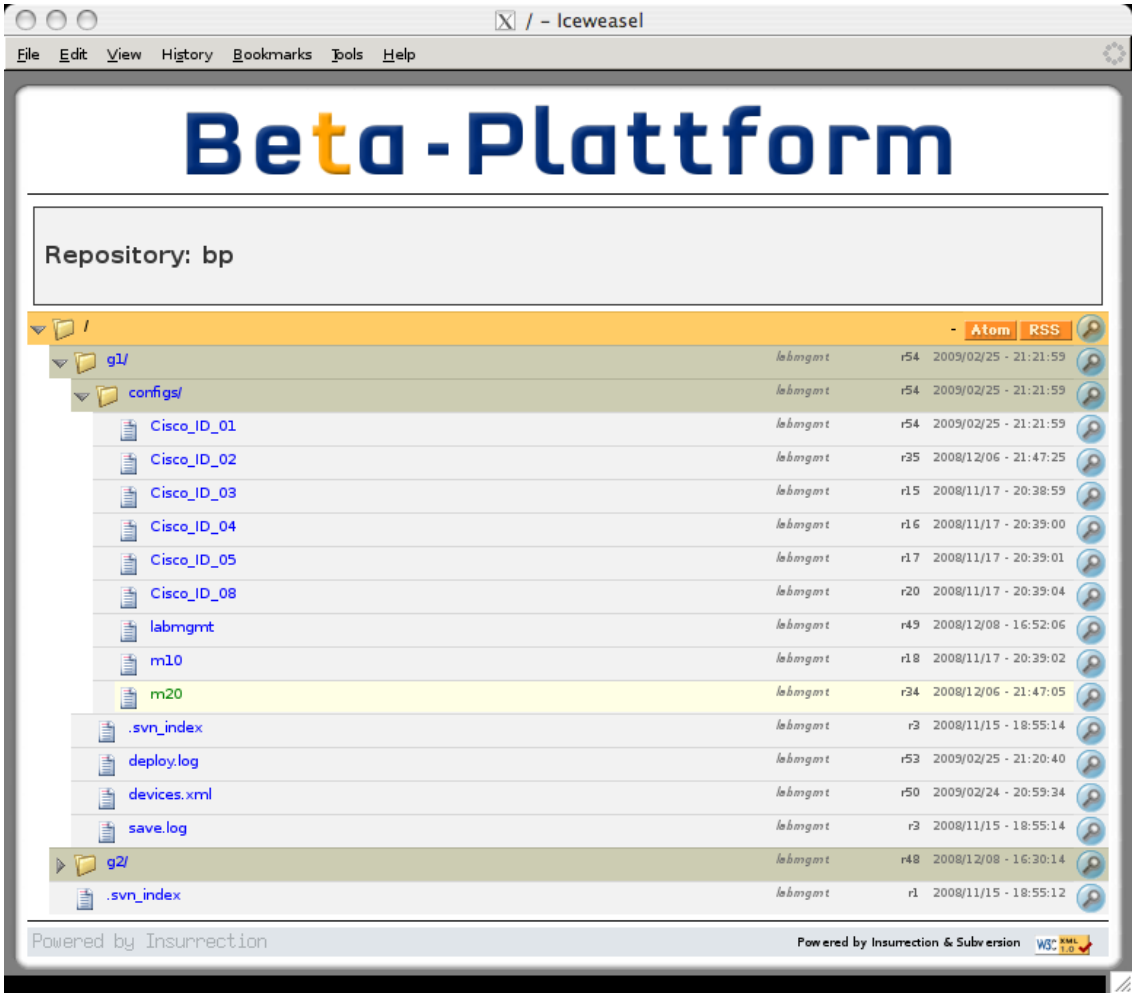


Abbildung 13: GUI Seite Gerätekonfigurationen

Das Format entspricht genau der vorhergehenden Seite.

Die folgenden Funktionen sind möglich:

- Wählt man eines der Geräte aus, so wird die konsolidierte Konfiguration als reiner Text im Browser angezeigt. Ein Beispiel für eine solche Konfiguration ist im Anhang zu finden.
- Die kleinen Lupen auf der rechten Seite führen zur Revisionshistorie des jeweiligen Gerätes.
- Besucht man die Order nicht durch Aufklappen sondern durch Doppelklick, kann man in der Hierarchie zurückgehen, indem man den gewünschten Teil des Pfades „<root>/g1/configs" in der Kopfzeile anwählt.

7.1.5 Revisionshistorie

Wählt man eine der Lupen aus, gelangt man zur Historie der entsprechenden Revision. Hier werden die fortgeschrittenen Funktionen realisiert.

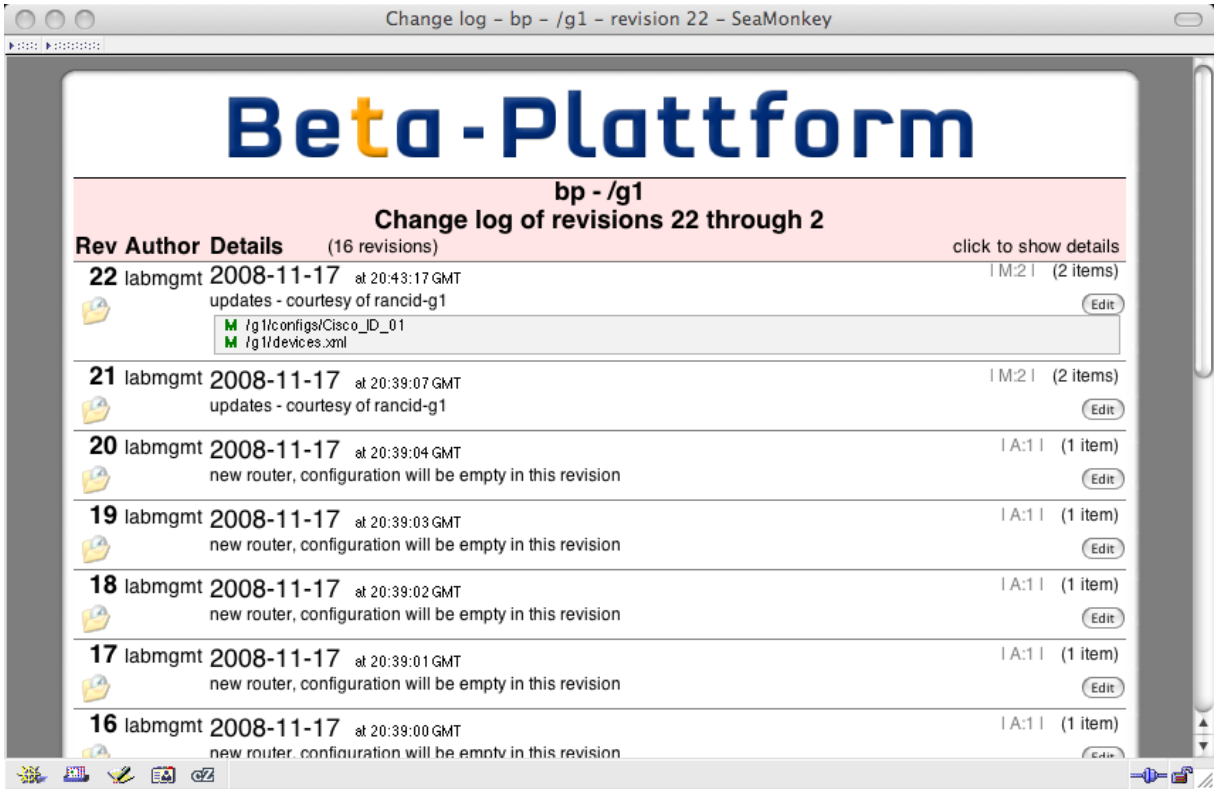


Abbildung 14: GUI Seite Revisionshistorie

Durch Auswahl eines der Felder zu den Revisionen werden die Details ein- und ausgeblendet. Durch Auswahl des rosafarbigen Feldes oben werden die Details für alle Revisionen ein- und ausgeblendet.

Zum Beispiel wird bei Revision 22 vor dem Pfad „/g1/configs/Cisco_ID_01" und mit „| M:2|" auf der rechten Seite angezeigt, welche Aktionen erfolgt sind und an wie vielen Geräten, bzw. Dateien. Die Buchstaben stehen für

- (A)dded – neu hinzugekommen
- (M)odified - geändert
- (D)eleted – gelöscht

Wenn man eines der Ordner-Symbole auf der linken Seite anwählt, gelangt man zur Ansicht zum Zeitpunkt der darüber angegebenen Revision.

Mit dem „Edit" Button kann die Historie im Web Browser geändert werden, wenn dies im „Hooks" Tab eingeschaltet wurde (siehe weiter unten). So kann der Benutzer dokumentieren welche Änderungen er vorgenommen hat und warum.

Wegen der Archivierungsfunktion werden Geräte, deren Buchung abgelaufen ist, nicht aus den Repositories entfernt. Der Fall „Deleted" tritt daher nur bei manueller Löschung mit SVN auf der Kommandozeile auf.

7.1.6 Kontextmenü der Revisionshistorie

Wählt man nun eine der Zeilen der Details aus, erscheint das Kontextmenü zum jeweiligen Gerät.

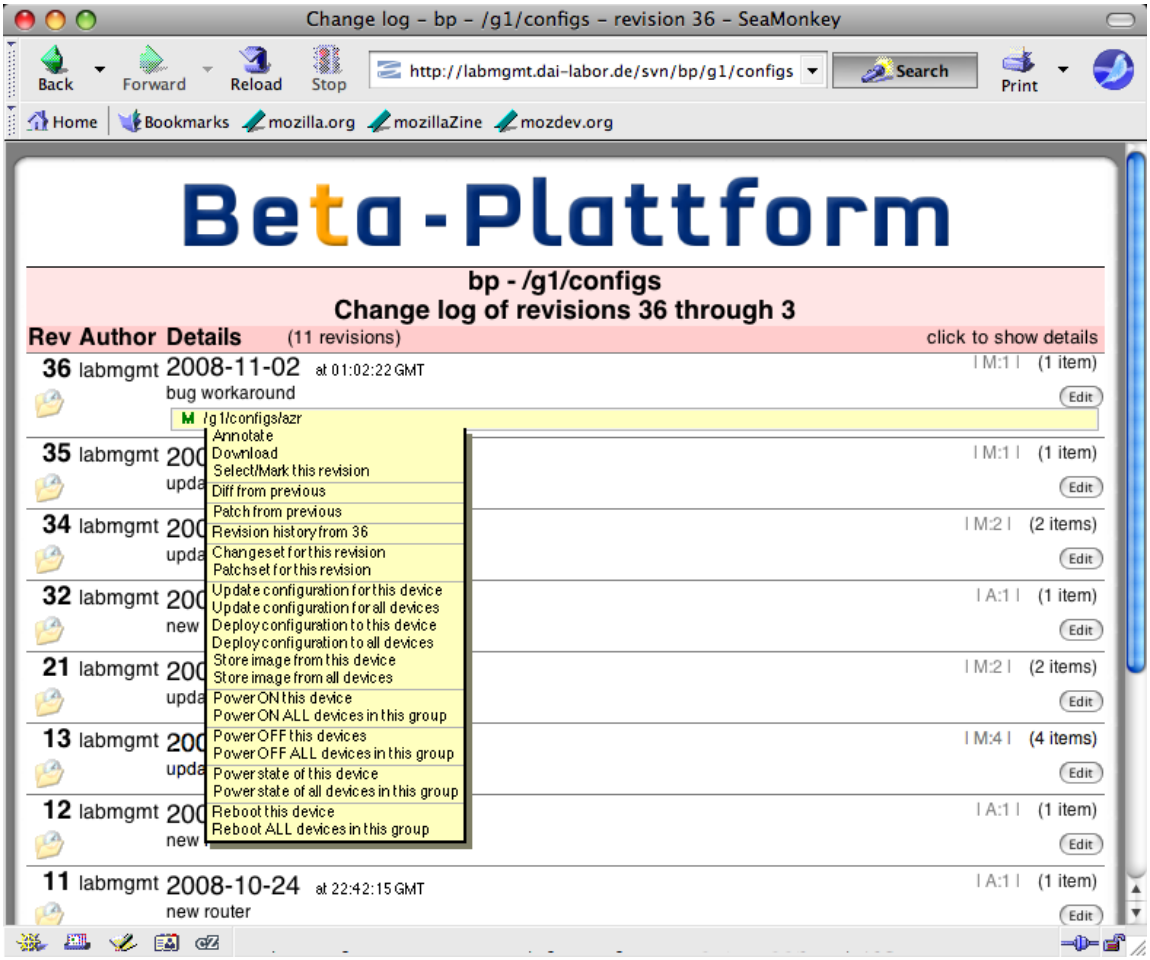


Abbildung 15: GUI Seite Kontextmenü der Revisionshistorie

Der Inhalt des Menüs hängt vom aktuellen Kontext ab und zeigt hier nicht alle Auswahlmöglichkeiten an. Die meisten Funktionen werden aber auf den folgenden Seiten beschrieben.

Die Funktionen „Diff", „Patch", „Changeset" und „Patchset" arbeiten auf der Revision der gewählten Zeile und der vorhergehenden Revision.

Will man diese Aktionen auf zwei beliebigen Revisionen ausführen, so muß man die erste mit „Select/Mark this Revision" auswählen. Bei der Auswahl einer anderen Revision enthält das Menü dann neue Einträge für die Operationen auf der markierten und der ausgewählten Revision.

Die Funktionen „Update", „Store", „Deploy", „Power" und „Reboot" sind die Erweiterungen, die implementiert wurden.

