

Dokumentation

Sys**V**entory

inventory your systems

Fächerübergreifende, interdisziplinäre Projektarbeit



IT HF 12-15

1. Inhaltsverzeichnis

1. Inhaltsverzeichnis	2
2. Projektvorfeld	3
2.1. IST- Zustand.....	3
2.2. Aufgabenstellung.....	3
2.3. Anforderungen Soll Zustand.....	3
2.3.1. <i>Funktional</i>	3
2.3.2. <i>Nicht- Funktional</i>	5
3. Projektziel	7
4. Zeitplanung / Meilensteine	8
5. Realisation	9
5.1. Organisation.....	10
5.2. Infrastruktur.....	12
5.2.1. <i>Monitoring</i>	14
5.3. Sammeln der Daten	15
5.3.1. <i>Start Scan</i>	15
5.3.2. <i>Discovery Bash Script</i>	15
5.3.3. <i>Power Shell Script</i>	15
5.3.4. <i>Ablaufdiagramm</i>	17
5.4. Datenbank.....	18
5.4.1. <i>Übersicht</i>	18
5.4.2. <i>Detailansicht - ERM</i>	19
5.5. Applikation	19
5.5.1. <i>Technologien</i>	20
5.5.2. <i>Continuous Integration</i>	21
5.5.3. <i>Komponenten</i>	22
5.5.4. <i>Funktionen</i>	24
6. Fazit	31
7. Anhang.....	32
7.1. Projekttagebuch	32
7.2. ERM.....	34
7.3. Zeitlicher Projektplan	35

2. Projektvorfeld

2.1. IST- Zustand

Unsere Gruppe bestehend aus Nico Florin, Daniel Ammann, Benjamin Rechsteiner, Andreas Gyr und Lukas Stadelmann wurde im Rahmen der FPA (Fächerübergreifende Projekt Arbeit) die Aufgabe übertragen, für das Zentrum für berufliche Weiterbildung etwas zu Entwickeln mit dem die IT Hardware und die darauf installierte Software überwacht werden kann. Es ist zurzeit nicht möglich zentral diese Daten zu erheben. Da es ein sehr grosser Aufwand wäre die ganzen Daten manuell zusammen zutragen wurde dies nicht gemacht. Änderungen an Systemen werden nicht erkannt und dadurch entstehende Probleme können teilweise nur mit grossem zeitlichem Aufwand behoben werden.

2.2. Aufgabenstellung

Es soll ein Tool entwickelt werden welches für das oben beschriebene Problem eine automatisierte Lösung bietet. Das Tool soll sowohl alle im Einsatz stehenden Computer wie auch Veränderungen der Hardware- und Softwarekonfiguration erkennen. Ebenfalls soll Sie zur Erhebung von Statistiken dienen.

Es handelt sich um ein fiktives Projekt.

2.3. Anforderungen Soll Zustand

2.3.1. Funktional

Scannen: Scannen von Computern innerhalb eines bekannten Netzes sowie die Speicherung von folgenden Informationen: Betriebssystem (Name des OS, Version des OS, 32 Bit / 64 Bit) Programme (Installierte Software, Version)

Prozessor (Beschreibung der CPU, Prozessorfamilie, Anzahl Kerne, Taktrate) Drucker (Liste der registrierten Printer, Treiber und Versionsnummer) Computer (SID, Name, IP-Adresse, RAM, Computertyp) NIC (IPV4/IPV6, MAC-Adresse, Bezeichnung, Standard-Gateway, DHCP, Subnet)

Speichern: Die gesammelten Daten werden in einer Datenbank gespeichert.

Historie: Die gesammelten Daten werden versioniert. Es entsteht somit eine Historie von Scan-Ergebnissen, die jederzeit abgerufen werden kann. Zudem sind zwischen einzelnen Versionen die Unterschiede sichtbar zu machen.

Anmelden: Die Applikation kann nur über Benutzernamen und Passwort benutzt werden. Die Applikation umfasst zudem ein einfaches Rollenkonzept, welche zwei Arten von Benutzer/innen unterscheidet: Visitor: Kann Daten in der Applikation betrachten, aber keine Scans durchführen.

Inventor: Kann dasselbe wie Visitor, zudem aber auch Scans durchführen und Daten verwalten.

Suchfunktion: Es ist möglich, mit Hilfe der Applikation nach folgenden Kriterien zu suchen: Computername, IP-Adresse, Typ, Prozessortyp, Betriebssystem Es ist zudem möglich, in den oben erwähnten Kriterien kombiniert zu suchen.

Visualisierung: Die Applikation soll die inventarisierten Computer in einer Liste darstellen, welche nach den Kriterien (siehe Suchfunktion) sortiert werden kann. Zudem können Kriterien (siehe Scannen) ein- / bzw. ausgeblendet werden. Es können aber maximal acht Kriterien gleichzeitig dargestellt werden.

Statistik: Das Statistikmodul gibt zu folgenden Fragen Auskunft: - Anzahl der durchgeführten Scans pro Woche/Monat/Jahr - Anzahl der Änderungen an Systemen pro Woche / Monat / Jahr - Anzahl verschiedener Betriebssysteme - Anzahl verschiedener Softwarepakete

2.3.2. Nicht- Funktional

<i>Technologien:</i>	Es ist ausschliesslich auf JAVA-Technologie sowie/oder Web-Technologien (HTML, JS, PHP) zurück zu greifen.
<i>Datenbank:</i>	Als Datenbank ist MySQL 5.x zu wählen. Es ist nicht notwendig, ein Datenbank- Abstraktionslayer zu implementieren um den Wechsel auf einen anderen Datenbankserver zu vollziehen.
<i>Datenschutz:</i>	Bei den meisten Informationen handelt es sich nicht um sensitive Daten. Daher ist es nicht notwendig, den Datentransport verschlüsseln. Einzig Benutzernamen und Passworte müssen mit einer Einweg-Hash-Funktion gesichert werden.
<i>Verfügbarkeit:</i>	Die Applikation muss rund um die Uhr verfügbar sein. Für die Verfügbarkeit muss eine entsprechende Umgebung realisiert werden. Unterbrüche müssen registriert und zugeteilte Administratoren informiert werden.
<i>Benutzerinterface:</i>	Das Benutzerinterface muss den gängigen Usability-Anforderungen entsprechen und somit einfach und intuitiv zu bedienen sein.
<i>Test:</i>	Die Lösung muss die funktionalen sowie nicht funktionalen Anforderungen erfüllen. Diese müssen bewiesen werden
<i>Support:</i>	Einige fachspezifische Themen sind bewusst nicht im Unterricht vermittelt worden, sondern muss durch die Projektgruppe selbst erarbeitet werden. Unterstützung bieten die Dozenten der angrenzenden Fächer. Die Dozenten dürfen als Berater bei Unklarheiten oder Fragen hinzugezogen werden.
<i>Dokumentation:</i>	Der Projektverlauf mit seinen Teil-Ergebnissen muss genau so in einem Abschlussbericht dokumentiert sein, wie die Lösung der Applikation selbst.

Arbeitsteilung: Das Projektteam muss die Arbeiten auf einzelne Team-Mitglieder aufteilen. Der erwartete Umfang des Resultats geht von der Arbeitsteilung aus. Es sind zwei Schwerpunkte in der Arbeit zu verorten:

Einerseits systemtechnische Aspekte – Bereitstellung der Infrastruktur für Simulation, Beschaffung der Gerätedaten, Bereitstellung der Gerätedaten, applikatorische Umgebung, Datenbank, Backup, Verfügbarkeit

Andererseits, programmiertechnische Aspekte – Übernahme der Gerätedaten, Speicherung und Auswertung der in der Datenbank befindlichen Daten, Visualisierung

3. Projektziel

Es soll ein gemäss Auftrag funktionsfähiger Prototyp abgeliefert werden, welcher transparent dokumentiert ist.

Zusätzlich sollen folgende Dokumente abgeliefert werden:

- Projektplan mit Soll- und Ist Vergleich
- Aufgabenteilung mit Arbeitspaketen
- Projekttagebuch mit Stundenkontrolle
- Architekturmodell nach UML
- Klassendiagramm nach UML

Abgabe Termin ist der 16.06.2015. An diesem Tag wird zu der Abgabe das Projekt in einer rund 15 minütigen Präsentation präsentiert.

4. Zeitplanung / Meilensteine

Folgende Meilensteine wurden durch den Auftraggeber vorgegeben:

Meilenstein	Datum	Beschrieb
MS [TGS]	14.04.2015	Einstieg und thematische Annäherung, Gruppenbildung und Strategie.
MS [PII]	28.04.2015	Präsentation Systeminfrastruktur, Technische Lösung für Informationsbeschaffung.
MS [PDB]	05.05.2015	Präsentation des Datenbankarchitekturmodells.
MS [PIZ]	18.05.2015	Präsentation des aktuellen Ist-Zustands.
MS [PAB]	16.06.2015	Abschlusspräsentation aller Gruppen sowie Abgabe der Arbeit.

Abbildung 1 - Meilensteintabelle

Im Anhang befindet sich die grobe Zeitplanung, welche beim Kick-Off Meeting festgelegt wurde.

5. Realisation

Bevor die eigentliche Realisationsphase beginnen konnte musste das Projektteam durch eine für das Projekt sehr wichtige Planungsphase, in der passende Technologien sowie die Hardwareumgebung definiert wurden. Ebenso wurden an diesen Planungsmeetings die Arbeitspakete der jeweiligen Teammitglieder geschnürt.

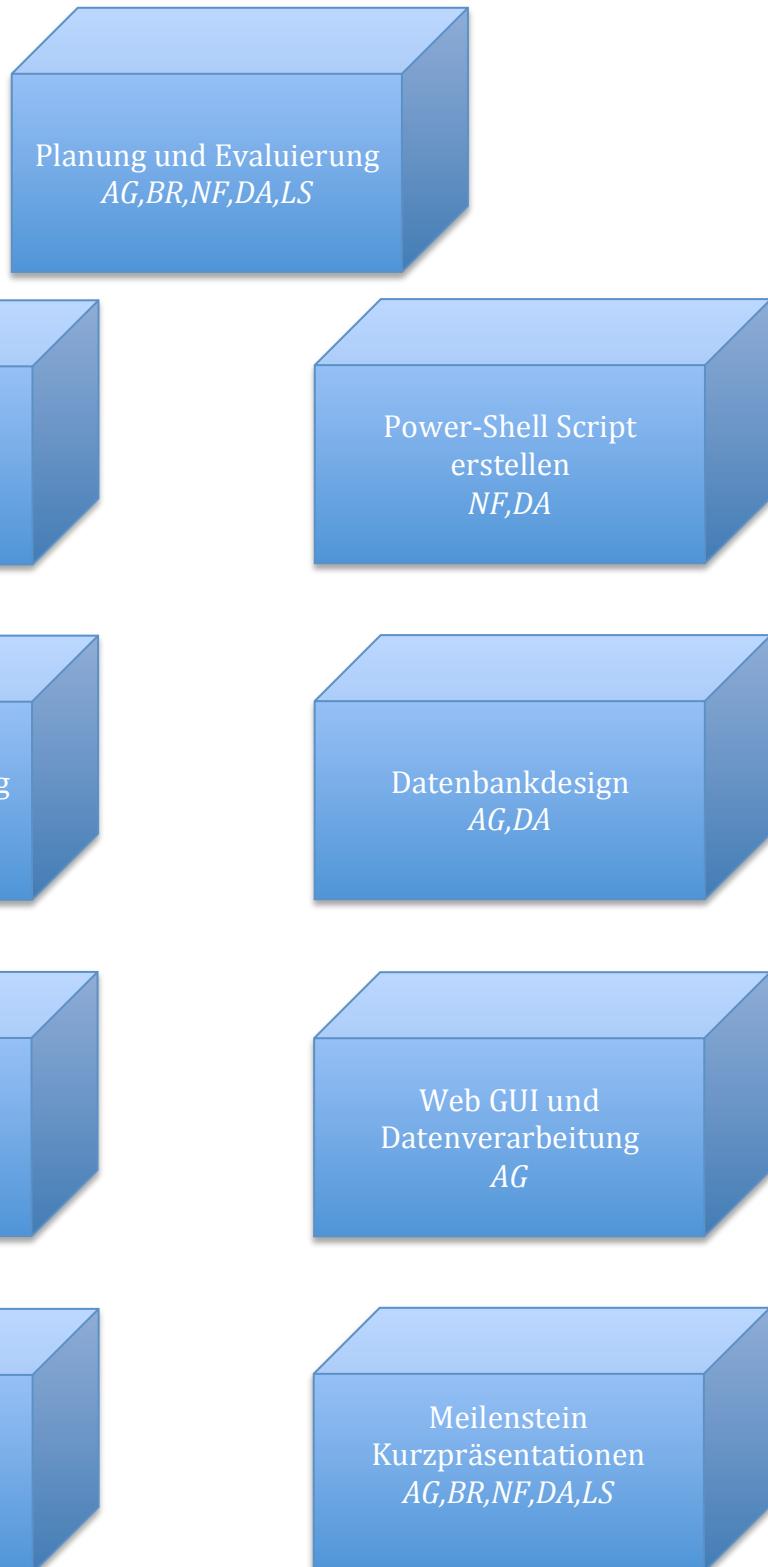
BR Benjamin Rechsteiner

DA Daniel Ammann

AG Andreas Gyr

NF Nico Florin

LS Lukas Stadelmann



5.1. Organisation

Das Projektteam hat sich jeweils in den dafür zur Verfügung gestellten Lektionen über die folgenden Arbeitsschritte ausgetauscht. Zudem wurden alle Dateien welche für das erreichen des Ziels nötig waren stets aktuell und für alle einsehbar auf *Seafile* zur Verfügung gestellt.

Die Daten für den Zugriff auf die Komponenten sind ebenfalls auf *Seafile* abgelegt.

Git

- <https://github.com/Cogax/SysVentryServer>
- Das Repo (prod Branch) wird einmal pro Minute im DocumentRoot (/srv/www) des Apaches gepullt.
- <https://travis-ci.org/Cogax/SysVentryServer> Build Server
- master Branch wird gebaut, wenn erfolgreich wird in den prod branch gepusht
- Hier noch ein Git-Repo für den Collector:
- git clone <root@172.17.82.22:/srv/git/sysventory-collector.git>

Authentication

- Dev-Environment: OpenVPN/SeafilePassword
- Windows Server: RDP administrator/PASSWORT
- Windows Client: RDP test/PASSWORT
- Linux Webserver: SSH root/PASSWORT
- Mysql: mysql (3306) root/PASSWORT

Monitoring

- Software: Monit
- Login: admin/PASSWORT
- Mysql-Server: <http://172.17.82.21:2812>
- Web-Server: <http://172.17.82.22:2812>

Networks

transfer01-01 (WAN)

- IPv4: 178.250.135.18/29
- IPv6: 2a02:16a0:ff02:1::18/64

mngt01-01

- IPv4: 172.17.81.0/24
- GWv4: 172.17.81.1
- IPv6: 2a02:16a0:ff02:81::0/64
- GWv6: 2a02:16a0:ff02:81::1

maintenance01-01

- IPv4: 172.17.82.0/24
- GWv4: 172.17.82.1
- IPv6: 2a02:16a0:ff02:82::0/64
- GWv6: 2a02:16a0:ff02:82::1

VM's

mysqldev01-01

- Type: Database
- OS: Debian 7.8
- IPv4: 172.17.82.21
- IPv6: 2a02:16a0:ff02:82::21

webdev01-01

- Type: Website
- OS: Debian 7.8
- IPv4: 172.17.82.22
- IPv6: 2a02:16a0:ff02:82::22

addev01-01

- Type: Active Directory
- OS: Windows Server 2012 R2 DC
- IPv4: 172.17.82.41
- IPv6: 2a02:16a0:ff02:82::41

colldev01-01

- Type: sysventory collector
- OS: Windows Server 2012 R2 DC
- IPv4: 172.17.82.42
- IPv6: 2a02:16a0:ff02:82::42

client01-01

- Type: clinet
- OS: Windows 7 x64
- IP: 172.17.82.101

client01-02

- Type: clinet
- OS: Windows 7 x64
- IP: 172.17.82.102

jenkinsdev01-01

- Type: Jenkins
- OS: Debian 7.8
- IPv4: 172.17.82.12
- IPv6: 2a02:16a0:ff02:82::12

5.2. Infrastruktur

Unsere Infrastruktur umfasst einen Supermicro X10SL7-F mit folgenden Spezifikationen:

Intel CPU E3-1230 v3
32GB DDR3 RAM
2x 800GB Intel SSD DC S3500

Sowie einen Cisco Catalyst 3750G – 24TS.

Diese Hardware- Infrastruktur wurde uns von der Firma Automatic Server AG zur Verfügung gestellt. Hierfür bedankt sich das Projektteam herzlichst.
Auf das System konnte das Projektteam jeder Zeit per OpenVPN zugreifen.

Nachfolgende Grafik zeigt welche Betriebssysteme im zu welchem Zweck verwendet werden.

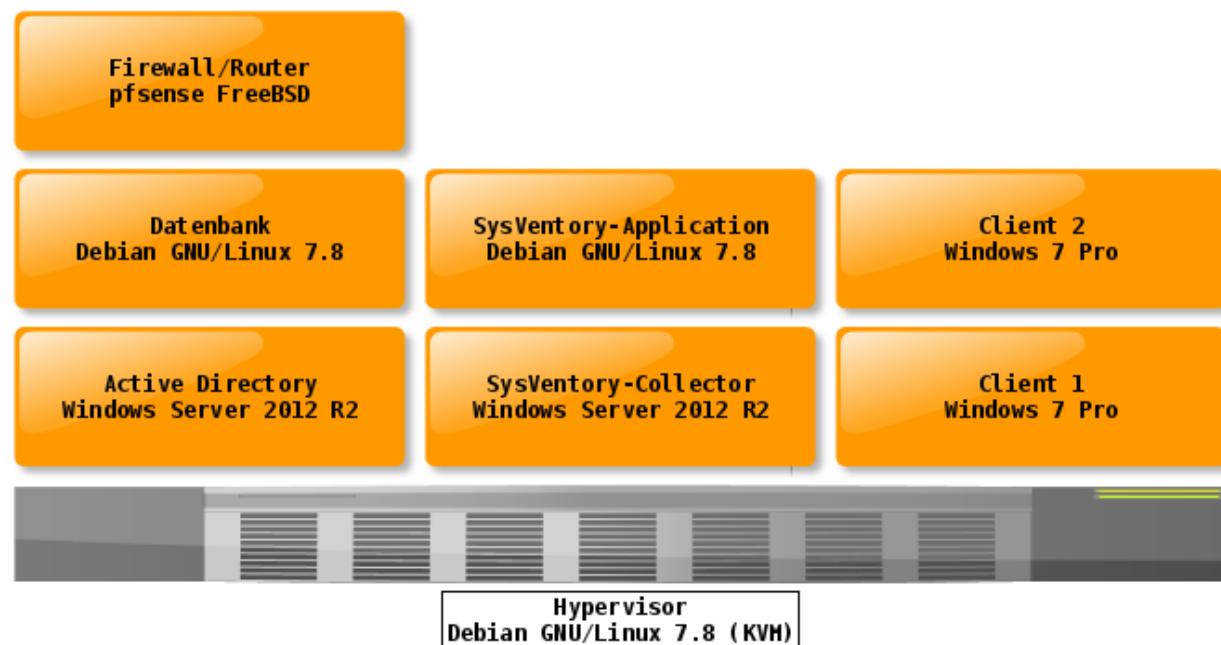


Abbildung 2 – Verwendete Betriebssysteme (Virtualisiert)

Auf dem Bild auf der nächsten Seite sind die einzelnen Komponenten inklusive der Verbindungen und deren Protokolle zu sehen.