7.1.7 Anzeige einer Konfiguration mit Annotations

Zu jeder Konfiguration kann man sich mit „Annotate" anzeigen lassen, welche Teile der Konfiguration aus welcher Revision stammen:

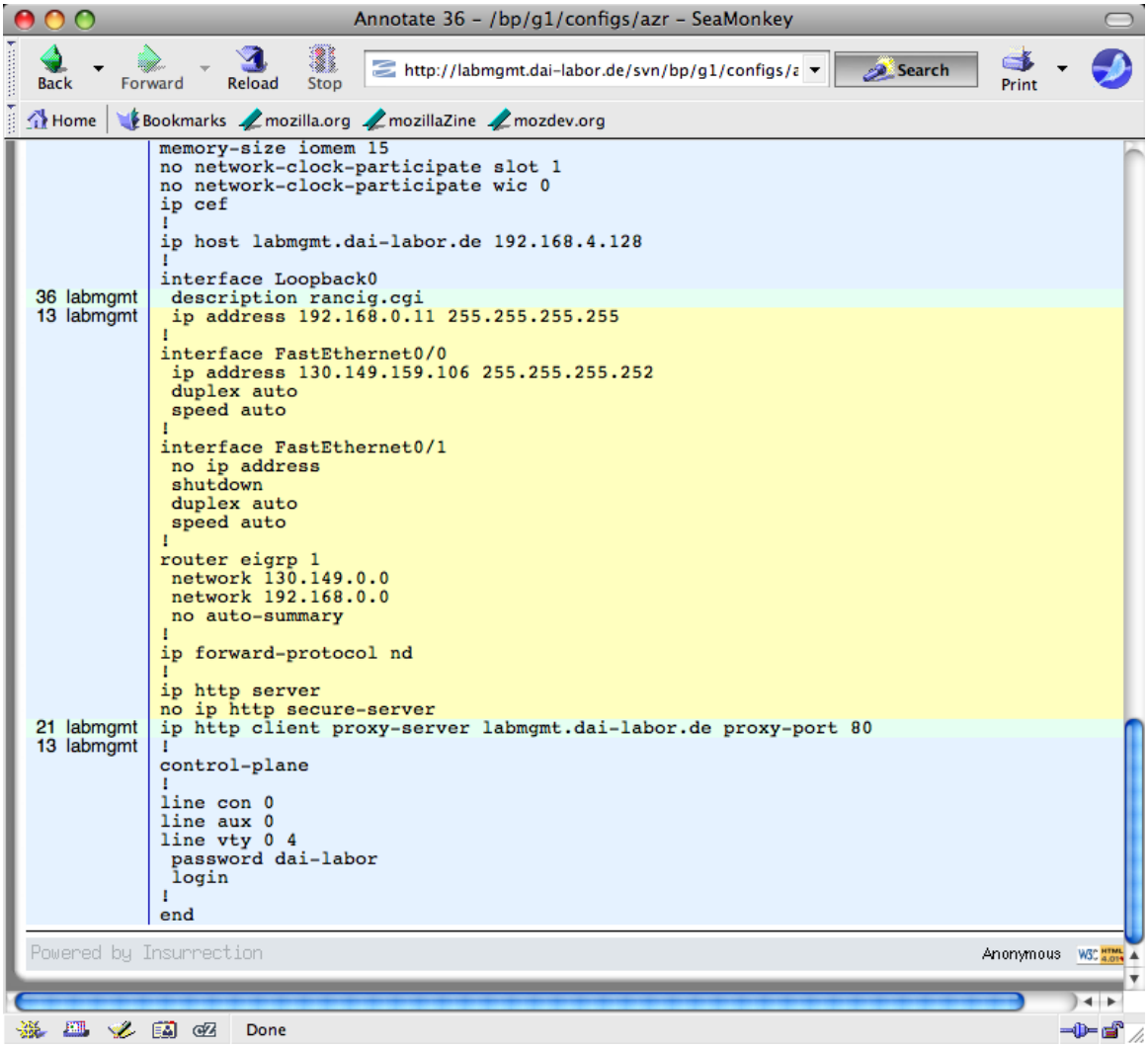


Abbildung 16: GUI Seite einer Konfiguration mit Revisionshinweisen

In der linken Spalte wird angezeigt, in welcher Revision die zugehörigen Zeilen in der rechten Spalte zustande gekommen sind.

Hier wurde zum Beispiel die Zeile „description rancid.cgi" in Revision 36 zuletzt bearbeitet.

Wählt man eines der Felder an, bekommt man die Historie angezeigt, in der die jeweilige Änderungen dokumentiert sind.

7.1.8 Anzeige der Unterschiede zwischen zwei Konfigurationen

Die Funktion „Diff" gibt die Unterschiede zweier Konfigurationen als „Context Diff" aus. Zusätzlich werden die geänderten Zeilen farblich hinterlegt.

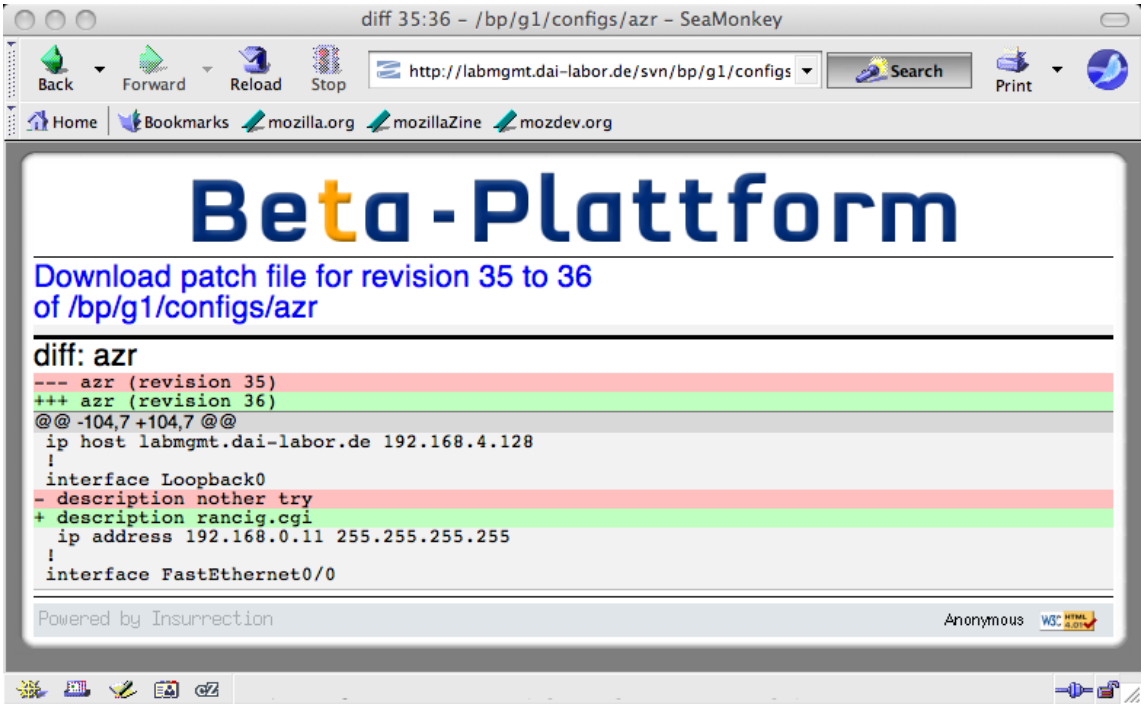


Abbildung 17: GUI Seite Anzeige der Unterschiede zweier Konfigurationen

- Zeilen mit einem „-" Minus wurden gelöscht und sind rot hinterlegt.
- Zeilen mit einem „+" Minus wurden hinzugefügt und sind grün hinterlegt.
- Eine Kombination aus roten und grünen Zeilen können auch geänderte Zeilen sein. Hier ist das bei den Zeilen mit „description" der Fall.

Im Normalfall wird der Unterschied zwischen der aktuellen und der vorherigen Revision angezeigt. Will man zwei beliebige Revisionen vergleichen, so kann man im Kontextmenü mit „Mark/Select" eine bestimme Revision auswählen. Im Kontextmenü der anderen Revision erscheint dann „Diff to selected revision: XX".

Die Funktion „Changeset" führt „Diff" für die ganze Gerätegruppe durch und zeigt das Ergebnis als eine einzige Seite an.

Die Funktionen „Patch" und „Patchset" entsprechen „Diff" und „Changeset", die Ergebnisse werden aber nicht als Webseiten angezeigt, sondern als Dateien zum Download erzeugt. Diese können dann von Tools wie „patch" verwendet werden.

7.1.9 Nutzung der RANCID Erweiterungen

Die Erweiterungen können ebenfalls über das Kontextmenü aufgerufen werden. Diese Funktionen sind:

- „Update Configuration for this device / all devices" stößt den Abruf der Konfigurationen von einem Gerät oder einer Gerätegruppe an.
- „Store image from this device / all devices" stößt den Abruf der Software Images oder Dateien von einem Gerät oder einer Gerätegruppe an.
- „Deploy configuration to this device / all devices" stößt den Rollback der Konfiguration von einem Gerät oder einer Gerätegruppe auf die Revision des Eintrags an.
- „Power on this device / all devices" schaltet ein Gerät oder eine Gerätegruppe ein.
- „Power off this device / all devices" schaltet ein Gerät oder eine Gerätegruppe aus.
- „Power state of this device / all devices" zeigt den Zustand des Anschlusses für ein Gerät oder eine Gerätegruppe an.
- „Reboot this device / all devices" führt einen Kaltstart für ein Gerät oder eine Gerätegruppe aus

Die Funktionen „Update", „Store" und „Deploy" laufen oft lange und werden daher als Batch Prozesse im Hintergrund abgearbeitet. Im GUI wird nur ausgegeben, welcher Befehl ausgeführt werden wird. Dabei werden in den Verzeichnissen der Gruppen zwei Dateien „save.log" und „deploy.log" mit kurzen Ergebnisausgaben angelegt. Die zugehörigen Protokolldateien mit detaillierten Logdaten liegen in „/home/labmgmt/rancid/bp/logs/".

Sofern Änderungen im Repository zustande kommen, werden die Änderungen per Email und RSS/ATOM Feed verbreitet.

Die Funktionen zum Strommanagement geben außer bei „Power State" nur im Fehlerfall etwas aus. Sie laufen aber inline ab. Eventuelle Meldungen werden also in der Webseite angezeigt.

Abbildung 18: GUI Seite RANCID Feedback

7.1.10 Ändern der Revisionshistorie

Wählt man in der Historie den „Edit" Button, kann man mit einem Editor die Änderungen am Labor für die jeweilige Revision dokumentieren. Dazu braucht man die notwendige Berechtigung zum Schreiben und die Funktion muss im „Hooks" Tab eingeschaltet sein.

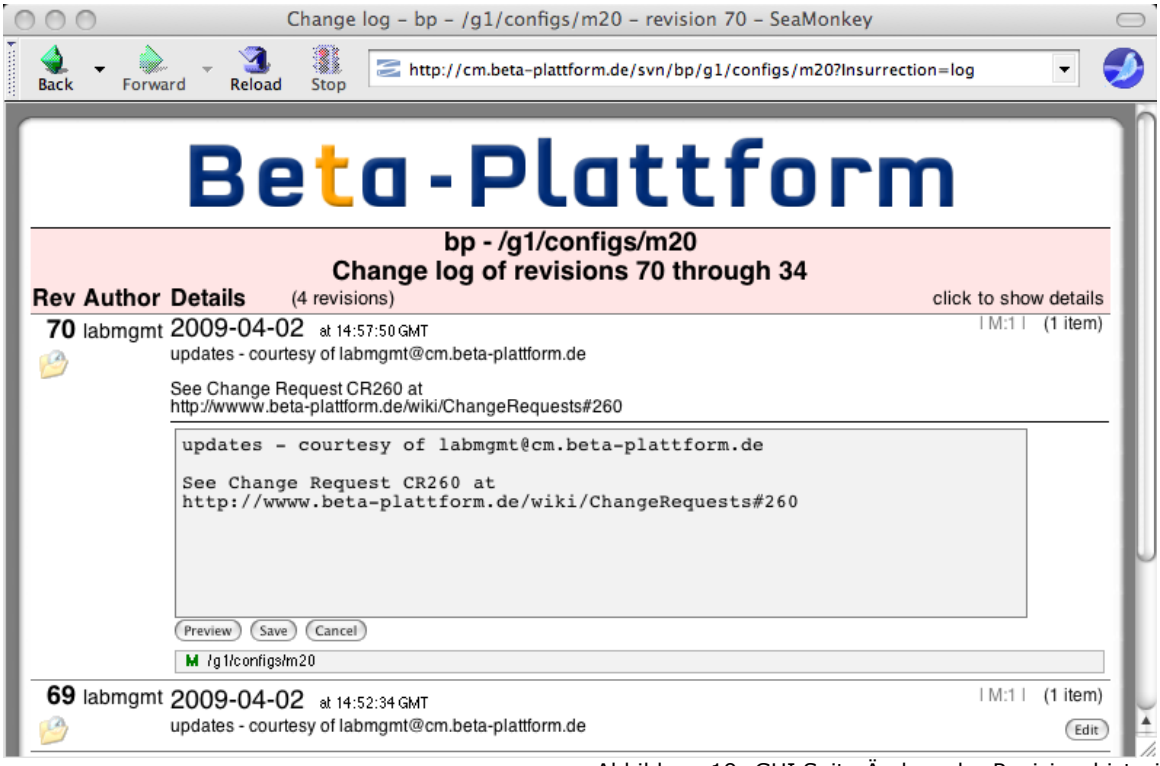


Abbildung 19: GUI Seite Ändern der Revisionshistorie

In dem Feld kann nun ein beliebiger Text eingegeben werden.