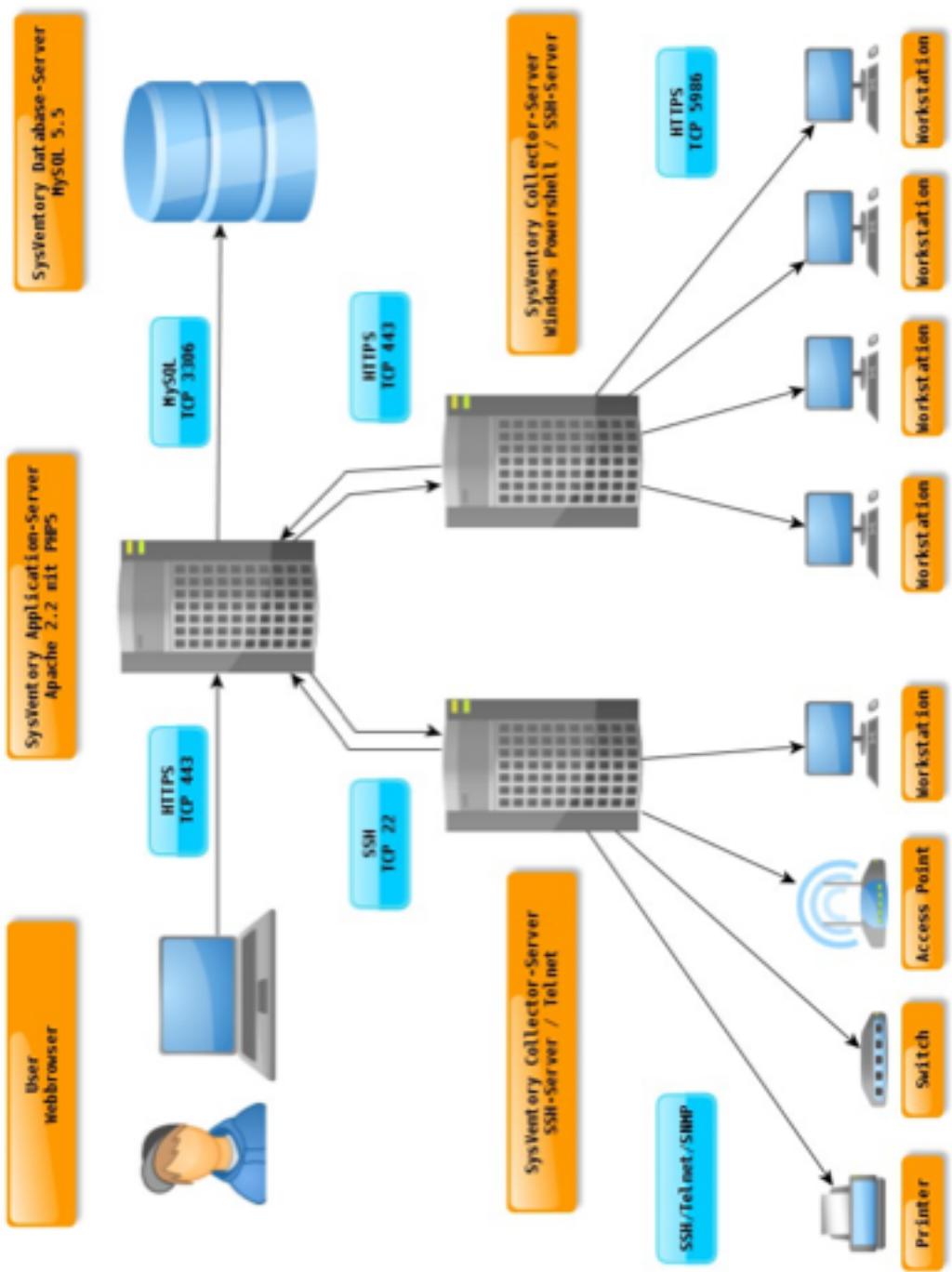


Abbildung 3 – Gesamte Infrastruktur

5.2.1. Monitoring

Gemäss dem Auftrag wird verlangt, dass die Applikation rund um die Uhr verfügbar sein muss. Um das zu bewerkstelligen, haben wir uns für das freie Monitoring-Tool Monit entschieden. Monit ist ein einfaches und kleines C-Programm, das über eine Textdatei konfiguriert werden kann. Mit diesem leicht verständlichen Syntax können Dateien, Verzeichnisse, Dienste und Prozesse überwacht werden. Als Event können verschiedenen Aktionen ausgeführt werden, wie z.B. Mailversand, Dienste neu starten oder Scripte ausführen. Wir haben unsere Systeme so eingerichtet das der Webserver Apache und die Mysql-Datenbank bei einem Ausfall automatisch neu gestartet werden und eine Mail an den Administartor versendet wird.

Process status

Parameter	Value
Name	mysqld
Pid file	/run/mysqld/mysqld.pid
Status	Running
Monitoring mode	active
Monitoring status	Monitored
Start program	'/etc/init.d/mysql start' timeout 30 second(s)
Stop program	'/etc/init.d/mysql stop' timeout 30 second(s)
Existence	If doesn't exist 1 times within 1 cycle(s) then restart else if succeeded 1 times within 1 cycle(s) then alert
Data collected	Mon, 15 Jun 2015 19:54:22
Process id	6106
Parent process id	5708
Process uptime	43d 5h 53m
Children	0
CPU usage	0.0% (Usage / Number of CPUs)
Total CPU usage (incl. children)	0.0%
Memory usage	7.6% [78448kB]
Total memory usage (incl. children)	7.6% [78448kB]
Pid	If changed 1 times within 1 cycle(s) then alert
Ppid	If changed 1 times within 1 cycle(s) then alert

Start service

Stop service

Restart service

Disable monitoring

Abbildung 4 – Monit

5.3. Sammeln der Daten

5.3.1. Start Scan

Das Sammeln der Daten ist bewusst sehr modular aufgebaut. In der Umgebung des ZBW sind nur Windows Clients vorhanden. Es wäre aber ohne grösseren Aufwand für andere Betriebssysteme, wie Linux, erweiterbar. Den Start bildet das Web-GUI, bzw. den automatischen cron Job. Dort kann ein Netzwerk-Range oder eine einzelne IP-Adresse angegeben werden und durch ein PHP-Command wird dann das Discovery Script auf dem Webserver gestartet.

/srv/script/discovery.sh

The screenshot shows a web-based interface titled "Scan erfassen". It has a text input field labeled "Netzwerk Range:" with a placeholder "192.168.1.0/24". Below it is a checkbox labeled "Netzwerk speichern". At the bottom is a blue button labeled "Scan".

Abbildung 5 - Erfassen eines Scans im Web-GUI

5.3.2. Discovery Bash Script

Dieses Script „discovery.sh“ wird auf dem Webserver gestartet. Übergabe-Parameter sind der Netzwerk-Range, bzw. eine IP-Adresse, sowie Username und Passwort.

Der Befehl sieht folgendermassen aus:

/srv/script/discovery.sh

Das Script scannt anschliessend alle IP-Adressen in dem Netzwerk mittels nmap und notiert sich alle Geräte die eine Rückmeldung geben. Die Liste der IPs der gescannten Geräte wird dann an das Power Shell-Script auf dem Collector-Server per SSH Befehl übergeben.

5.3.3. Power Shell Script

Das Script „start_collect.ps1“ wird auf dem Collector-Server gestartet. Übergabe-Parameter sind eine IP-Liste, bzw. eine IP-Adresse, sowie Username und Passwort.

Ein Befehl könnte z.B wie folgt aufgebaut sein:

```
.\\start_collect.ps1 -ServerList 172.17.82.101-172.17.82.102 -  
Username zbw\\administrator -Password Password
```

Für jede übergebene IP-Adresse werden nun die Client-Daten gesammelt. Dies geschieht über die von Windows Power Shell zur verfügbar gestellten CIM Cmdlets. Mittels CIM (Common Information Model) können diverse Informationen eines Remote-Computers ausgelesen werden. Diese Daten werden dann sauber strukturiert pro Client in einer XML Datei gespeichert.

Anschliessend werden alle XML Dateien ausgelesen und mittels http POST Request an PHP-Script auf dem Webserver gesendet, welcher die Dateien parsed und in den Inhalt in die Datenbank schreibt.

5.3.3.1. Beispiel XML Datei

```
<?xml version="1.0" encoding="UTF-8"?>
<composition>
    <machine>
        <manufacturer>Bochs</manufacturer>
        <model>Bochs</model>
        <serial_number></serial_number>
        <memory>2147069952</memory>
        <computer_name>CLIENT01-01</computer_name>
        <uuid>79B60242-1F48-F7E9-8169-FB15DFB0B212</uuid>
    </machine>
    <operating_system>
        <name>Microsoft Windows 7 Professional </name>
        <version>6.1.7601</version>
        <service_pack>1</service_pack>
        <architecture>64-Bit</architecture>
    </operating_system>
    <softwares>
        <software>
            <name>Microsoft .NET Framework 4 Extended</name>
            <version>4.0.30319</version>
        </software><software>
            <name>Microsoft .NET Framework 4 Client Profile</name>
            <version>4.0.30319</version>
        </software>
    </softwares>
    <cpus>
        <cpu>
            <name>Intel Xeon E312xx (Sandy Bridge)</name>
            <family>Intel64 Family 6 Model 42 Stepping 1</family>
            <clock>3300</clock>
            <cores>1</cores>
        </cpu>
    </cpus>
    <network_interfaces>
        <network_interface>
            <description>Intel(R) PRO/1000 MT-Netzwerkverbindung</description>
            <ipv4>172.17.82.101</ipv4>
            <ipv4_subnet>255.255.255.0</ipv4_subnet>
            <ipv4_default_gateway>172.17.82.1</ipv4_default_gateway>
            <ipv6>2a02:16a0:ff02:82:44d0:53b2:4010:b3dc</ipv6>
            <ipv6_subnet>64</ipv6_subnet>
            <ipv6_default_gateway>fe80::5054:ff:fe6d:bd37</ipv6_default_gateway>
            <dhcp>0</dhcp>
            <mac>52:54:00:9C:30:73</mac>
        </network_interface>
    </network_interfaces>
    <printers>
        <printer>
            <name>Microsoft XPS Document Writer</name>
            <driver>Microsoft XPS Document Writer</driver>
            <version>2.1.0.0</version>
        </printer><printer>
            <name>Fax</name>
            <driver>Microsoft Shared Fax Driver</driver>
            <version>1.0.0</version>
        </printer>
    </printers>
</composition>
```