Die Buttons haben folgende Funktionen:

- Mit „Save" werden die Änderungen gesichert.
- Mit „Cancel" wird der Vorgang abgebrochen.
- Mit „Preview" bekommt man eine Vorschau der Änderungen angezeigt.

Es kann kein HTML verwendet werden.

7.2 Administration vom GUI und den Repositories

Zum Zugriff auf die Administration muß man sich auf der Startseite mit „Login" anmelden und über die notwendigen Rechte verfügen. Auf der Startseite erscheinen dann „Admin" Buttons für alle Repositories für die man entsprechende Rechte hat.

Abbildung 20: GUI Seite Login zur Administration

Hier sind diese Funktionen erreichbar:

- Mit „Change Password" kann man sein Passwort ändern.
- Mit dem linken „Admin" Button kann das GUI selbst verwaltet werden. Insbesondere können dort neue Repositories angelegt werden.
- Mit den rechten „Admin" Buttons können die jeweiligen Repositories verwaltet werden.

Über jede Admin Seite ist auch die jeweilige Benutzerverwaltung erreichbar.

7.2.1 Anlegen neuer Repositories und Verwaltung der Systemadministratoren

Mit dem GUI können für RANCID neue Repositories angelegt werden. Alle angegebenen Gerätegruppen werden in einem einzigen Repository abgelegt.

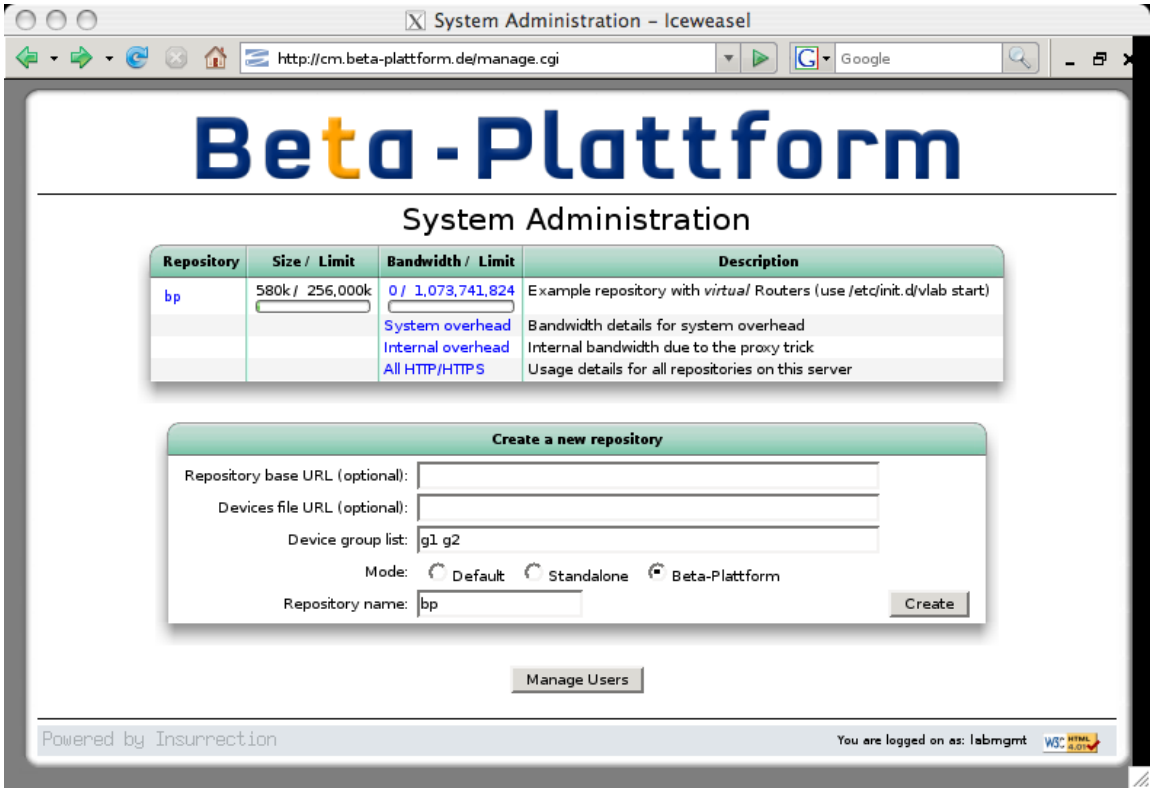


Abbildung 21: GUI Seite Adminstration der Repositories

Die Parameter sind:

- Der Name des Repository, hier „bp"
- Eine beliebig lange Liste von Gruppennamen, hier „g1 g2". Es sollten aber nicht zu viele Gruppen sein, da sonst der „Devices"-Tab zur Verwaltung sehr lang wird.
- Optional das URL von dem die Datei mit den Gerätedaten im Format „devices.xml" geladen werden soll. Ohne Angabe wird der Default DEVICES_URL aus „_template.conf.xml" um den String „devices.xml" ergänzt.
- Optional das URL für den Speicherort des Subversion Repository. Repositories auf anderen Servern werden derzeit nur eingeschränkt unterstützt. Ohne Angabe wird der Default CVSROOT aus „_template.conf.xml" um den String „bp" ergänzt.
- Der Modus in dem das Repository betrieben werden soll. Die Option „Standalone" ist für den lokalen Betrieb in einem Testbed, also unabhängig von der Beta-Plattform. Bei der Option „Beta-Plattform" werden für den föderierten Betrieb notwendige Funktionen ausgeführt, zum Beispiel wird geprüft ob die Buchung der Geräte noch besteht. Bei „Default" gilt der Wert aus „_template.conf.xml".

Sollen später eine weitere Gerätegruppe hinzugefügt werden, so muss dies in „bp.conf.xml" auf der Kommandozeile als neue Gruppe eingetragen werden. Dies wird im Kapitel „Nutzung mit der Kommandozeile" erklärt.

Unter dem Button „Manage Users" wird eingestellt, welche Benutzer das System verwalten dürfen.

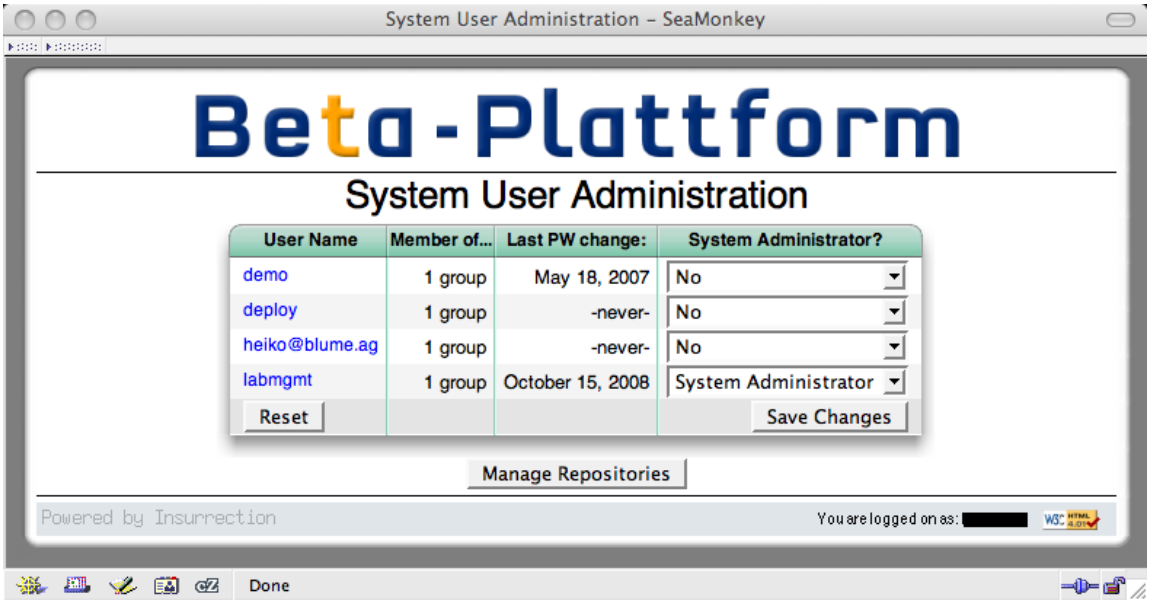


Abbildung 22: GUI Seite Verwaltung der Administratoren

Hier kann das Privileg zur Verwaltung des Systems vergeben werden. Dies umfasst

- Das Anlegen neuer Repositories
- Die hier beschriebene Funktion selbst
- Ansicht der Statistiken
- Die Verwaltung aller Repositories

7.2.2 Anlegen neuer Repository Benutzer

Die Administration der Repositories hat eine eigene Benutzerverwaltung. Mit dem Reiter „New User" legt man neue Benutzer für das Repository an:

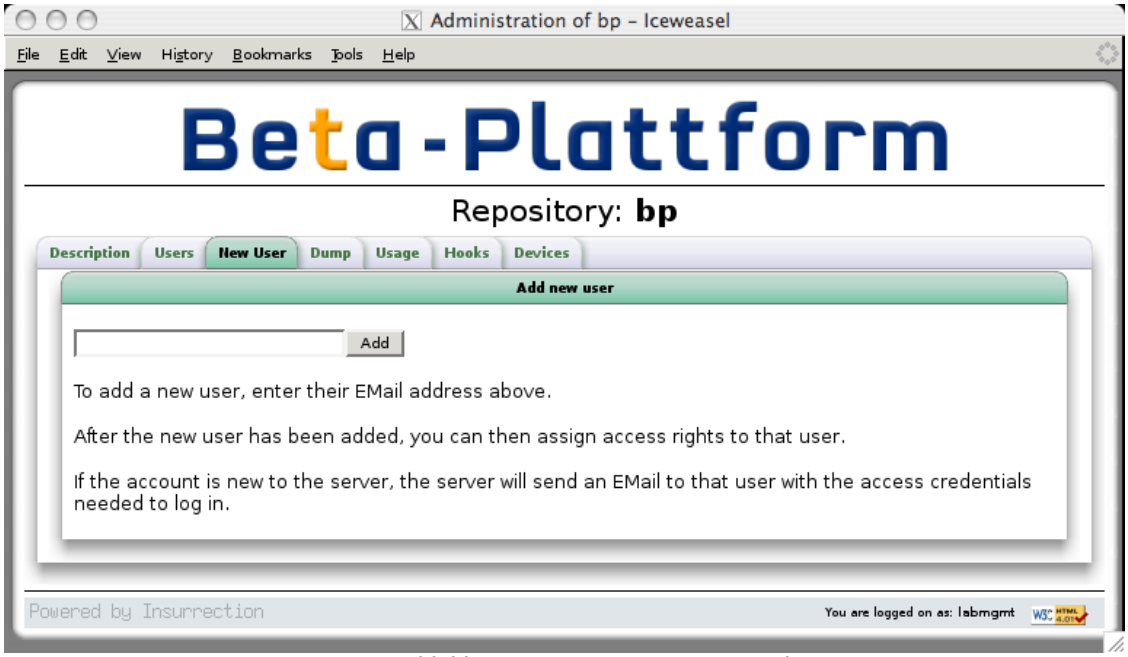


Abbildung 23: GUI Seite zum Einrichten neuer Repository Benutzer

Man gibt die Email Adresse des neuen Benutzers an. An diese Adresse wird dann das automatisch generierte Passwort gesendet.

Neue Benutzer können nur lesend auf das Repository zugreifen. Erweiterte Rechte können im „Users"-Tab vergeben werden.

7.2.3 Zuweisen von Rechten an Repository Benutzer

Den Benutzern kann man im Reiter „Users" verschiedene Rechte zuweisen:

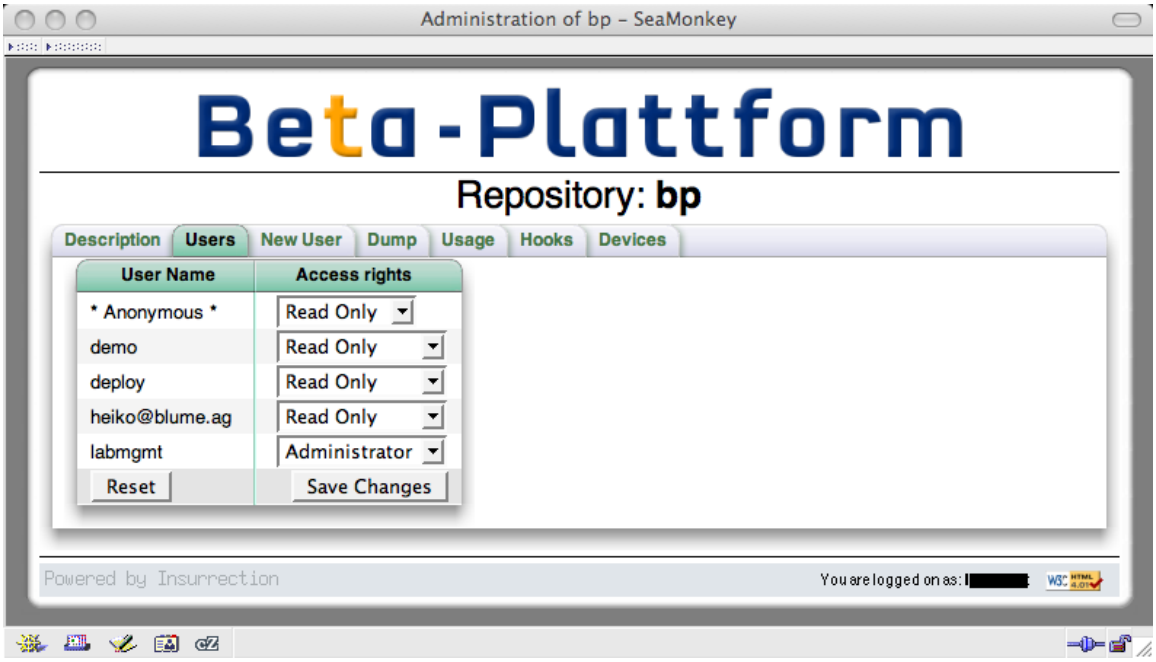


Abbildung 24: GUI Seite Administration der GUI Benutzer

Hier kann für jeden angemeldeten Benutzer und für anonyme Besucher festgelegt werden, welche Rechte bestehen sollen:

- Nur Lesen des Repositories
- Lesen und Schreiben des Repositories
- Administration des Repositories

Für die „Deploy" Funktion muss der Zugriff für anonyme Benutzer auf „Read Only" sein.

7.2.4 Verwaltung der Gerätegruppen und Geräte

Auf dieser Seite stehen zahlreiche Funktionen zur Verfügung. Für jede Gerätegruppe wird eine eigene Box dargestellt, hier ist exemplarisch nur eine zu sehen:

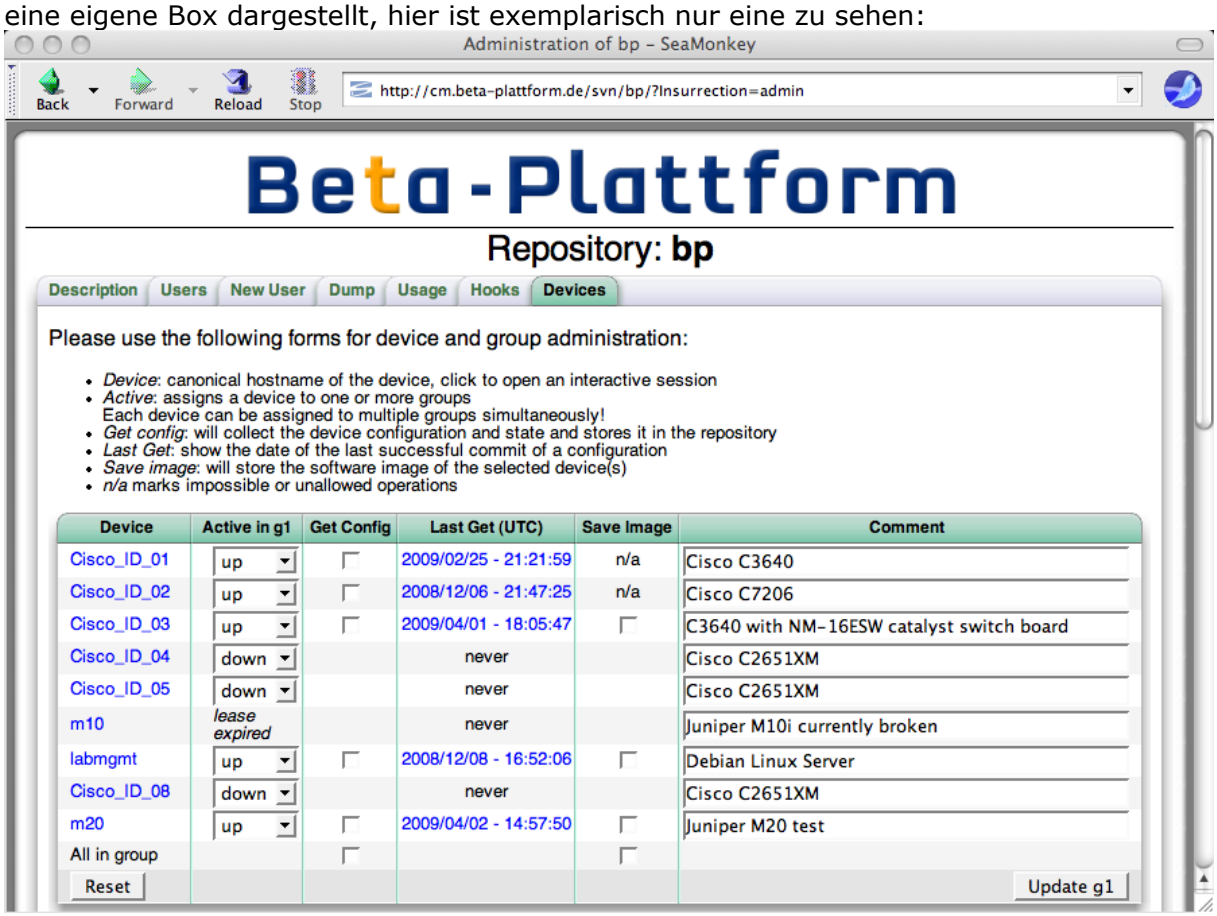


Abbildung 25: Gerätegruppe im Devices-Tab

Wie im Kontextmenü sind viele Aktionen auf den Geräten möglich:

- Zuordnung der Geräte zu den Gerätegruppen („up", „down"). Jedes Gerät kann mehreren Gruppen angehören, hat aber in jeder Gruppe eine separate Konfiguration.
- Es können Kommentaren zu den Geräten eingegeben werden
- Der Start einer Session ist durch Klicken auf den Namen des Gerätes möglich
- Die Anzeige der Konfigurationen ist durch Klicken auf den Zeitstempel in der Spalte „Last Get" möglich

- „Get" holt die aktuelle Konfiguration eines Gerätes oder einer Gerätegruppe. Dies gilt nur für Geräte, die der jeweiligen Gruppen zugeordnet („up") sind.

- „Save" sichert das Image bzw. die Dateien eines Gerätes oder einer Gerätegruppe entsprechend den Einstellungen in „devices.xml".

Der untere Teil des Devices-Tab enthält die Funktionen für Deploy und Update:

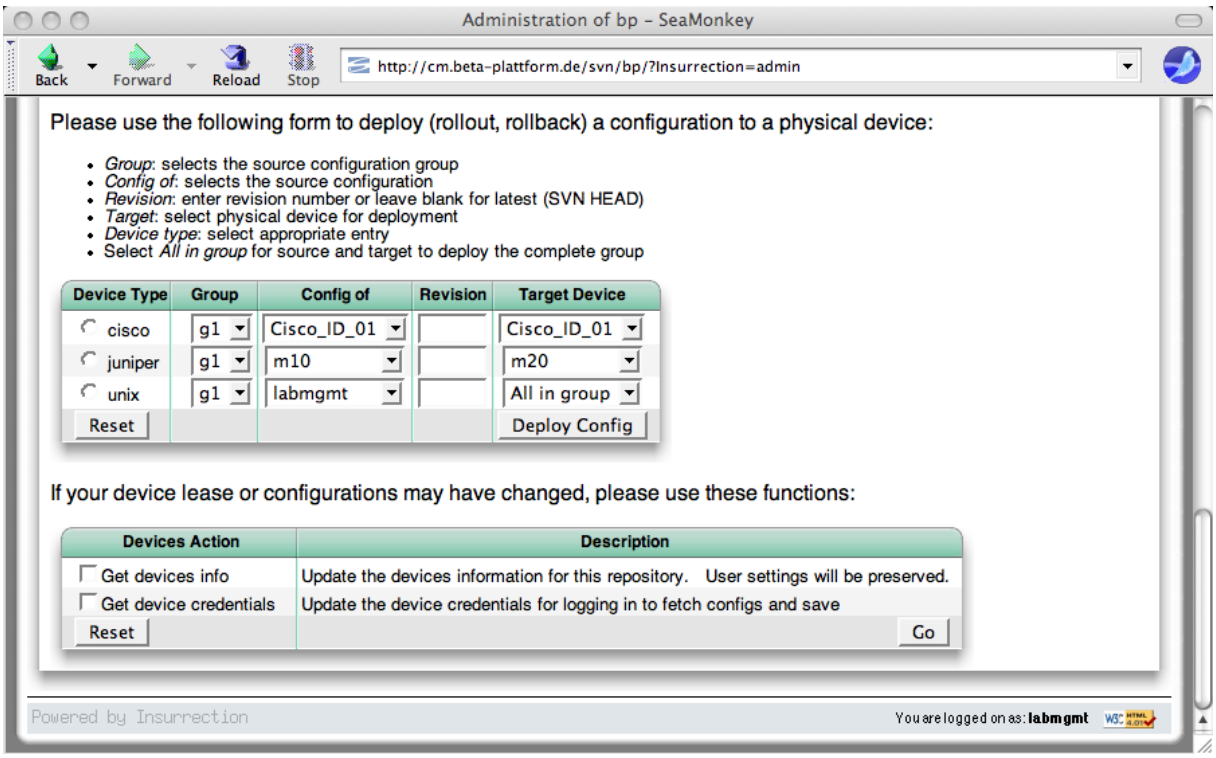


Abbildung 26: Deploy- und Update-Funktion im Devices-Tab

- „Deploy" überträgt und aktiviert eine bestimmte Konfiguration aus dem Repository auf das angegeben Gerät oder der Gerätegruppe, sofern das jeweilige Gerät das unterstützt bzw. die Operation erlaubt ist. Dies kann auch die Konfiguration eines anderen Gerätes und aus einer anderen Gruppe sein, solange sie vom selben Typ ist. Wenn keine Revision angegeben wird, kommt die zuletzt archivierte (SVN „HEAD") zum Tragen.

  Wegen eines Bugs in IOS (nicht konform zu RFC2616[21]) musste ein Workaround in der Konfiguration implementiert werden. Ausserdem wird das Feature in Cisco IOS noch weiterentwickelt und ist daher nur eingeschränkt nutzbar. Z.B. bleibt in älteren Releases die Reihenfolge von Route-Maps nicht erhalten. Es sollten daher die entsprechenden Release Notes für das Image auf dem Gerät beachtet werden. Siehe auch das Kapitel „Nutzung auf der Kommandozeile"

- Es können die Gerätedaten („devices.xml", „.cloginrc") aktualisiert werden. Dies sollte nur dann notwendig sein, wenn sich die Buchungen der Geräte verändert haben, oder eine Aktualisierung der Gerätedaten (z.B. durch die Beta Plattfom) vorgenommen wurde.

Es ist zu beachten, dass die Benutzer die Überlappung der Gruppenzugehörigkeit bei Nutzung der „Get" Funktion selbst berücksichtigen müssen. Führt man die Funktion in der „falschen" Gruppe aus, können Inkonsistenzen der verschiedenen Szenarien entstehen, die man eigentlich durch die Gruppen trennen wollte. Der Sinn der

[21]http://blog.ioshints.info/2006/12/cisco-ios-violates-rfc-2616-http11.html

Überlappung liegt z.B. in der Möglichkeit den selben Ethernet Switch für mehrere Szenarien zu nutzen und die Konfiguration – auch wenn sie Teile anderer Szenarien enthält – in jeder Gruppe archivieren zu können.

7.2.5 Backup eines Repositories als Dump

Hier kann man einen Backup des gesamten Repositories im Subversion Format über den Web Browser herunterladen.

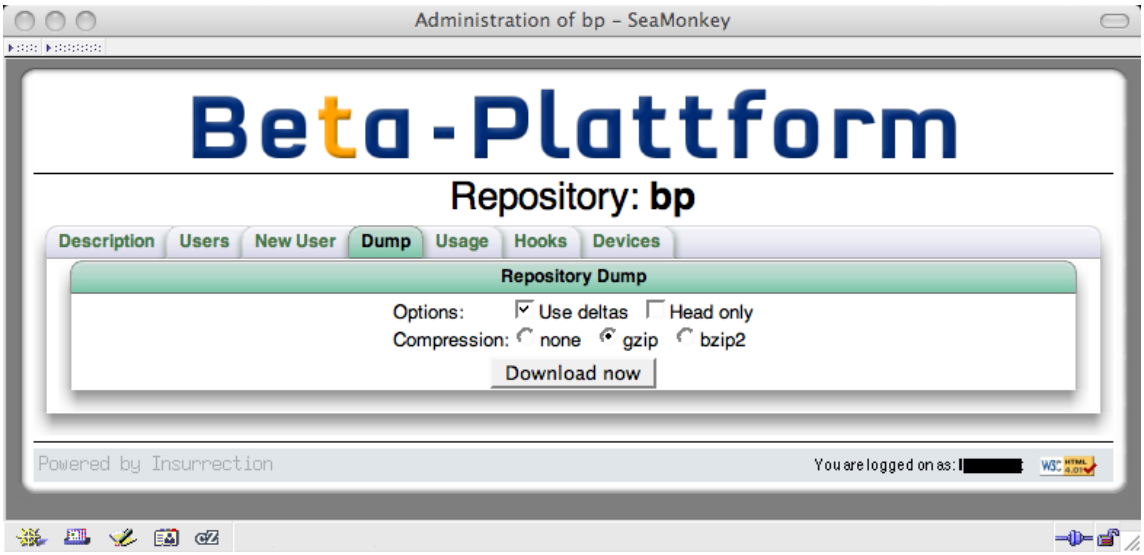
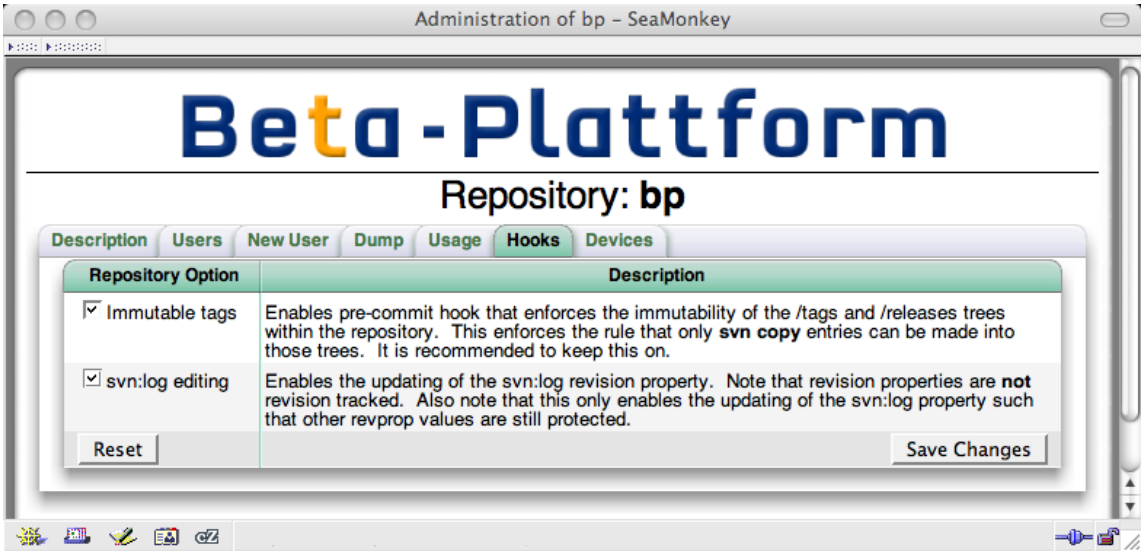


Abbildung 27: GUI Seite Backup eines Repositories

Die Datei wird dabei auf den Rechner geladen, auf dem der Browser läuft, also nicht unbedingt auf den Rechner auf dem die Laborverwaltung läuft. Bei „Head only" wird nur der aktuelle Stand erzeugt, also ohne frühere Zustände.