5.3.4. Ablaufdiagramm

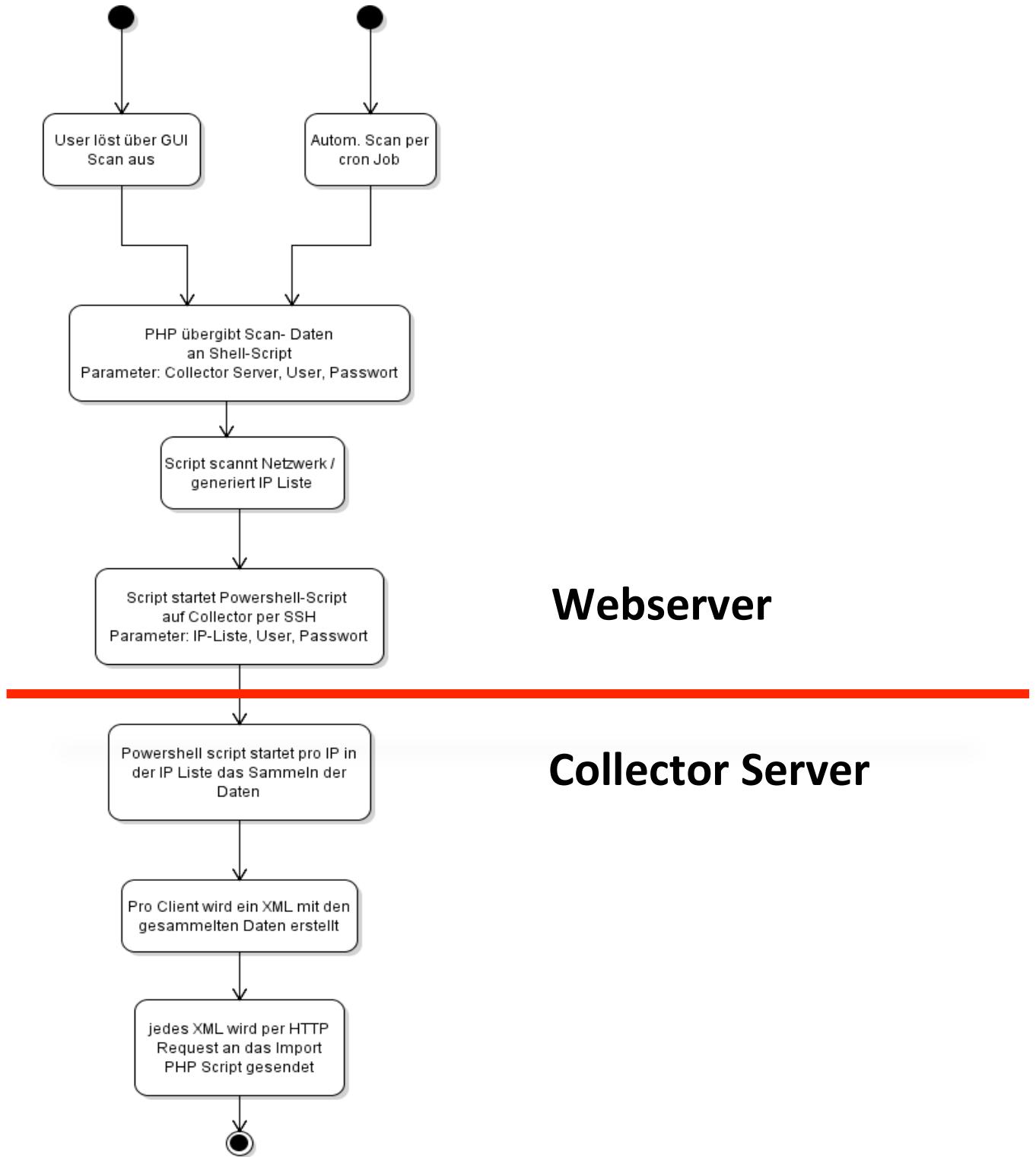


Abbildung 6 - Ablaufdiagramm

5.4. Datenbank

Für das Speichern der Daten wird auf eine MySQL 5.5-Installation auf einem Linuxserver zurückgegriffen.

Diese Datenbankumgebung wird in diesem Kapitel im Detail beschrieben.

5.4.1. Übersicht

Eine Übersicht über die Datenbankstruktur kann man sich am einfachsten anhand des Entity Diagrams verschaffen:

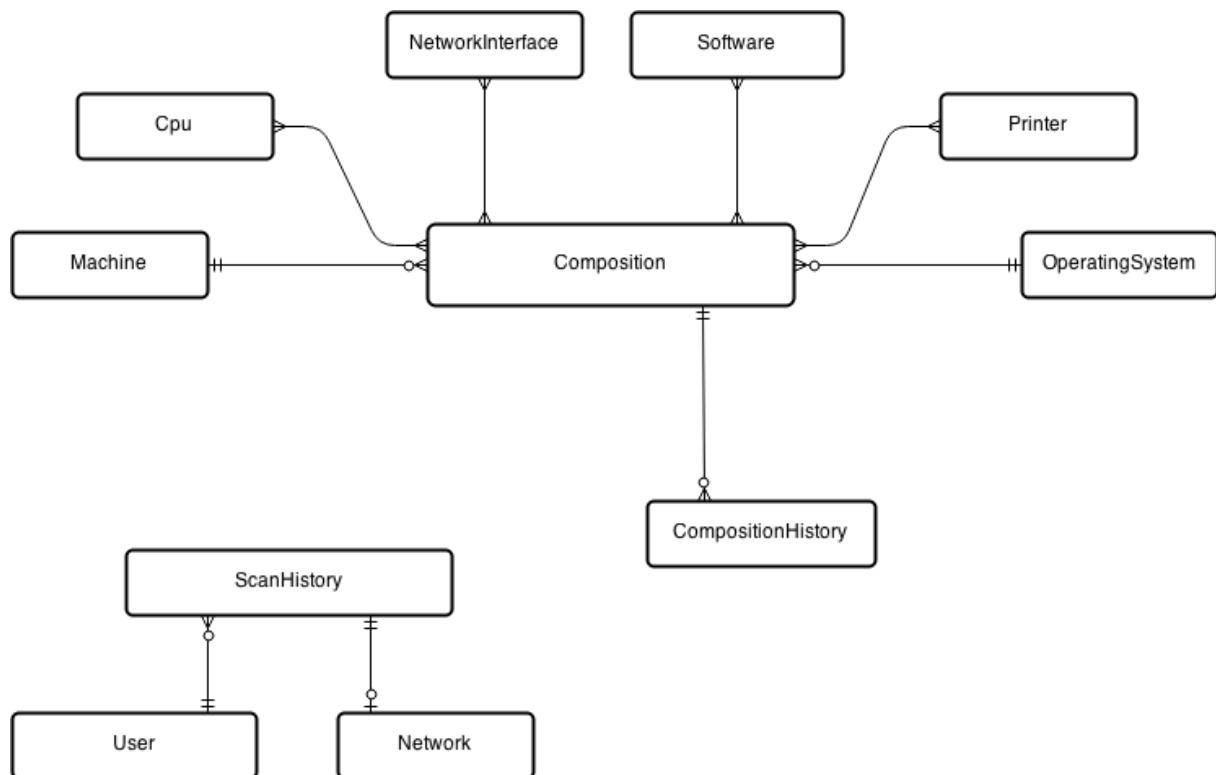


Abbildung 7 - Datenbankstruktur

Hier wird ersichtlich, dass für jeden Computer ein Eintrag in der Tabelle *Composition* erstellt wird. Um die gängige Praxis der Normalisierung umzusetzen, werden die einzelnen Eigenschaften eines Computers, wie z.B. die CPU oder die Printer, in weitere Tabellen abgespeichert und entsprechend Verknüpft.

Des Weiteren werden auch die einzelnen Scavorgänge, inkl. Informationen zum Benutzer, der den Scavorgang gestartet hat, sowie die IP-Adresse(n) des Scans, sowie natürlich ein Zeitstempel, in weitere Tabellen gespeichert.

5.4.2. Detailansicht – ERM

Das ERM bietet einen detaillierten Überblick über die einzelnen Tabellen und deren Verknüpfungen.

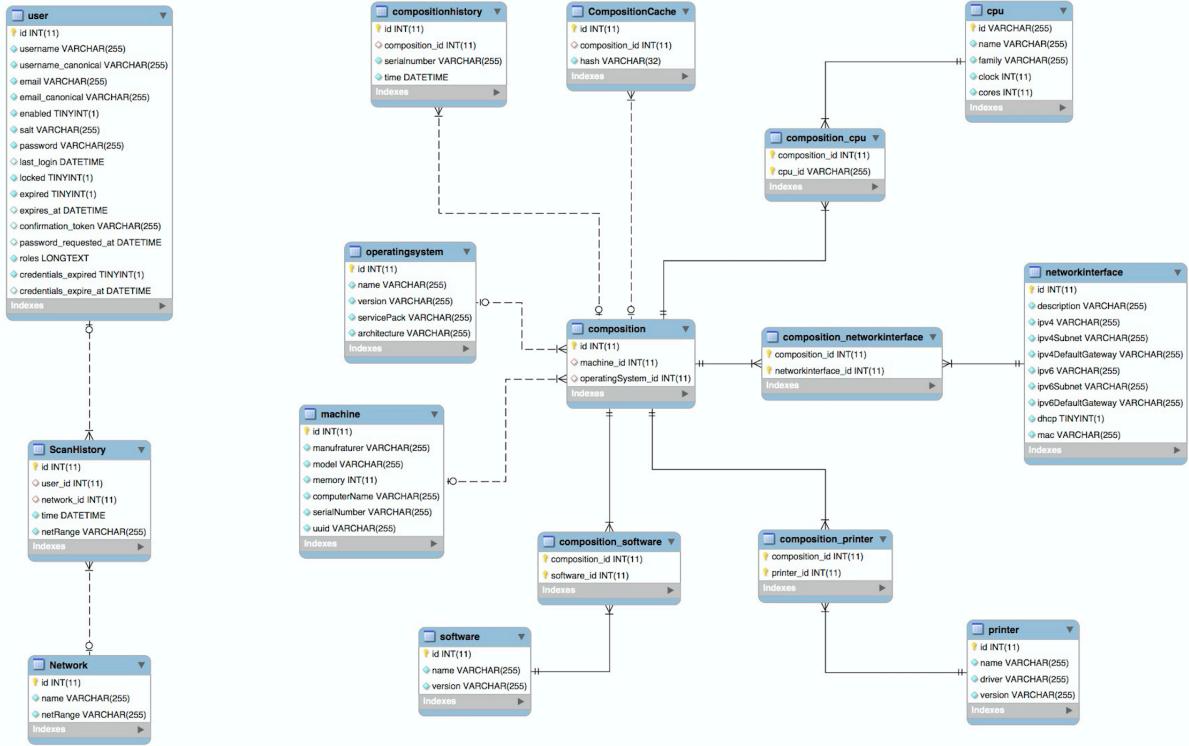


Abbildung 8 - ERM

Das ERM ist in voller Auflösung im Anhang enthalten.