7.2.6 Hooks zum Einschalten der History-Editierung

Um das Ändern der Revisionshistorie zu ermöglichen muss man hier den zweiten Toggle setzen und die Änderung sichern.



7.3 Nutzung mit der Kommandozeile

7.3.1 Basiskonfiguration der Gerätegruppen

Normalerweise werden alle Aktionen vom Benutzer „labmgmt" ausgeführt. Nur Änderungen an Systemdateien sollten durch „root" erfolgen! Sonst kann es zu Problemen mit den Berechtigungen der Dateien kommen.

Am Beispiel des „bp" Testbed werden hier die Schritte zum Einrichten einer Gerätegruppe dargestellt. Normalerweise wurde dies bereits mit dem GUI erledigt. Es zeigt aber auch die relevanten Dateien und deren Formate auf.

Im DNS oder in „/etc/hosts" müssen die Hostnamen auf die IP Adressen gebunden werden. Hier verwenden wir „/etc/hosts":

```
# Virtual Routers

# C3640
192.168.0.1       Cisco-ID-01
# C7206
192.168.0.101     Cisco-ID-02
```

In „/home/labmgmt/rancid/bp.conf.xml" muß der Gruppenname eingetragen sein:

```
LIST_OF_GROUPS="g1 g2"
```

Nur sofern neu, muß das Repository für die neue Gruppe initialisiert werden:

```
labmgmt> rancid-cvs -f /home/labmgmt/rancid/bp.conf.xml g1
```

In „/home/labmgmt/rancid/**bp/g1**/devices.xml" sind die Geräte eingetragen (der Abruf erfolgte vorher von der Beta Plattform oder im Standalone Betrieb für „bp" aus /home/labmgmt/BP/bp/devices/").

```
<?xml version="1.0"?>
<rancidDevices id="bp">
    <device id="1" version="2">
       <comment>Cisco 2651XM </comment>
        <hostname>Cisco_ID_01</hostname>
        <accessURL>telnet://Cisco-ID-01:23/</accessURL>
        <loginrcURL>http://localhost/BP/bp/d1</loginrcURL>
        <rancidState>collected</rancidState>
        <deviceType>cisco</deviceType>
        <collectFlag>up</collectFlag>
        <deployAllowed>false</deployAllowed>
        <saveImageAllowed>true</saveImageAllowed>
        <saveImage>off</saveImage>
        <saveImageURL>ftp://user:pass@cm/BP/bp/save</saveImageURL>
        <deviceControl>
            <hostname>pwrctrl.beta-plattform.de</hostname>
            <outlet>3</outlet>
            <readCommunity>secret</readCommunity>
            <writeCommunity>secret</writeCommunity>
            <powerStateOID>sPDUOutletCtl</powerStateOID>
            <powerCtlOID>sPDUOutletCtl</powerCtlOID>
            <powerOnOP>outletOn</powerOnOP>
            <powerOffOP>outletOff</powerOffOP>
            <rebootOP>outletReboot</rebootOP>
        </deviceControl>
    </device>
</rancidDevices>
```

Mögliche Gerätetypen: agm, alteon, baynet, cat5, cisco, css, enterasys, erx, extreme, ezt3, force10, foundry, hitachi, hp, juniper, mrtd, netscaler, netscreen, procket, redback, riverstone, smc, tnt, unix, zebra.

Damit RANCID sich auf den Geräten anmelden kann, müssen die zugehörigen Namen und Passwörter der Benutzer in „/home/labmgmt/rancid/bp/.cloginrc" eingetragen sein. **Die Datei wird aber nicht manuell gepflegt.** Das Format der Datei ist sehr variabel und wird im Anhang erklärt. Hier ein Beispiel, mit dem RANCID mit dem Benutzer „backup" direkt nach dem Login in den privilegierten „enable" Zustand kommt:

```
add user        Cisco-ID-03       backup
add password    Cisco-ID-03       {*******}
add autoenable  Cisco-ID-03       1
add method      Cisco-ID-03       ssh
```

Für jedes Gerät wird ein solcher Datensatz von dem jeweiligen <loginrcURL> des Gerätes in „devices.xml" geladen. Bei der Standalone Installation liegen diese Dateien in „/home/labmgmt/BP/devices/". Sie werden beim Anlegen des Repositories geladen, und können mit der o.g. Update Funktion aktualisiert werden.

So kann das Repository mit den Konfigurationen der Geräte der Gruppe gefüllt werden:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml g1
```

Die Konfigurationen werden in „/home/labmgmt/rancid/bp/**g1**/configs/" abgelegt und dann in das Repository eingebracht.

7.3.2 Konfiguration eines Gerätes anzeigen

Alle gängigen Programme wie „emacs", „vi", „more" usw, können verwendet werden, um die Dateien anzuzeigen oder zu bearbeiten:

```
labmgmt> less /home/labmgmt/rancid/bp/g1/configs/Cisco_ID_01
```

Die Dateien dürfen dort aber nicht verändert werden!

Aus den Repositories können die Konfigurationen z.B. mit cURL abgerufen werden:

```
curl http://cm.beta-plattform.de/svn/bp/g1/configs/Cisco_ID_01
```

7.3.3 Holen der Konfigurationen einer Gerätegruppe

Mit diesem Befehl werden die Konfigurationen aller Geräte der Gruppe geholt:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml g1
```

Der Vorgang kann lange dauern, insbesondere dann, wenn viele Geräte nicht erreichbar sind. Es wird dann mehrfach versucht diese zu erreichen. Die Anzahl der Versuche kann in „bp.conf.xml" eingestellt werden.

Die Konfigurationen werden in „/home/labmgmt/rancid/**bp**/**g1**/configs/" abgelegt.

Ohne Angabe eines Arguments werden alle Gerätegruppen abgearbeitet.

Die Log Dateien liegen in „/home/labmgmt/rancid/**bp**/logs/".

7.3.4 Holen der Konfiguration eines einzelnen Gerätes

Analog zum vorherigen Befehl wird durch die Option „-r" die Konfigurationen eines einzigen Gerätes der Gruppe geholt:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml \
            -r Cisco_ID_01 g1
```

Die Konfiguration wird in „/home/labmgmt/rancid/**bp**/**g1**/configs/**Cisco_ID_01**" abgelegt.

7.3.5 Übertragen von Konfigurationen aus dem Repository auf Geräte

Für alle Geräte, für die in „devices.xml" die Deploy Funktion erlaubt ist, können Konfigurationen aus dem Repository auf die Geräte übertragen werden. Dies ist insbesondere bei Cisco IOS abhängig von der Version der installierten Software und dem Einverständnis der Testbedbetreiber. Der Testbedbetreiber muss dafür sorgen, dass wichtige Teile der Konfiguration nicht überschrieben werden können, z.B. die IP Addressen der Management Interfaces. Dies kann über die Authorization Funktionen von AAA Servern wie RADIUS und TACACS+ oder lokal in der Konfiguration erfolgen. Es empfiehlt sich für den Notfall eine „goldene Konfiguration" vorzubereiten um den Normalzustand wieder herstellen zu können. Bei Cisco IOS heisst dieses Feature „Resilient Configuration[22]". Bei JUNOS heisst das Feature „Rescue Configuration[23]". Bei JUNOS muss ein Benutzer verwendet werden, der nach dem Login direkt in das CLI kommt und nicht in die UNIX Shell, daher ist der Benutzer „root" nicht geeignet.

Cisco IOS

Wegen eines Bugs in IOS (nicht konform zu RFC2616[24]) musste ein Workaround in der Konfiguration implementiert werden:

```
ip http client proxy-server cm.beta-plattform.de proxy-port 80
```

Dadurch werden alle HTTP Request vom Router durch den internen Proxy auf cm.beta-plattform.de geleitet. Sollte für den Router eine DNS Auflösung des Hostnames nicht möglich sein muss zusätzlich ein Host Eintrag erfolgen:

```
ip host cm.beta-plattform.de 104.254.254.43
```

Ausserdem wird das Feature in Cisco IOS derzeit noch weiterentwickelt und ist daher nur eingeschränkt nutzbar. Z.B. bleibt die Reihenfolge von Route-Maps nicht erhalten. Es sollten daher die entsprechenden Release Notes für das Image auf dem Gerät beachtet werden.

[22]http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_resil_config.html
[23]http://www.juniper.net/techpubs/software/junos/junos94/swconfig-cli/returning-to-a-previous-JUNOS-config.html#id-10782526
[24]http://blog.ioshints.info/2006/12/cisco-ios-violates-rfc-2616-http11.html

Um einen bestimmten Revisionsstand aus dem Repository auf ein laufendes Gerät zu übertragen, wird der folgenden Aufruf verwendet:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml \
-D http://cm.beta-plattform.de/svn/bp/g1/configs/Cisco_ID_01?r=HEAD \
      -r Cisco_ID_04
```

Hier wird die neueste Konfiguration („HEAD") des Routers „Cisco_ID_01" aus der Gerätegruppe „g1" auf den Router „Cisco_ID_04" übertragen. Soll eine bestimmte frühere Revision übertragen werden muss anstelle von „HEAD" die Revision angeben werden, z.B. „27".

Um eine komplette Gerätegruppe auf eine bestimmte Revision umzustellen, ist der Aufruf nach diesem Schema möglich:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml \
-D http://cm.beta-plattform.de/svn/bp/g1?r=23 g2
```

So werden alle in der Gruppe „g2" mit „up" aktivierten Geräte die entsprechenden Konfigurationen aus Gruppe „g1" in der Revision „23" übertragen.

Juniper JUNOS

Um einen bestimmten Revisionsstand aus dem Repository auf ein laufendes Gerät zu übertragen, wird der folgenden Aufruf verwendet:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml \
-D http://cm.beta-plattform.de/svn/bp/g1/configs/m10?r=23 \
      -r m10
```

Hier wird die Revision „23" des Routers „m10" aus der Gerätegruppe „g1" auf den Router „m10" übertragen, also ein Rollback des selben Gerätes. Soll die neueste Revision verwendet werden, so muss „HEAD" angegeben oder „?r=23" weggelassen werden.

Um eine komplette Gerätegruppe auf eine bestimmte Revision umzustellen, ist der Aufruf nach diesem Schema möglich:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml \
-D http://cm.beta-plattform.de/svn/bp/g1?r=23 g2
```

So werden alle in der Gruppe „g2" mit „up" aktivierten Geräte die entsprechenden Konfigurationen aus Gruppe „g1" in der Revision „23" übertragen.

7.3.6 Sichern der Software Images oder Dateien einer Gerätegruppe

Mit diesem Befehl werden durch die Option „-i" die Software Images oder Dateien aller Geräte der Gruppe gesichert:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml -i g1
```

Bei Cisco Routern wird das laufende Image gesichert, sofern es nicht über das Netzwerk geladen wurde. Bei UN*X Maschinen wird entsprechend den Parametern in „devices.xml" ein „tar", „cpio" oder ein Script mit den angegebenen Argumenten ausgeführt.

Bei Aufruf durch das GUI oder „rancid-run" werden die Images auf dem in „devices.xml" angebenen URL oder defaultmässig in „bp.conf.xml" global definierten Store

„IMAGE_URL" abgelegt. Dabei wird der Name der Gerätegruppe automatisch angehängt. Es muß auf dem Zielserver also z.B. das Verzeichniss „g1" bereits existieren.

7.3.7 Holen des Software Image oder Dateien eines einzelnen Gerätes

Analog zum vorherigen Befehl wird durch die Optionen „-i -r" das Software Image oder Dateien eines einzigen Gerätes der Gruppe geholt:

```
labmgmt> rancid-run -f /home/labmgmt/rancid/bp.conf.xml \
            -i -r Cisco_ID_01 g1
```

Will man beliebige Dateien direkt von einem Gerät abrufen, so kann dies durch den Aufruf des Skriptes „rancid-image" erfolgen. In der Kommandozeile sind dazu Optionen für den Pfad der Datei auf dem Geräte und optional das URL des Zielservers anzugeben. In diesem Fall wird an den URL nichts angehängt.

```
rancid-image [-dVl] [-f <Flashpath> ] <device> [ <Image-URL> ]
```

Zum Beispiel mit den Befehlen

```
BASEDIR=/home/labmgmt/rancid/bp
export BASEDIR
rancid-image -f flash:c2500-is-l.123-19.bin 192.168.0.1 \
             ftp://egon:SeCrEt@192.168.0.23/backup/
```

wird das „.bin" Image in das Unterverzeichnis „backup" von Egon's Heimatverzeichnis auf dem FTP Server 192.168.0.23 geschrieben. Über die BASEDIR Variable wird der Zugriff auf die „.cloginrc" Datei ermöglicht.

7.3.8 Strommanagement

Die Syntax zum Aufruf des Skriptes lautet:

```
apc-power <group> <device> on | off | reboot | status | name
          use <device> = all to process all devices in group
          default command is 'status'
```

Folgende Funktionen sind implementiert worden:

- „on" schaltet ein Gerät oder eine Gerätegruppe ein.
- „off" schaltet ein Gerät oder eine Gerätegruppe aus.
- „status" zeigt den Zustand des Anschlusses für ein Gerät oder eine Gerätegruppe an.
- „reboot" führt einen Kaltstart für ein Gerät oder eine Gerätegruppe aus

Die notwendigen Informationen für SNMP werden der Datei „devices.xml" entnommen. Zum Beispiel kann das Gerät „Cisco_ID_01" aus der Gerätegruppe „g1" so eingeschaltet werden

```
labmgmt> apc-power -f /home/labmgmt/rancid/bp.conf.xml \
             g1 Cisco_ID_01 on
```

Für Geräte die in „devices.xml" als „down" markiert sind, kann nur der Status und Name des Anschlusses aus dem Powerswitch ausgelesen werden. Die Funktionen, die den Strom schalten werden nur dann ausgeführt, wenn das Gerät in der Gruppe als „up" markiert ist.

7.3.9 Fehlersuche und erweiterte Protokollierung

Die Protokolldateien „error.log" und „access.log" von Apache bzw. Insurrection für alle Repositories liegen in „/home/labmgmt/logs". Die Protokolldateien für die „default" Instanz von Apache liegen in „/var/log/apache2", bei Problemen sollte auch dort nachgesehen werden.

Die Protokolldateien von RANCID für das Repository „bp" liegen in „/home/labmgmt/rancid/**bp**/logs". Die Namen der Dateien bestehen immer aus dem Gruppennamen und einem Timestamp.

Einzelnen Befehlen kann auf der Kommandozeile die Option '-d' mitgegeben werden. Global kann in „bp.conf.xml" eine detaillierte Protokollierung eingestellt werden

```
DEBUG="on"
```

Bei vielen Skripten muss die Shell Variabe „BASEDIR" auf das Arbeitsverzeichnis für das jeweilige Repository gesetzt werden:

```
BASEDIR=/home/labmgmt/rancid/bp; export BASEDIR
```

7.4 Automatisierung des Abrufens der Konfigurationen

Der zyklische Abruf der Konfigurationen und Software Images erfolgt mit CRON Einträgen für den Benutzer „labmgmt". Dazu werden mit „crontab -e" z.B. die folgenden Einträge gemacht:

```
0 0,4,8,12,16,20 * * * /usr/local/bin/rancid-run -f \
              /home/labmgmt/rancid/bp.conf.xml
0 3 * * 1 /usr/local/bin/rancid-run -f \
              /home/labmgmt/rancid/bp.conf.xml -i g1
```

So würden von „bp" die Konfigurationen alle vier Stunden und die Software Images der Gruppe „g1" einmal wöchentlich abgerufen.

Standardmässig ist kein automatischer Abruf eingerichtet.

7.5 Nutzung des virtuellen Cisco Labors

Mit DynaMIPS und DynaGen können Netze aus virtuellen Routern aufgebaut werden. Ein Beispiel für eine Emulation für das Testbed „bp" wurde als „vlab" implementiert.

Die Startsequenz dafür ist

```
/etc/init.d/vlab start
```

Das Boot Script „/etc/inet.d/vlab" bzw. „/etc/rc2.d/S99vlab" startet zwei abgehängte „screen" Sessions. Eine, in der mittels „/home/labmgmt/DynaLab/vlab/vlab.screenrc" DynaMIPS und das „vlab.routes" Script laufen und eine, in der DynaGen mit der Konfiguration in „/home/labmgmt/DynaLab/vlab/vlab.net" läuft.

An diese Sessions kann man sich zur Diagnose und Administration wieder anhängen:

- Als Benutzer „labmgmt" mit „screen -r -S DYNAGEN"
- Als Benutzer „root" mit „screen -r -S DYNAMIPS"

Zusätzlich müssen im Linux Routen zu den Netzen im virtuellen Labor zu der Adresse der Schnittstelle des Routers zeigen, die gebrückt wird. Dies erfolgt mit dem Script „/home/labmgmt/DynaLab/vlab/vlab.routes". Es wird vom Boot-Skript aufgerufen. Die zu routenden Netze sind darin hartkodiert.

Die Adressvergabe an den Router erfolgt per Default mit dem DHCP des LAN.

Vom Boot Script wird „vlab.routes" ohne Argument aufgerufen. Dann wird das Script „get-vgw-ip.pl" aufgerufen, dass die Adresse auf Layer 2 aus CDP[25] Paketen mitliest. Dazu muss auf dem Interface des Routers CDP mit „cdp enable" aktiviert sein.

```
usage: get-vgw-ip.pl -i <interface> -r <routername> -p <routerport>
```

Ist auf dem Router CDP abgeschaltet, muss die Adresse dem Script „`vlab.routes`" als Argument übergeben werden.

Um die Routen zu entfernen, kann „vlab.routes" mit der Option „-d" aufgerufen werden.

Nach einem VM Resume muss „vlab.routes" manuell aufgerufen werden:

```
root@labmgmt:/# /home/labmgmt/DynaLab/vlab/vlab.routes
obtaining gw address with CDP, this will take a while
got 192.168.4.150 - trying that
PING 192.168.4.150 (192.168.4.150) 56(84) bytes of data.
64 bytes from 192.168.4.150: icmp_seq=1 ttl=255 time=16.3 ms
64 bytes from 192.168.4.150: icmp_seq=2 ttl=255 time=8.42 ms
64 bytes from 192.168.4.150: icmp_seq=3 ttl=255 time=6.95 ms
64 bytes from 192.168.4.150: icmp_seq=4 ttl=255 time=4.93 ms
64 bytes from 192.168.4.150: icmp_seq=5 ttl=255 time=3.99 ms

--- 192.168.4.150 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 3.994/8.140/16.391/4.406 ms
192.168.4.150 is reachable

Using address 192.168.4.150 as vlab gateway
```

Manuell kann das „vlab" so gestartet werden:

Benutzer	***Kommandozeile***	***Nutzung***
root	cd /home/labmgmt/DynaLab/vlab	Zum Arbeitsverzeichnis
root	/usr/local/bin/dynamips -H 7200	Hypervisor für virtuelle Router
labmgmt	cd /home/labmgmt/DynaLab/vlab	Zum Arbeitsverzeichnis
labmgmt	/usr/local/bin/dynagen vlab.net	Startet virtuelle Router
root	sh vlab.routes	LAN Anbindung & Routen
labmgmt	telnet localhost 2003	Zugang zur Konsole Router 3

Tabelle 7: Startsequenz virtuelle Router

Das Telnet und Enable Passwort für die virtuellen Router ist „vlab".

Um lange Startphasen durch das Dekomprimieren der IOS Images beim Booten zu vermeiden, sollte man diese vorher entpacken. Hier ein Beispiel für ein 7200er Image:

1. unzip c7200-advipservicesk9-mz.124-11.T1.bin
2. mv C7200-AD.BIN ~labmgmt/DynaLab/images/c7200-advipservicesk9-mz.124-11.T1.img

[25]Cisco Discovery Protocol

Für jedes neue Image muß das „Idle-PC" Verfahren[26] durchgeführt werden, da sonst schon ein einziger virtueller Router die CPU des Hosts voll auslastet.

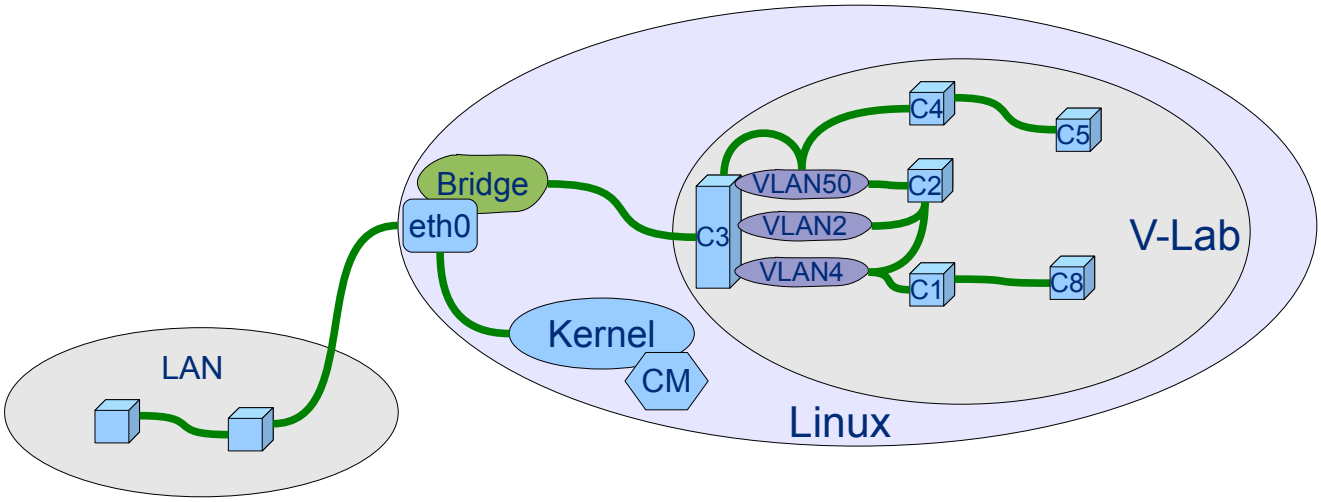
Dazu muss man sich an die „DYNAGEN" Session anhängen, um das CLI von Dynagen benutzen zu können.

Die Datei „vlab.net" für „dynagen" hat folgendes Format:

```
ghostios = true
sparsemem = true

[localhost]

[[7200]]
npe = npe-400
ram = 160
image   =   /home/labmgmt/DynaLab/images/c7200-advipservicesk9-mz.124-
11.T1.img

[[3640]]
ram=128
image = /home/labmgmt/DynaLab/images/c3640-is-mz.124-13b.img

[[2651XM]]
ram=128
image = /home/labmgmt/DynaLab/images/c2600-ipbasek9-mz.124-13b.img

[[Router Cisco_ID_01]]
model = 3640
console = 2001
fa0/0 = Cisco_ID_03 fa2/1
fa1/0 = Cisco_ID_08 fa0/0

[[Router Cisco_ID_02]]
model = 7200
console = 2002
fa0/0 = Cisco_ID_03 fa2/2

[[Router Cisco_ID_03]]
model = 3640
console = 2003
slot1 = NM-1FE-TX
slot2 = NM-16ESW
f0/0 = NIO_linux_eth:eth1

[[Router Cisco_ID_04]]
model = 2651XM
console = 2004
fa0/0 = Cisco_ID_03 fa2/4
fa1/0 = Cisco_ID_05 fa0/0
autostart = False

[[Router Cisco_ID_05]]
model = 2651XM
console = 2005
autostart = False

[[Router Cisco_ID_08]]
model = 2651XM
console = 2008
idlepc = 0x811ad4d4
```

[26] http://dynagen.org/tutorial.htm#_Toc165530750

Die Konfigurationen in „vlab.net“ und in den „.nvram“ Dateien der virtuellen Router erzeugen diese Topologie:



C3: C3640 Router mit Catalyst-Modul

Abbildung 28: Netz-Topologie Virtuelle Router

Durch die Bridge können die virtuellen Router mit dem Linux Kernel und Geräten im LAN kommunizieren, sofern das Routing innerhalb des V-Lab, der Linux Kernels (durch „vlab.routes“) und gegebenfalls des LAN entsprechend eingerichtet wurde.