5.5. Applikation

Die Applikation wurde mit PHP geschrieben. Dabei wurde das PHP5 Framework „Symfony2“ eingesetzt. Wir hatten noch nie mit diesem Framework gearbeitet, wollten uns aber schon von Anfang an damit auseinandersetzen. Dieses Projekt hat uns dabei optimal geholfen, auch wenn wir dafür extrem viel Zeit aufwenden mussten um uns in alle Technologien einzulesen.

5.5.1. Technologien

Symfony2 ist so aufgebaut, dass es nicht als ein fix fertiges Framework dient. Es bedient sich ebanfalls an diversen anderen Frameworks und Tools. Man kann sich sein Framework somit auf die gewünschten Bedürfnisse zusammenstöppeln.

<https://symfony.com/>

Nebst Symfony2 wurde mit **Doctrine2**, einem ORM gearbeitet. Dies führte auch zu der Datenbankarchitektur im Kapitel 5.4. Grundsätzlich wurden die Entitäten und deren Assoziationen in einer PHP Klasse mit entsprechenden Annotations definiert.

<http://www.doctrine-project.org/>

Eine weitere nennenswerte Komponente ist die Template Engine **Twig**. Twig erlaubt es den PHP Code komplett von der View (also dem HTML) zu trennen. Zudem bietet es eine Reihe an Funktionalitäten und logischen Konstrukten die die Ausgaben vereinfachen.

<http://twig.sensiolabs.org/>

Einige weitere Komponenten welche ich nennen möchte sind speziell für Symfony2 konzipiert. Symfony2 setzt viel Wert auf DependencyInjection. Es wird also möglichst versucht das System zu entkoppeln und einzelne Komponenten zu verwenden, diese werden Bundles genannt. Im Internet gibt es eine Reihe solche OpenSource Bundles für Symfony2.

Besonders betonen möchten wir hier den „**FOSUserBundle**“, welcher uns eine komplette grundlegende Userverwaltung brachte, auf welcher wir wunderbar aufbauen konnten.

<https://github.com/FriendsOfSymfony/FOSUserBundle>

Der „**JMSSerializerBundle**“ half uns dabei, das importierte XML in Objekte zu mappen.

<https://github.com/schmittjoh/JMSSerializerBundle>

Es wurden noch einige weitere Bundles benutzt. Weitere Informationen dazu findet man im Composer File der Software.

5.5.2. Continuous Integration

Auch bei diesem Thema fehlte mir zuvor jegliche praktische Erfahrung. Wir hatten im SAuD Kurs (Software, Architektur und Design) zwar kurz einige Prinzipien durchgenommen, jedoch noch nie ein Projekt damit umgesetzt. Wir wollten unbedingt mal einen Continuous Integration Workflow aufsetzen.

Unsere Projekte hosten wir eigentlich immer auf **Github**. Nicht nur weil wir Fan von OpenSource sind, sondern auch um ein persönliches Portfolio an Projekten online zu haben. Da geniale an OpenSource Projekten welche auf Github sind ist, dass es diverse Tools zur kostenlosen Verwendung gibt.

<https://github.com/Cogax/SysVentryServer>

So sind wir auch auf **Travis-CI** gestossen, welches einen kostenlosen Test-Build-Server für Github Projekte bietet. In einer kleinen Datei ([.travis.yml](#)) werden die Builds definiert. Wir haben das Projekt immer auf 3 PHP Versionen getestet, 5.3, 5.4 und 5.5.
<https://travis-ci.org/Cogax/SysVentryServer>

Symfony2 untertützt (natürlich) auch Unit tests. In PHP heisst die meistgenutzte Library dazu **PHPUnit**. Wir haben die wichtigsten Teile der Software mit Tests abgesichert, für TDD (noch eine neue Technologie) hat es leider in diesem Projekt nicht mehr gereicht.
<https://phpunit.de/>

Um die Software schlussendlich sauber auf den Produktiven Server zu deployen haben wir **Jenkins** verwendet. Wir haben einen Pipeline an Jobs angelegt, welche bei einer entsprechenden Github Aktion ausgeführt werden und die gesamte Software von A-Z sauber aufsetzt.

<https://jenkins-ci.org/>

5.5.2.1. Workflow

- Auf Github wurden 2 Branches erstellt, *master* und *prod*. Wir haben jeweils mit dem *master* Branch lokal gearbeitet und die Änderungen auf Github gepusht.
- Travis hat dies bemerkt und nach jedem push in den master Branch von uns, hat er automatisch den master branch gepullt, 3 Umgebungen aufgesetzt (PHP 5.3, 5.4 und 5.5) und darauf die Tests ausgeführt.
- Falls die Test fehlschlugen bekamen wir eine E-Mail
- Im Falle erfolgreicher Tests, wurde unser [deploy.sh](#) Skript ausgeführt, welches den aktuellen Stand in den *prod* Branch gepackt hat und auf Github gepusht hat.
- Jenkins hat dann bemerkt, dass sich der *prod* Branch geändert hat und hat die Pipeline an Jobs ausgeführt um die Software auf den Produktiven Server zu deployen.

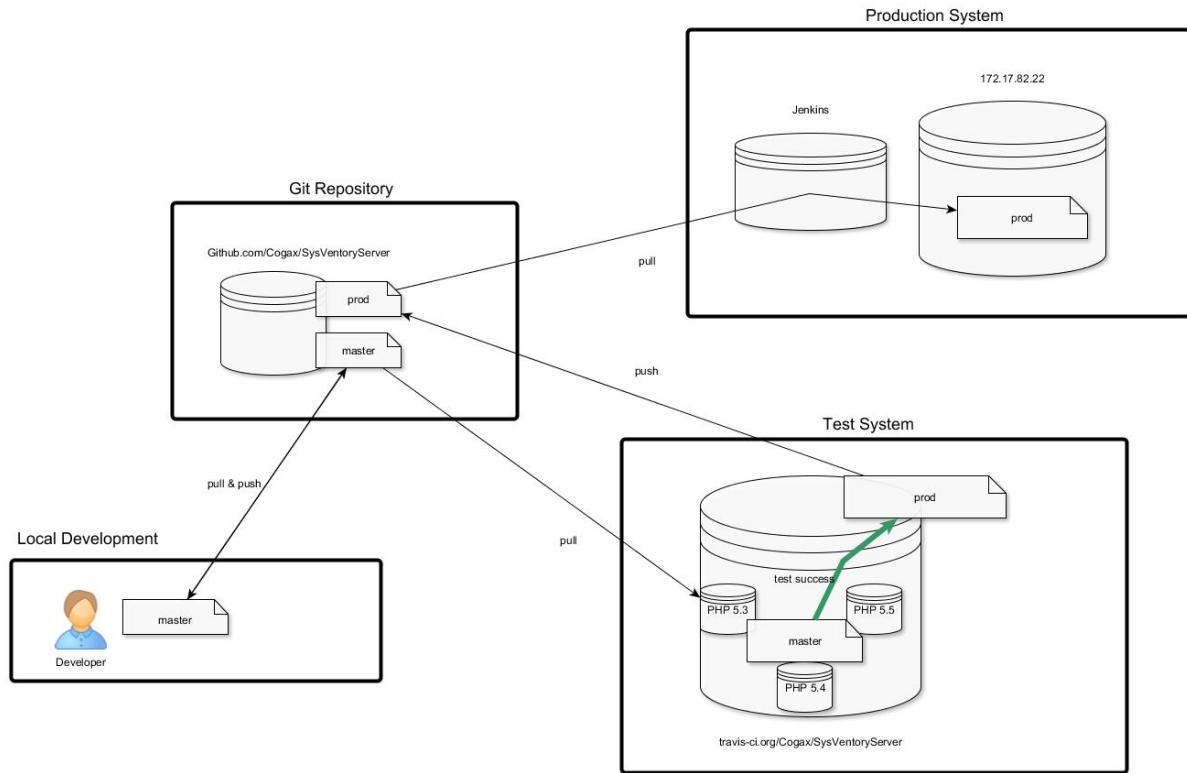


Abbildung 9 - Workflow

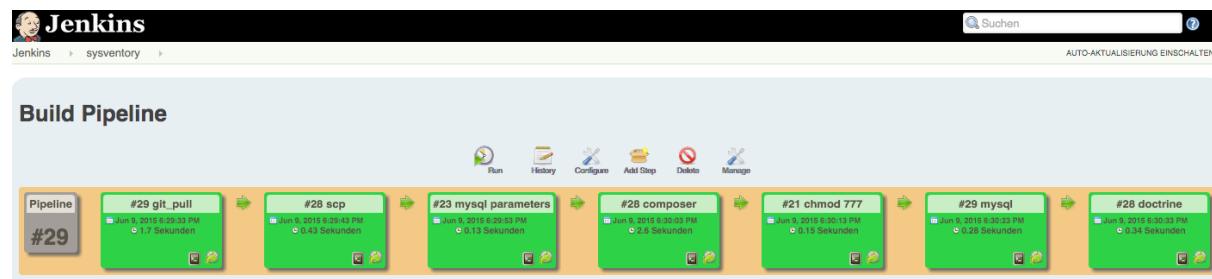


Abbildung 10 – Build Pipeline Jenkins

5.5.3. Komponenten

Die Software an sich besteht nicht nur aus dem Web-GUI, sondern auch aus der Schnittstelle um den XML Request vom Collector entgegenzunehmen.

5.5.3.1. Web-GUI

Die Daten werden bei Import in die Datenbank geschrieben und anschliessen in der Web GUI angezeigt. Im Frontend wurde mit den Frameworks **Bootstrap** und **jQuery** gearbeitet.

<http://getbootstrap.com/> <https://jquery.com/>

5.5.3.2. Schnittstelle

Die Schnittstelle wurde eigenhändig Programmiert, basierend auf dem http-Protokoll. Um immer richtige Daten zu empfangen wurde ein [XML Schema](#) erstellt, welches die ankommenden Daten auf ihre Korrektheit prüft.

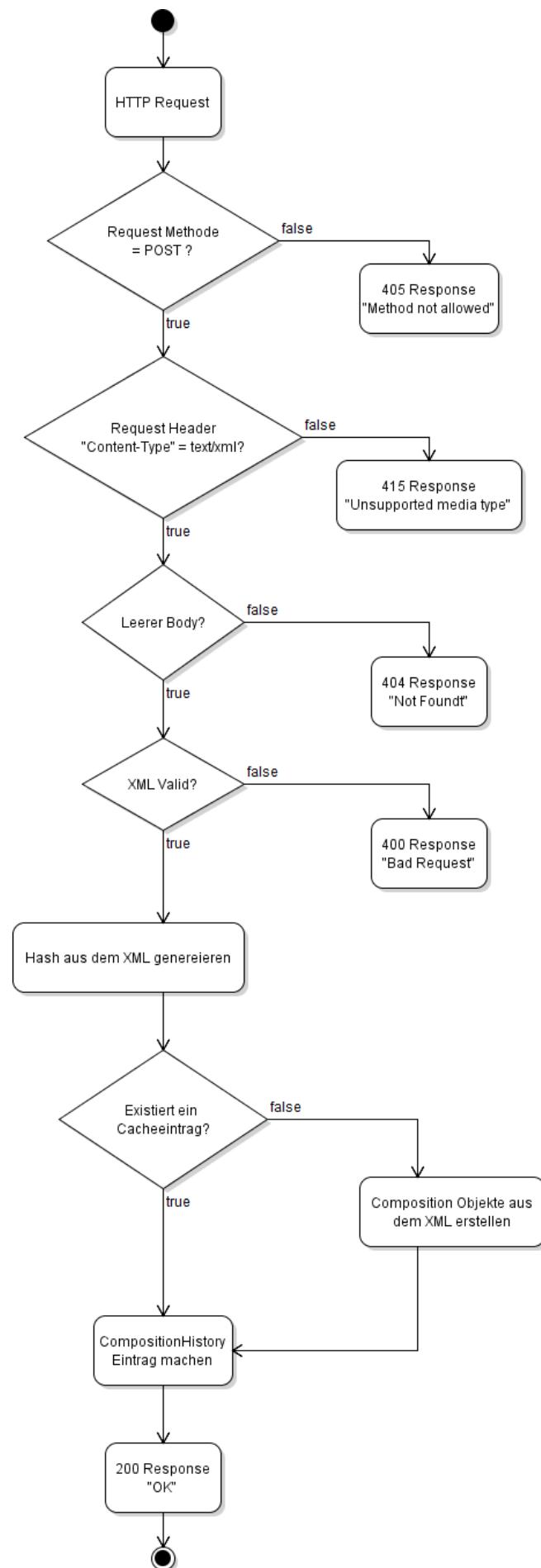


Abbildung 11 – HTTP- Request

5.5.4. Funktionen

5.5.4.1. Login

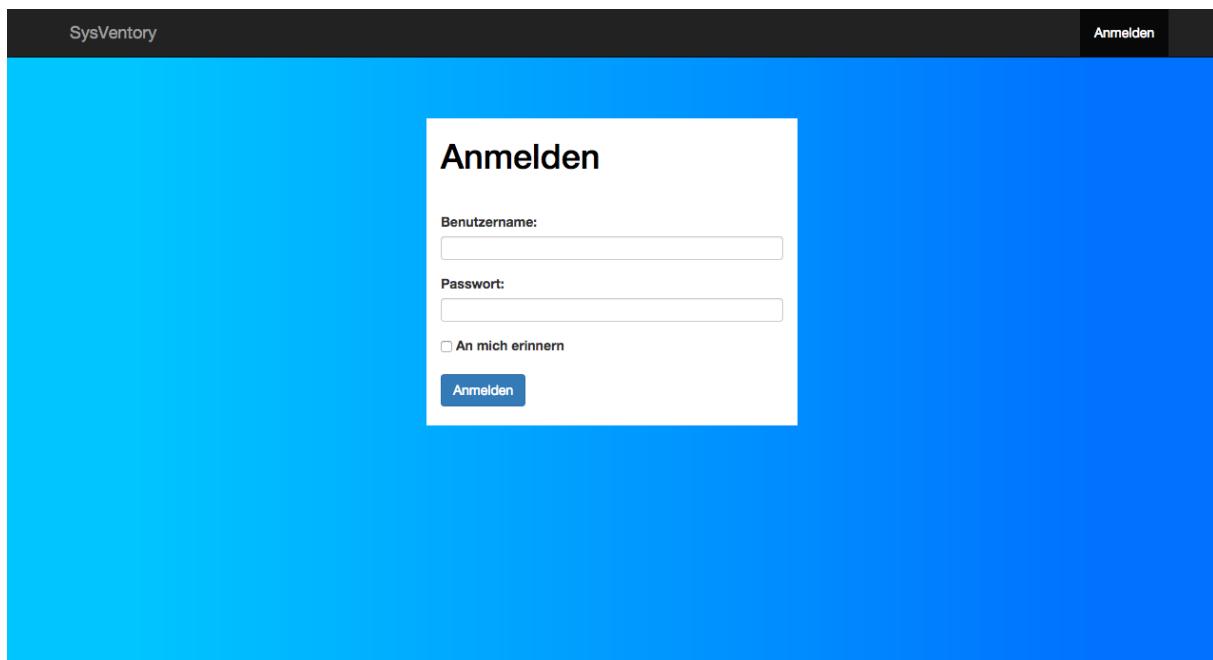


Abbildung 12 – Ansicht Login

Geschützt wird die gesamte Software von einem Usersystem, welches auf verschiedenen Rollen basiert. Es wurden zwei Benutzer mit je einer anderen Rolle erstellt. Es sind die Benutzer *inventor* und *visitor*.

5.5.4.2. Startseite

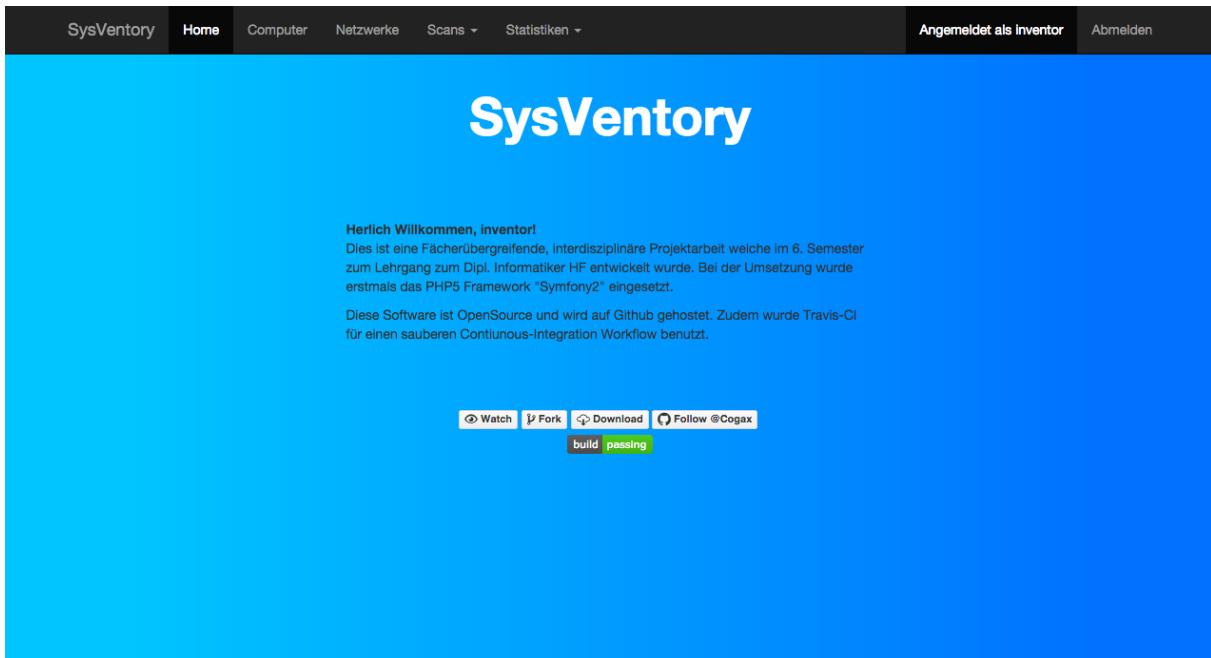


Abbildung 13 – Ansicht Welcom

Nach dem Login erhält man Zugang zur Startseite. Je nach Rolle werden die verfügbaren Menüpunkte angezeigt oder ausgeblendet. Zudem sind die Seiten noch einzeln abgesichert. Oben rechts sieht man mit welchem User man eingeloggt ist und kann sich wieder Abmelden.

5.5.4.3. Computer Übersicht

Alle Computer					
Uuid	Computername	IP	CPU	Betriebssystem	letzter Scan
79B60242-1F48-F7E8-8169-FB15DFB0B212	CLIENT01-01	172.17.82.101	Intel Xeon E312xx (Sandy Bridge)	Microsoft Windows 7 Professional (64-Bit)	15.06.2015 21:00:23
858170E4-16A8-41C4-AA1A-358DBB5C50CE	CLIENT01-02	172.17.82.102	Intel Xeon E312xx (Sandy Bridge)	Microsoft Windows 7 Ultimate (64-Bit)	15.06.2015 21:00:24

Abbildung 14 – Ansicht Alle Computer

Klickt man auf „Computer“ sieht man alle Computer die einmal bei einem Scan entdeckt worden sind. Als eindeutige Identifizierung wird hier die Uuid verwendet. Diese Liste ist durchsuchbar, sortierbar und wird bei einer grösseren Menge auf mehrere Seiten verteilt um dem Benutzer eine angenehme Übersicht zu bieten.

5.5.4.4. Computer

The screenshot shows the SysInventory software interface for a specific machine. At the top, there's a navigation bar with tabs for 'SysInventory', 'Home', 'Computer' (which is selected), 'Netzwerke', 'Scans', and 'Statistiken'. On the right side of the header are 'Angemeldet als Inventor' and 'Abmelden' buttons.

The main content area has a large blue header with the machine's Uuid: 'Machine #79B60242-1F48-F7E9-8169-FB15DFB0B212'. Below it, there's a large number '3' indicating three versions found, followed by the text 'Versionen gefunden'.

A table titled 'Versionen' lists three entries:

ID	Datum	Alt	Neu
1	15.06.2015 21:00:23	<input type="radio"/>	<input checked="" type="radio"/>
4	15.06.2015 20:23:09	<input type="radio"/>	<input type="radio"/>
3	15.06.2015 20:22:32	<input type="radio"/>	<input type="radio"/>

At the bottom of this section is a 'Vergleichen' (Compare) button.