8 Anhang B

8.1 Beispiel eines konsolidierten Gerätezustands

```
!RANCID-CONTENT-TYPE: cisco
!
!Chassis type: 3640 - a 3600 router
!CPU: R4700, R4700 CPU at 100MHz, impl 33, Rev 1.2
!
!Memory: main 124928K/6144K
!Memory: nvram 125K
!
!Processor ID: 00000000
!
!Power: Redundant Power System is  present.
!
!Image: Software: C3640-IS-M, 12.4(13b), RELEASE SOFTWARE (fc3)
!Image: Compiled: Tue 24-Apr-07 20:31 by prod_rel_team
!Image: tftp://255.255.255.255/unknown
!
!ROM Image: Version 12.4(13b), RELEASE SOFTWARE (fc3)
!
!
!
!Flash: System flash directory:
!Flash: File  Length   Name/status
!Flash:   1   840      vlan.dat
!Flash: [8388604 bytes used, 0 available, 8388604 total]
!Flash: 8192K bytes of processor board System flash (Read/Write)
!
!Flash: nvram: Directory of nvram:/
!Flash: nvram:   123  -rw-        1484                    <no date>  startup-config
!Flash: nvram:   124  ----           5                    <no date>  private-config
!Flash: nvram:     1  -rw-           0                    <no date>  ifIndex-table
!Flash: nvram: 129016 bytes total (126451 bytes free)
!
!Interface: FastEthernet0/0, AMD Am79c971
!Interface: FastEthernet1/0, AMD Am79c971
!
!Slot 0: hvers 1.0 rev B0
!Slot 0: part 800-03490-01, serial 7720321
!
!Slot 1: hvers 1.0 rev B0
!Slot 1: part 800-03490-01, serial 7720321
!
!Slot 2: type FastEthernet, 16 ports
!Slot 2: hvers 1.0 rev E0
!Slot 2: part 800-15156-01, serial 00000000000
!
!NAME: "3640 chassis",          DESCR: "3640 chassis"
!VID: 0xFF
!SN: 00000000
!NAME: "3640 Chassis Slot 0",   DESCR: "3640 Chassis Slot"
!NAME: "One port Fastethernet TX", DESCR: "One port Fastethernet TX"
!PID: NM-1FE-TX=
!VID: 1.0
!SN: 7720321
!NAME: "FastEthernet0/0",       DESCR: "AmdFE"
!NAME: "3640 Chassis Slot 1",   DESCR: "3640 Chassis Slot"
!NAME: "One port Fastethernet TX", DESCR: "One port Fastethernet TX"
!PID: NM-1FE-TX=
!VID: 1.0
!SN: 7720321
!NAME: "FastEthernet1/0",       DESCR: "AmdFE"
!NAME: "3640 Chassis Slot 2",   DESCR: "3640 Chassis Slot"
!NAME: "16 Port 10BaseT/100BaseTX EtherSwitch", DESCR: "16 Port 10BaseT/100BaseTX EtherSwitch"
!PID: NM-16ESW=
!VID: 1.0
!SN: 000000000000
!NAME: "FastEthernet2/0",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/1",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/2",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/3",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/4",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/5",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/6",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/7",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/8",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/9",       DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/10",      DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/11",      DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/12",      DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/13",      DESCR: "Fast Ethernet"
```

```
!NAME: "FastEthernet2/14",      DESCR: "Fast Ethernet"
!NAME: "FastEthernet2/15",      DESCR: "Fast Ethernet"
!NAME: "3640 Chassis Slot 3",   DESCR: "3640 Chassis Slot"
!
!VTP: VTP Version                      : 2
!VTP: Configuration Revision           : 1
!VTP: Maximum VLANs supported locally : 256
!VTP: Number of existing VLANs         : 9
!VTP: VTP Operating Mode               : Server
!VTP: VTP Domain Name                  :
!VTP: VTP Pruning Mode                 : Disabled
!VTP: VTP V2 Mode                      : Disabled
!VTP: VTP Traps Generation             : Disabled
!VTP: MD5 digest                       : 0x1C 0x82 0x83 0xD1 0x82 0x3A 0x36 0x8B
!VTP: Local updater ID is 130.149.159.1 on interface Vl2 (lowest numbered VLAN interface
found)
!
!VLAN: VLAN Name                              Status    Ports
!VLAN: ---- -------------------------------- --------- -------------------------------
!VLAN: 1    default                               active    Fa2/0, Fa2/3, Fa2/5, Fa2/6, Fa2/7,
Fa2/8, Fa2/9, Fa2/10, Fa2/11, Fa2/12
!VLAN:                                                    Fa2/13, Fa2/14, Fa2/15
!VLAN: 2    VLAN2                             active
!VLAN: 4    VLAN4                             active
!VLAN: 40   VLAN40                            active
!VLAN: 50   VLAN50                            active    Fa2/4
!VLAN: 1002 fddi-default                      active
!VLAN: 1003 token-ring-default                active
!VLAN: 1004 fddinet-default                   active
!VLAN: 1005 trnet-default                     active
!VLAN: VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
!VLAN: ---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
!VLAN: 1    enet  100001     1500  -      -      -        -    -        1002   1003
!VLAN: 2    enet  100002     1500  -      -      -        -    -        0      0
!VLAN: 4    enet  100004     1500  -      -      -        -    -        0      0
!VLAN: 40   enet  100040     1500  -      -      -        -    -        0      0
!VLAN: 50   enet  100050     1500  -      -      -        -    -        0      0
!VLAN: 1002 fddi  101002     1500  -      -      -        -    -        1      1003
!VLAN: 1003 tr    101003     1500  1005   0      -        -    srb      1      1002
!VLAN: 1004 fdnet 101004     1500  -      -      1        ibm  -        0      0
!VLAN: 1005 trnet 101005     1500  -      -      1        ibm  -        0      0
!
!
config-register 0x2102
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Cisco_ID_03
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$IJy2$hxIzPICwwgvuwXolgt5tq1
!
no aaa new-model
memory-size iomem 5
!
ip cef
!
interface Loopback0
 description Yeah.
 ip address 192.168.0.100 255.255.255.255
!
interface FastEthernet0/0
 ip address dhcp
 speed auto
 half-duplex
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/0
!
interface FastEthernet2/1
 switchport mode trunk
!
interface FastEthernet2/2
 switchport mode trunk
!
interface FastEthernet2/3
!
interface FastEthernet2/4
 switchport access vlan 50
!
interface FastEthernet2/5
```

```
!
interface FastEthernet2/6
!
interface FastEthernet2/7
!
interface FastEthernet2/8
!
interface FastEthernet2/9
!
interface FastEthernet2/10
!
interface FastEthernet2/11
!
interface FastEthernet2/12
!
interface FastEthernet2/13
!
interface FastEthernet2/14
!
interface FastEthernet2/15
!
interface Vlan1
 no ip address
!
interface Vlan2
 ip address 192.168.1.1 255.255.255.248
!
interface Vlan40
 ip address 192.168.1.66 255.255.255.240
!
router eigrp 1
 network 192.168.1.0
 network 192.109.39.0
 network 192.168.0.0
 no auto-summary
!
ip http server
!
control-plane
!
line con 0
line aux 0
line vty 0 4
 password vlab
 login
!
end
```

8.2 Glossar

APT	Advanced Packaging Tool
ATM	Asynchronous Transfer Mode
ATOM	Atom Syndication Format
CDP	Cisco Discovery Protocol
CLI	Command Line Interface
CVS	Concurrent Versioning System
CM	Configuration Management
CPU	Central Processing Unit
CGI	Common Gateway Interface
DAV	Distributed Authoring and Versioning
DSL	Digital Subscriber Line
FTP	File Transfer Protocol
FTPS, SFTP	Secure File Transfer Protocols
GPL	GNU Public License
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
HTTPS	Secure Hypertext Transport Protocol
ITIL	IT Infrastructure Library
IPSEC	Internet Protocol Security
IT	Information Technology
IP	Internet Protocol
IOS	Cisco Systems Internet Operating System
LAN	Local Area Network
MIB	Management Information Base
PDU	Power Distribution Unit
POS	Packet Over SONET
RADIUS	Remote Authentication Dial In User Service
RANCID	Really Awesome New Cisco confIg Differ
RFC	Request For Comment
RSS	Really Simple Syndication
SCP	Secure Copy
SSH	Secure Shell
SSL	Secure Socket Layer
SVN	Subversion
TACACS	Termial Access Controller Access-Control System
TFTP	Trivial File Transfer Protocol
TCP	Transmission Control Protocol

TCL	Tool Command Language
URL	Universal Ressource Locator
VPN	Virtual Private Network
VLAN	Virtual Local Area Network
WLAN	Wireless Local Area Network
WWW	World Wide Web
XML	Extensible Markup Language
XSLT	Extensible Style-Sheet Language Tranformation

8.3 Weiterführende Literatur

Im Internet finden sich eine Reihe von nützlichen Dokumenten. Hier eine Auswahl.

Thema	**Quellen**
„Version Control with Subversion“	http://svnbook.red-bean.com/
„Subversion Frequently Asked Questions“	http://subversion.tigris.org/faq.html
„Version Management with CVS“	http://ximbiot.com/cvs/manual
„The Insurrection Web Toosl for Subversion“ Installation, Internals, Source Code	http://svn.code-host.net/svn/Insurrection/trunk/
„XSH – XML Editing Shell“	http://xsh.sourceforge.net/
„GNU Screen“ Virtual VT100 Terminal Manager	http://www.gnu.org/software/screen/
„Net::CDP - advertiser/listener for the Cisco Discovery Protocol“	http://search.cpan.org/~mchapman/Net-CDP-0.09/lib/Net/CDP.pm
„DynaGUI frontend to DynaMIPS“	http://dynagui.sourceforge.net/
„GNS3 - Graphical Network Simulator“	http://www.gns3.net/
„7200emu Forum“	http://7200emu.hacki.at/
„RANCID Frequently Asked Questions“	http://www.shrubbery.net/rancid/FAQ
(Cisco IOS Internals and exploits)	http://www.phenoelit-us.org/
„TACACS+ Daemon“	http://www.shrubbery.net/tac_plus/
„The Cisco-centric Open Source Community“	http://cosi-nms.sourceforge.net/
„NetOps Device Management Tools“	http://sourceforge.net/projects/netops/
„The NOC Project – Operation Support System“	http://trac.nocproject.org/trac/
Cisco IOS „Configuration Replacement and Configuration Rollback“	http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_config-rollback_ps6350_TSD_Products_Configuration_Guide_Chapter.html
„Cisco IOS Resilient Configuration“	http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_resil_config.html

8.4 Lizenzen

This installation contains different software packages, the main components have the following licenses.

THIS LIST IS NOT EXHAUSTIVE, PLEASE MAKE SURE YOU ADHERE TO ALL LICENSES.

Debian:
License: mixed licenses
License text: http://www.debian.org/legal/licenses/
Web site: http://www.debian.org/

Subversion:
License: Individual license
License text: http://subversion.tigris.org/license-1.html
Web site: http://subversion.tigris.org/

Insurrection:
License: GPL
License text: http://www.gnu.org/licenses/licenses.html#GPL
Web site: http://insurrection.tigris.org/

RANCID:
License: Individual license
License text: http://www.shrubbery.net/rancid/LICENSE.txt
Web site: http://www.shrubbery.net/rancid/

XML::XSH2:
License: Perl Artistic License
License text: http://search.cpan.org/src/PAJAS/XML-XSH2-2.1.1/Artistic
Web site: http://search.cpan.org/dist/XML-XSH2/

xmlstarlet:
License: MIT
License text: http://xmlstar.sourceforge.net/license.php
Web site: http://xmlstar.sourceforge.net/

Net::CDP
License: GPL
License text: http://search.cpan.org/src/MCHAPMAN/Net-CDP-0.09/README
Web site: http://search.cpan.org/src/MCHAPMAN/Net-CDP-0.09/

DynaMIPS:
License: GPL
Web site: http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator

DynaGEN:
License: GPL
Web Site: http://dynagen.org/

8.5 Quellen

VMware: Various, VMware, Inc. , Virtual Machine Software, 2007, http://www.vmware.com/

DEBIAN: Linus Torvalds et al, SPI , Debian Linux, 2007, http://www.debian.de/

RANCID: Kilmer et al, Shrubbery Networks, Inc. , RANCID, 2007, http://www.shrubbery.net/rancid/

CURL: Community, cURL groks URLs, 2008

SVN: Karl Fogel et al, Tigris.org , Subversion, 2007, http://subversion.tigris.org/

CVS: Brian Berliner et al, Free Software Foundation , Concurrent Versions System, 2007, http://de.wikipedia.org/wiki/Concurrent_Versions_System

IOS: Various, Cisco Systems, Inc. , Cisco Internetwork Operating System, 2007, http://www.cisco.com/en/US/products/sw/iosswrel/products_ios_cisco_ios_software_category_home.html

APC: APC Corp., American Power Conversion Corp. , Switched PDU, 2007, http://www.apcc.com/products/family/index.cfm?id=70

INSUR: Michael Sinz, MKsoft , Insurrection, 2007, http://insurrection.tigirs.org/

DYNAMIPS: Christophe Fillot, University of Technology of Compiegne , Cisco Simulator, 2007, http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator

DYNAGEN: Greg Anuzelli, Greg Anuzelli , DynaMIPS Network Configuration Generator, 2007, http://dynagen.org/

Apache: Various, Apache Software Foundation , Apache HTTP Server, 2007, http://httpd.apache.org/

8.6 Dateiformate

In diesem Dokument werden die neuen XML Formate erläutert. Das alte Format basierte auf Doppelpunkt-getrennten Feldern und Shell Variablen. Um die Integration der Subsysteme zu ermöglichen wurden die Formate geändert, erweitert und die Programme daran angepasst.

Weitere Informationen sind unter folgender Adresse zu finden:

http://www.beta-plattform.de/softwarerepository/packages/8

8.6.1 Format von .conf.xml (ehemals „rancid.conf“)

Für jeden Mandanten gibt es eine separate „.conf.xml“ Datei, z.B. „bp.conf.xml“. Beim Anlegen eines neuen Mandant mit dem GUI wird das Template „/home/labmgmt/rancid/_template.conf.xml“ um die mandantenspezifischen Werte ergänzt. Beim Anlegen mit „rancid-cvs“ auf der Kommandozeile muss eine fertige Datei „/home/labmgmt/rancid/*repository*.conf.xml“ manuell angelegt werden.