Below this, there are several sections with tables:

- Computer:** Shows basic hardware information like Uuid, Hersteller (Bochs), Model (Bochs), Speicher (2147069952), Computer Name (CLIENT01-01), and Seriennummer.
- Betriebssystem:** Shows the operating system as Microsoft Windows 7 Professional, Version 6.1.7601, Service Pack 1, and Architektur (64-Bit).
- Netzwerkarten:** Shows network card details: IP (v4) 172.17.82.101, Subnet 255.255.255.0, IP (v4) Default Gateway 172.17.82.1, IP (v6) 2a02:16a0:ff02:82:44d0:53b2:4010:b3dc, IP (v6) Subnet 64, IP (v6) Default Gateway fe80::5054:ff:fe6d:bd37, DHCP Nein, MAC 52:54:00:9C:30:73, and Beschreibung (Intel(R) PRO/1000 MT-Netzwerkverbindung).
- CPU's:** Shows a single CPU entry: Intel Xeon E312xx (Sandy Bridge), Familie Intel64 Family 6 Model 42 Stepping 1, Clock 3300, and Cores 1.
- Drucker:** Shows two printer entries: Microsoft XPS Document Writer (Treiber Microsoft XPS Document Writer, Version 7.0.790) and Fax (Treiber Microsoft Shared Fax Driver).
- Software:** Shows software installations: Java 7 Update 79 (64-bit) (Version 7.0.790), Microsoft .NET Framework 4 Extended (Version 4.0.30319), and Microsoft .NET Framework 4 Client Profile (Version 4.0.30319).

At the bottom left is a 'Zurück zur Übersicht' (Back to Overview) button.

Abbildung 15 – Ansicht Machine

Klickt man auf die Uuid eines Computers, so kommt man zur Detailansicht dieses Computers. Ganz oben sieht man nochmals die Uuid, gefolgt von der Versionierung und den einzelnen Daten. Bei den Versionen hat man die Möglichkeit durch einen klick auf das Datum die Daten zu diesem Zeitpunkt zu sehen. Mit „Alt“ und „Neu“ kann man zwei Versionen markieren, welche man direkt miteinander vergleichen möchte.

5.5.4.5. Versionen vergleichen

Kompositionen vergleichen: #4 / #1

Computer					
Uuid	Hersteller	Model	Speicher	Computer Name	Seriennummer
79B60242-1F48-F7E9-8169-FB15DFB0B212	Bochs	Bochs	2147069952	CLIENT01-01	

Betriebssystem				
Name	Version	Service Pack	Architektur	
Microsoft Windows 7 Professional	6.1.7601	1	64-Bit	

Netzwerkarten							
IP (v4)	IP (v4) Subnet	IP (v4) Default Gateway	IP (v6)	IP (v6) Subnet	IP (v6) Default Gateway	DHCP	MAC
172.17.82.101	255.255.255.0	172.17.82.1	2a02:16a0:ff02:82:44d0:53b2:4010:b3dc	64	fe80::5054:ff:fe6d:bd37	Nein	52:54:00:9C:30:73
Intel(R) PRO/1000 MT-Netzwerkverbindung							

CPUs				
Name	Familie	Clock	Cores	
Intel Xeon E312xx (Sandy Bridge)	Intel64 Family 6 Model 42 Stepping 1	3400	1	
Intel Xeon E312xx (Sandy Bridge)	Intel64 Family 6 Model 42 Stepping 1	3300	1	

Drucker		
Name	Treiber	Version
Microsoft XPS Document Writer	Microsoft XPS Document Writer	
Fax	Microsoft Shared Fax Driver	

Software	
Name	Version
Microsoft .NET Framework 4 Extended	4.0.30319
Microsoft .NET Framework 4 Client Profile	4.0.30319
Java 7 Update 79 (64-bit)	7.0.790

Abbildung 16 – Ansicht Kompositionen vergleichen

Wählt man zwei Versionen zum vergleichen so werden diese in einer Art „Diff“ Darstellung angezeigt. Eine **rote Zeile** bedeutet, dass diese in der alten Version, aber nicht mehr in der neuen Version vorhanden ist. Eine **grüne Zeile** bedeutet, dass diese in der neuern Version hinzugefügt wurde.

5.5.4.6. Netzwerke

The screenshot shows the SysInventory interface with the 'Netzwerke' tab selected. The main title is 'Gespeicherte Netzwerke'. A table lists one network entry:

ID	Name	Range	Actions
1	maintenance01-01	172.17.82.0/24	Bearbeiten Scanen

A blue button at the bottom left says 'Neues Netzwerk erfassen'.

Abbildung 17 – Ansicht Gespeicherte Netzwerke

Wenn man auf Netzwerke klickt sieht man die gespeicherten Netzwerke. Diese Netzwerke können bei einem Scan zum speichern markiert werden. Diese gespeicherten Netzwerke werden einmal täglich automatisch gescannt. Zugriff auf die Aktionen *Bearbeiten* und *Scanen* hat nur die Inventor-Rolle.

5.5.4.7. Scan erfassen

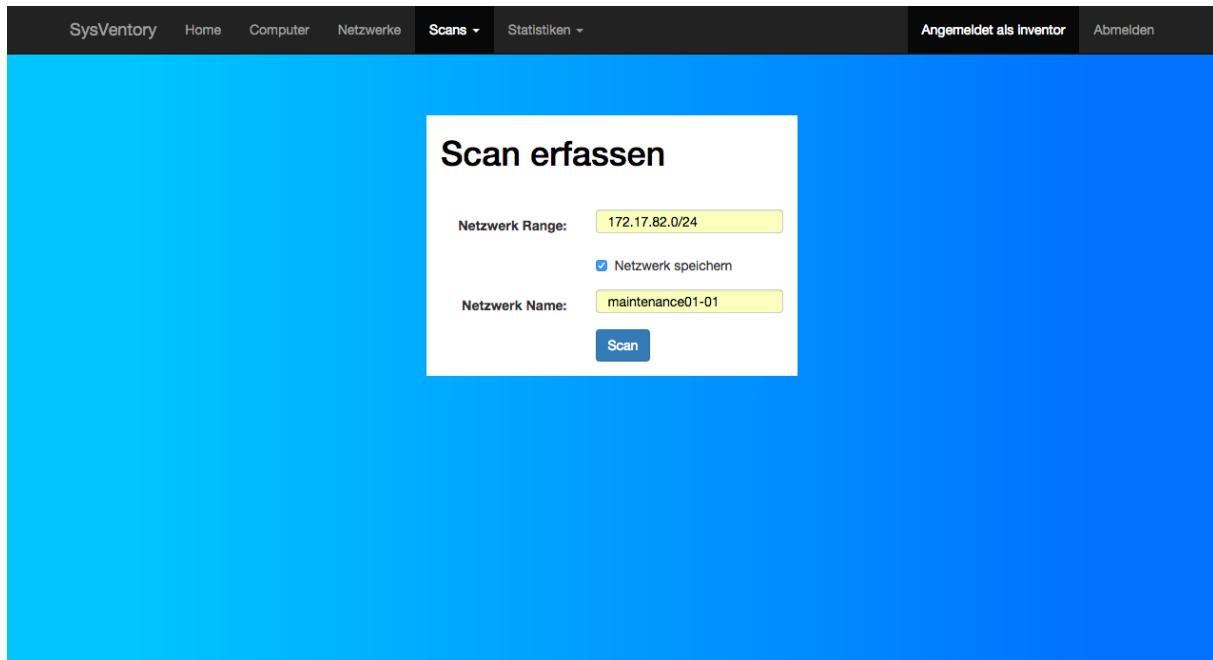


Abbildung 18 – Ansicht Scan erfassen

Einen neuen Scan ist leicht zu erfassen. Erst wird der Netzwerk-Range angegeben. Dieser kann in folgenden Formaten erfasst werden:

- 172.17.82.101 (IP)
- 172.17.82.101-172.17.82.105 (Range)
- 172.17.82.0/24 (IP & Subnetz)

Zudem kann die Option „Netzwerk speichern“ gewählt werden, worauf das Eingabefeld für den Netzwerk Name erscheint. Damit wird dieses Netzwerk in die Netzwerkliste hinzugefügt und wird automatisch gescannt.

5.5.4.8. Scan History

The screenshot shows a table titled "Scan History" with three rows of data. The columns are "Id", "Time", and "Netrange".

Id	Time	Netrange
3	15.06.2015 21:00:04	172.17.82.0/24
2	15.06.2015 20:00:04	172.17.82.0/24
1	15.06.2015 19:49:02	172.17.82.0/24

Abbildung 19 – Ansicht Scan History

Die Scan-History listet alle durchgeführten Scans auf.

5.5.4.9. Statistikmodul

The screenshot shows a pie chart titled "Software" with various segments. The legend lists the software names and their percentages:

Software	Percentage
Java 7 Update 79 (64-bit) (7.0.790)	16.7%
Microsoft .NET Framework 4 Extens... (4.0.30319.42000)	16.7%
Microsoft .NET Framework 4.5 (4.5.50727.1)	8.3%
iTunes (12.1.2.27)	8.3%
Bonjour (3.0.0.10)	8.3%
Apple Software Updater (1.0.1)	8.3%
Apple Mobile Device Support (1.0.1)	8.3%
Google Update (2.2.10.100)	8.3%
Apple Application Support (1.0.1)	8.3%
Apple Application Support (1.0.1)	8.3%

Abbildung 20 – Ansicht Eingesetzte Software

Das Statistikmodul umfasst viele Statistiken in welchen etliche Daten grafisch ausgewertet werden können. Eine davon ist zum Beispiel die Statistik über die eingesetzte Software.

6. Fazit

Das Projekt konnte erfolgreich realisiert werden. Dies ist auf ein funktionierendes und gut organisiertes Projektteam zurück zu führen. Die einzelnen Komponenten im Team bearbeiteten die Ihnen zugeteilten Aufgaben stets fristgerecht und in einer guten Qualität.

Aus den Problemen welche sich im Verlauf des Projekts ergaben, konnten wertvolle Erkenntnisse entnommen welche wir später ins Projekt einfließen lassen konnten.

Wir würden das Projekt wieder ähnlich angehen. Das lange Kick – Off Meeting war ein gutes Instrument um das genaue Vorgehen zu diskutieren. Es war ein wichtiger Faktor um nicht in einer Sackgasse zu landen.

Die Zeitplanung konnte vom ganzen Team eingehalten werden, sodass es nie zu kritischen Punkten kam.

Auch wurden die Stärken der einzelnen Teammitglieder optimal eingesetzt.

An dieser Stelle möchten wir uns nochmals herzlichst bei der Firma Automatic Server bedanken, welche uns während der Zeit des Projekts eine sehr performante und zuverlässige Umgebung zur Verfügung stellte.

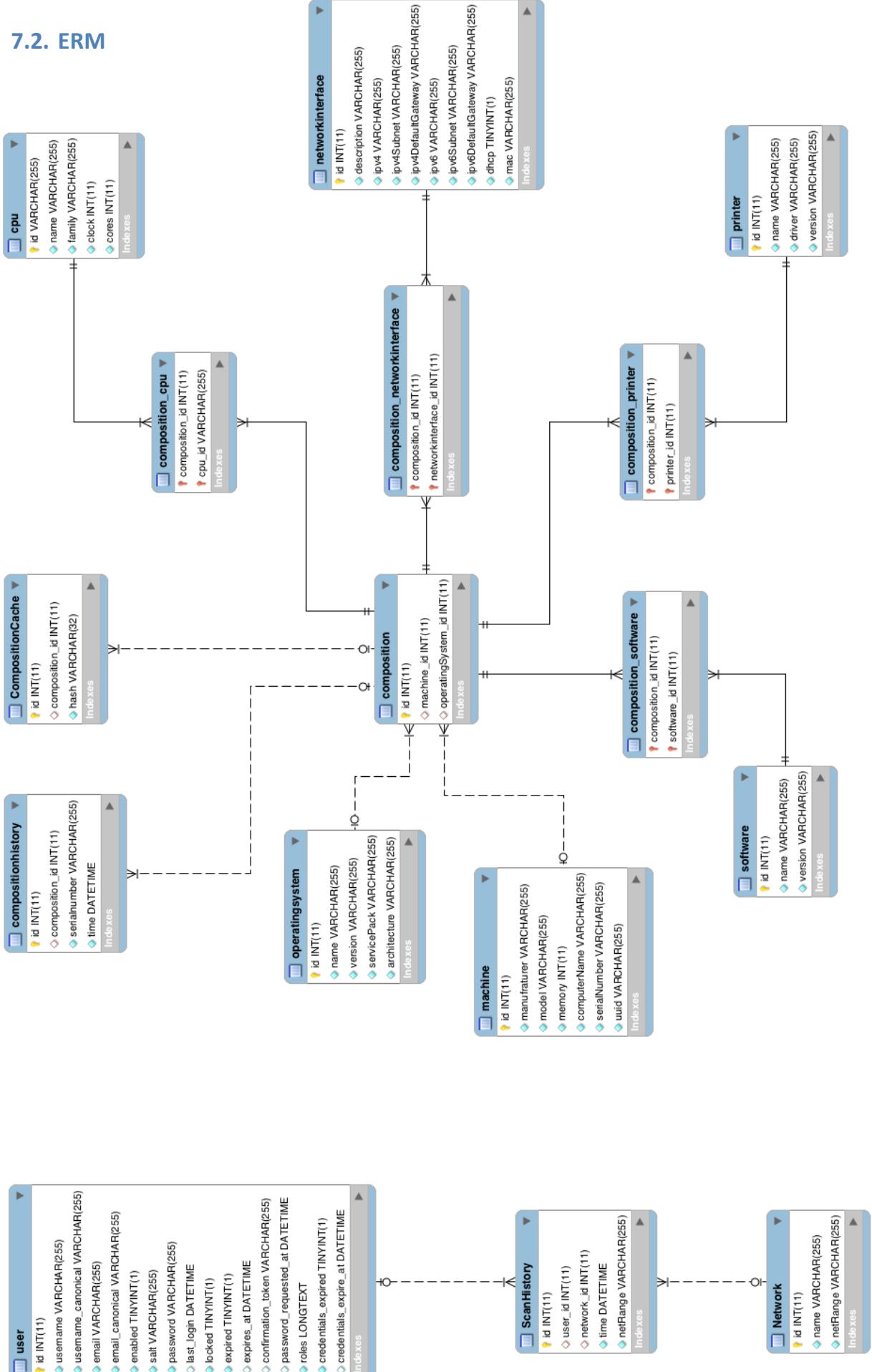
7. Anhang

7.1. Projekttagebuch

Wann	Wer	Zeitaufwand in h	Durchgeführte Tätigkeit
14.04.15	Ganzes Team	15	Kick Off, Vorgehens Besprechung
15.04.15	Lukas Stadelmann	0.5	Erstellen des Projekttagebuchs
15.04.15	Nico Florin	2	Einarbeit techn. Informationsbeschaffung
16.04.15	Nico Florin	2	Start Powershell Script für Informationsbeschaffung
17.04.15	Andreas Gyr	2	Einarbeiten Datenbank Möglichkeiten
18.04.15	Daniel Ammann	3	DB
18.04.15	Benjamin Rechsteiner	3	Infrastruktur
20.04.15	Benjamin Rechsteiner	4	Infrastruktur
21.04.15	Benjamin Rechsteiner	2	Infrastruktur
23.04.15	Andreas Gyr	2	DB
23.04.15	Daniel Ammann	1.5	DB
18.05.15	Lukas Stadelmann	2	AD einrichten
19.04.15	Nico Florin	2	Powershell
19.04.15	Lukas Stadelmann	3	Arbeitspakete fertig definieren
22.04.15	Benjamin Rechsteiner	4	Grafiken zur Infrastruktur
24.04.15	Lukas Stadelmann	2	Vorbereitung Doku
28.04.15	Nico Florin	3	Powershell
28.04.15	Ganzes Team	15	Weiteres Vorgehen besprechen/ MS PII
30.04.15	Andreas Gyr	2	DB
30.04.15	Daniel Ammann	2	DB
30.04.15	Lukas Stadelmann	2	DB
02.05.15	Nico Florin	1	Powershell
04.05.15	Ganzes Team	15	MS vorbereitung
05.05.15	Ganzes Team	17.5	Weiteres Vorgehen besprechen/ MS PDB
07.05.15	Benjamin Rechsteiner	2.75	Jenkins
07.05.15	Andreas Gyr	2.75	Jenkins
07.05.15	Lukas Stadelmann	2	Cronjob
12.05.15	Ganzes Team	15	Weiterführen der begonnen Arbeiten
18.05.15	Ganzes Team	17.5	Weiteres Vorgehen besprechen/ MS PIZ
21.05.15	Nico Florin	2	Powershell
21.05.15	Daniel Ammann	2	Powershell
29.05.15	Nico Florin	4	Powershell (HTTP Request)
02.06.15	Ganzes Team	10	Besprechung weiteres Vorgehen
02.06.15	Nico Florin	2	Dokumentation Scripts
02.06.15	Lukas Stadelmann	2	Doku
02.06.15	Daniel Ammann	2	Doku
02.06.15	Benjamin Rechsteiner	3	Schnittstellen
02.06.15	Andreas Gyr	3	Schnittstellen
06.06.15	Andreas Gyr	4	GUI/Auswertung
08.06.15	Lukas Stadelmann	2	Testen des Systems
09.06.15	Nico Florin	1	Anpassung Powershell
09.06.15	Benjamin Rechsteiner	2	Fehler Behebungen
11.06.15	Andreas Gyr	5	GUI/Auswertung
13.06.15	Ganzes Team	10	Testen der Funktionalitäten
14.06.15	Lukas Stadelmann	5	Doku

14.06.15	Andreas Gyr		GUI/Auswertung fertigstellen
14.06.15	Daniel Ammann	4.5	Doku/Präsentation
15.06.15	Daniel Ammann	3.5	Doku/Präsentation
15.06.15	Lukas Stadelmann	6	Doku/Präsentation
15.06.15	Nico Florin	1	Präsentation Vorbereitung
15.06.15	Benjamin Rechsteiner	1	Präsentation Vorbereitung
15.06.15	Andreas Gyr	2	Präsentation Vorbereitung
16.06.15	Ganzes Team	1	Abgabe und Präsentation
	TOTAL	226.5	

7.2. ERM



7.3. Zeitlicher Projektplan

2015												
Januar						April						
S	M	D	M	D	F	S	S	M	D	M	F	S
4	5	6	7	8	9	10	5	6	7	8	9	10
11	12	13	14	15	16	17	12	13	14	15	16	17
18	19	20	21	22	23	24	19	20	21	22	23	24
25	26	27	28	29	30	31	26	27	28	29	30	31
Februar						Mai						
S	M	D	M	D	F	S	S	M	D	M	F	S
1	2	3	4	5	6	7	3	4	5	6	7	8
8	9	10	11	12	13	14	10	11	12	13	14	15
15	16	17	18	19	20	21	17	18	19	20	21	22
22	23	24	25	26	27	28	24	25	26	27	28	29
März						Juni						
S	M	D	M	D	F	S	S	M	D	M	F	S
1	2	3	4	5	6	7	1	2	3	4	5	6
8	9	10	11	12	13	14	7	8	9	10	11	12
15	16	17	18	19	20	21	14	15	16	17	18	19
22	23	24	25	26	27	28	28	29	30	31	30	31
Juli						August						
S	M	D	M	D	F	S	S	M	D	M	F	S
5	6	7	8	9	10	11	5	6	7	8	9	10
12	13	14	15	16	17	18	12	13	14	15	16	17
19	20	21	22	23	24	25	19	20	21	22	23	24
26	27	28	29	30	31	1	26	27	28	29	30	31
Oktober						November						
S	M	D	M	D	F	S	S	M	D	M	F	S
4	5	6	7	8	9	10	4	5	6	7	8	9
11	12	13	14	15	16	17	11	12	13	14	15	16
18	19	20	21	22	23	24	18	19	20	21	22	23
25	26	27	28	29	30	31	25	26	27	28	29	30
September						Dezember						
S	M	D	M	D	F	S	S	M	D	M	F	S
1	2	3	4	5	6	7	1	2	3	4	5	6
8	9	10	11	12	13	14	8	9	10	11	12	13
15	16	17	18	19	20	21	15	16	17	18	19	20
22	23	24	25	26	27	28	22	23	24	25	26	27
29	30	31					29	30				

Planung
Realisierung
Test- Phase
Schlussarbeiten Dokumentierung