- Alle <variable> Elemente werden als Shell Environment Variablen exportiert
- Alle <command> Elemente werden als Shell Kommandos ausgeführt
- Das „id“ Attribut muss auf den Namen des Mandanten gesetzt werden
- Alle Parameter sind je Mandant gültig, insbesondere PAR_COUNT, was bei der CPU Belastung beachtet werden sollte
- „MAILDOMAIN“ und die Benutzung von „/etc/aliases“ wurde durch die Elemente <bp:userEmail> und <bp:adminEmail> in „devices.xml“ ersetzt

Element	*Attr.*	*Type*	*Default*	*Synopsis*
<rancidConf>				
	id	String		Name for this repository
<commands>				
<command>umask 022</command>		String		Shell file creation mask
</commands>				
<environment>				
<variable name="MODE"></variable>		String	standalone	If set to "BP" verify device lease with Beta Plattform
<variable name="BASEDIR"></variable>		String	/home/labmgmt/rancid	Working area base directory
<variable name="CVSROOT"></variable>		URL	file:///home/labmgmt/repositories	Version control base directory
<variable name="LOGDIR"></variable>		String	$BASEDIR/$id/logs	RANCID logging directory
<variable name="LIST_OF_GROUPS"></variable>		String	„default"	Device group names **with quotes**
<variable name="DEVICES_URL"></variable>		URL	file:///home/labmgmt/rancid	Base directory of .device.xml files
<variable name="IMAGE_URL"></variable>		URL		Base directory for images
<variable name="TERM"></variable>		String	network	
<variable name="PATH"></variable>		String	/usr/local/bin:/usr/bin: /usr/local/bin:/usr/sbin:/bin	
<variable name="TMPDIR"></variable>		Path	/tmp	
<variable name="RCSSYS"></variable>		String	svn	SVN or CVS for RANCID, but the GUI only supports SVN
<variable name="FILTER_PWDS"></variable>		yes/no	NO	Remove passwords from configs ?

Element	*Attr.*	*Type*	*Default*	*Synopsis*
				Must be „no“ for the deploy feature
<variable name="NOCOMMSTR"></variable>		yes/no	NO	Remove SNMP communities ? Must be „no“ for the deploy feature
<variable name="MAX_ROUNDS"></variable>		1..n	1	Max retries to fetch configs
<variable name="OLDTIME"></variable>		1..n		Time threshold for warnings that configs were not fetched this long
<variable name="PAR_COUNT"></variable>		1..n	5	Maximum number of parallel fetch config processes
<variable name="DEBUG">on</variable>		on/off		Debugging flag
</environment>				
</rancidConf>				

Tabelle 8: Format .conf.xml

8.6.2 Format von devices.xml (früher „router.db")

Version: 2

Für jeden Mandanten wird eine eigene Datei „devices.xml" von „DEVICES_URL" in der Datei „.conf.xml" des Mandanten abgerufen. Sie muss alle notwendigen Gerätedaten enthalten.

- In der Beta-Plattform wird die Datei vom Hardware Repository geladen
- In der Standalone-Installation wird sie standardmässig als „http://localhost/BP/*repository*/devices.xml" über den lokalen Apache von „/home/labmgmt/BP/*repository*/devices.xml" geladen, kann aber von jedem URL den „cURL" beherrscht geladen werden, insbesondere von „file:///".
- Für jede Gerätegruppe wird „devices.xml" separat gespeichert

Namespace Deklaration

```
xsi:schemaLocation="http://www.beta-plattform.de/devices.xsd"
xmlns:bp="http://www.beta-plattform.de"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

Das Schema „devices.xsd" kann vom Software Repository geladen werden:

http://www.beta-plattform.de/softwarerepository/packages/8"

Origin Codes und Werte

Die Gerätedaten stammen aus verschiedenen Quellen. In der Beta-Plattform sind folgende definiert:

- TB: Die Daten kommen von einem Testbed-Betreiber
- BP: Die Daten kommen von der Beta-Plattform
- CM: Interne Nutzung durch das Configuration Management
- User: Vom Mandanten oder dem GUI Administrator gewählt
- Default Werte sind **fett**

Element	***Attri-bute***	***Type***	***Origin***	***Synopsis***
<bp:devices>				Per-user set of allocated devices
	id	String	BP	Repository name for this device set
	version	1..n	BP	Devices XML Spec version
	userEmail	String	User	User's Email for RANCID notifications
	adminEmail	String	User	User's Administrative Email for RANCID notifications
<bp:device>				Per-device elements
	id	1..n	BP	Global device identifier
	version	1..n	BP	Version of information for this device
<bp:comment>		String	User	Free text comment
<bp:hostname>		String	BP	Canonical name of device, not FQDN. Name of the device in the CM GUI and the SVN repositories, so it MUST NOT change. I.e. „TB1_dev2“. A prefix like „TB1_“ can be used for wildcard login credentials, but the entry must have a name so it will be included last in a „cat TB_*“ operation (see below)
<bp:accessURL>		URL	TB	Protocol and (ideally) FQDN / IP and port of device, i.e. „ssh://c1.bla.com:22/“. This will be used to actually login to the device by the CM Scripts and GUI links. The (FQ)DN must be EXACTLY the same as the hostname in the credentials files. Port values in the credentials files supersede. The protocol(s) will always be used per the credentials files, so external GUI users may access through NAT port mappings while the internal CM is using the ports directly.
<bp:loginrcURL>		URL	TB	URL for a file containing the credentials and login procedure details in RANCID .cloginrc format (see below).
<bp:rancidState>		„new“, „added“, „deleted“, **empty**	CM	Internal flag for CM state **MUST be empty in file pulled from <DEVICES_URL>** (i.e. BP)
<bp:deviceType>		RANCID	BP	Specifies vendor of device, i.e. „cisco“, see RANCID documentation at

Element	***Attri-bute***	***Type***	***Origin***	***Synopsis***
		device type		http://www.shrubbery.net/rancid/man/router.db.5.html for all types.
<bp:collectFlag>		„up", „**down**"	User	Defines if a device is active in this device group
<bp:deployAllowed>		„true", „**false**"	TB	Defines if transfer of config TO the device is allowed / possible
<bp:saveImageAllowed>		„true", „**false**"	TB	Defines if transfer of images FROM the device is allowed / possible
<bp:saveImage>		„true", „**false**"	User	Defines if transfer of images FROM the device is desired by the user regularly
<bp:saveImageURL>		URL	TB	Defines storage for images, may be empty if not allowed. Must be a directory. **Subdirectories „URL/CM-group-name" must exist!**
<bp:saveMethod>		„**tar**", „cpio" or a script name	TB	The command to use for saving on UN*X. Assumes GNU versions of tar and cpio and presence of bzip2 in Path. The script variant gets arguments „saveListPath saveExcludeListPath OutputFileName" on the command line, but the save operation to <bp:saveImageURL> must **not** be done by the script.
<bp:saveListPath>		String	TB	Name of file on the device containing all filenames to be saved
<bp:saveExcludeListPath>		String	TB	Name of file on the device containing all filenames **not** to be saved. Not used for cpio method.
<bp:deviceControl>				Optional SNMP power control information for remote outlet on/off Currently tested for APC products only http://www.apcc.com/products/family/index.cfm?id=70
<bp:hostname>		String	TB	Adressable hostname of device control unit (PDU)
<bp:readCommunity>		String	TB	SNMP GET community
<bp:writeCommunity>		String	TB	SNMP WRITE community

Element	***Attri-bute***	***Type***	***Origin***	***Synopsis***
<bp:powerCtlOID>		SNMP OID	TB	Table OID to be indexed by <bp:outlet> to control power of device
<bp:powerOnOP>		SNMP value	TB	Value to set <bp:powerCtlOID> to power on device
<bp:powerOffOP>		SNMP value	TB	Value to set <bp:powerCtlOID> to power off device
<bp:rebootOP>		SNMP value	TB	Value to set <bp:powerCtlOID> to reboot (power cycle) device
<bp:powerStateOID>		SNMP OID	TB	Table OID to be indexed by <bp:outlet> to return device power status
<bp:outlet>		0..n	TB	SNMP table index of device's power outlet
</bp:deviceControl>				
</bp:device>				
</bp:devices>				

Tabelle 9: Format devices.xml

8.6.3 Schema Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Mit XMLSpy v2008 sp1 (http://www.altova.com) von Chris Zander (T-Systems GEI
GmbH) bearbeitet -->
<!--W3C Schema erstellt mit XMLSpy v2008 sp1 (http://www.altova.com)-->
<!--Auf Spec 0.8 angepasst von Heiko Blume-->
<xs:schema xmlns:bp="http://www.beta-plattform.de"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.beta-
plattform.de" elementFormDefault="qualified" attributeFormDefault="qualified"
version="2">
	<xs:element name="writeCommunity" type="xs:string"/>
	<xs:element name="saveImageURL" type="xs:anyURI"/>
	<xs:element name="saveImageAllowed" type="xs:boolean"/>
	<xs:element name="saveImage" type="xs:boolean"/>
	<xs:element name="saveListPath" type="xs:string"/>
	<xs:element name="saveExcludeListPath" type="xs:string"/>
	<xs:element name="rebootOP" type="xs:string"/>
	<xs:element name="readCommunity" type="xs:string"/>
	<xs:element name="rancidState">
		<xs:simpleType>
			<xs:restriction base="xs:string">
				<xs:enumeration value="added"/>
				<xs:enumeration value="collected"/>
				<xs:enumeration value="new"/>
				<xs:enumeration value="missed"/>
				<xs:enumeration value="deleted"/>
				<xs:enumeration value=""/>
			</xs:restriction>
		</xs:simpleType>
	</xs:element>
	<xs:element name="saveMethod">
		<xs:simpleType>
			<xs:restriction base="xs:string">
				<xs:enumeration value="tar"/>
				<xs:enumeration value="cpio"/>
			</xs:restriction>
		</xs:simpleType>
	</xs:element>
	<xs:element name="rancidDevices">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="bp:device" maxOccurs="unbounded"/>
			</xs:sequence>
			<xs:attribute name="id" type="xs:string" use="required"/>
			<xs:attribute name="userEmail" type="xs:string"
				use="required"/>
			<xs:attribute name="adminEmail" type="xs:string"
				use="required"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="powerStateOID" type="xs:string"/>
	<xs:element name="powerOnOP" type="xs:string"/>
	<xs:element name="powerOffOP" type="xs:string"/>
	<xs:element name="powerCtlOID" type="xs:string"/>
	<xs:element name="outlet" type="xs:byte"/>
	<xs:element name="loginrcURL" type="xs:anyURI"/>
	<xs:element name="hostname" type="xs:string"/>
	<xs:element name="deviceType" type="xs:string"/>
	<xs:element name="deviceControl">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="bp:hostname"/>
				<xs:element ref="bp:outlet"/>
				<xs:element ref="bp:readCommunity"/>
				<xs:element ref="bp:writeCommunity"/>
				<xs:element ref="bp:powerStateOID"/>
				<xs:element ref="bp:powerCtlOID"/>
				<xs:element ref="bp:powerOnOP"/>
				<xs:element ref="bp:powerOffOP"/>
				<xs:element ref="bp:rebootOP"/>
			</xs:sequence>
		</xs:complexType>
	</xs:element>
	<xs:element name="device">
		<xs:complexType>
			<xs:sequence>
```

```
                        <xs:element ref="bp:comment"/>
                        <xs:element ref="bp:hostname"/>
                        <xs:element ref="bp:accessURL"/>
                        <xs:element ref="bp:loginrcURL"/>
                        <xs:element ref="bp:rancidState"/>
                        <xs:element ref="bp:deviceType"/>
                        <xs:element ref="bp:collectFlag"/>
                        <xs:element ref="bp:deployAllowed"/>
                        <xs:element ref="bp:saveImageAllowed"/>
                        <xs:element ref="bp:saveImage"/>
                        <xs:element ref="bp:saveImageURL"/>
                        <xs:element ref="bp:saveMethod"/>
                        <xs:element ref="bp:saveListPath"/>
                        <xs:element ref="bp:saveExcludeListPath"/>
                        <xs:element ref="bp:deviceControl"/>
                </xs:sequence>
                <xs:attribute name="version" type="xs:byte" use="required"/>
                <xs:attribute name="id" type="xs:byte" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="deployAllowed" type="xs:boolean"/>
    <xs:element name="comment" type="xs:string"/>
    <xs:element name="collectFlag">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="down"/>
                <xs:enumeration value="up"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="accessURL" type="xs:anyURI"/>
</xs:schema>
```

8.6.4 Format der Device Credentials Dateien

Hier werden die Dateien beschrieben, die mit <bp:loginrcURL> adressiert werden. Sie enthalten die Benutzernamen, Passwörter und andere Informationen, die benötigt werden um auf die Geräte zuzugreifen. Die Dateien können komplex sein.

- Es gibt je Gerät eine Datei
- Fehlende Dateien werden ignoriert. Dateien mit identischen Namen überschreiben zuvor abgerufene. Dies kann für Wildcard-Operationen genutzt werden, oder um alle Daten in einer einzigen Datei zu speichern.
- Alle Dateien werden in eine einzige „.cloginrc" Datei vereinigt. Die Reihenfolge der Einträge ist signifikant. Wildcard Einträge müssen nach spezifischen Einträgen folgen.
- Es gibt eine „include" Direktive mit der lokale Dateien eingebunden werden können
- Die Dateien enthalten TCL Code, es muss auf korrekte Syntax geachtet werden
- Die Gerätenamen müssen zu denen in den <bp:accessURL> Elementen identisch sein.
- Die Methoden „telnet" und „ssh" können beide mit einem Port angegeben werden, z.B. „ssh:10022"

Beispiel für spezifischen Eintrag:

```
add autoenable   TB1_dev2    0
add password     TB1_dev2    {lab23423}     {hdhdh3s}
add method       TB1_dev2    telnet
```

Beispiel für einen Wildcard Eintrag:

```
add autoenable   TB1_*       0
add password     TB1_*       {NsjSU78}     {KkKs882}
add method       TB1_*       ssh
```

Weitere Details unter

http://www.shrubbery.net/rancid/man/cloginrc.5.html