

Einbinden externer Systemmanagement-Programme in die SAP-CCMS-Monitoring-Architektur





Inhaltsverzeichnis

1EINLEITUNG	3
2AUFGABE EXTERNER SCHNITTSTELLEN	4
3EINFÜHRUNG IN DIE MONITORING-ARCHITEKTUR	5
4KONZEPT DER EXTERNEN SCHNITTSTELLEN	7
5XM – SCHNITTSTELLEN (EXTERNAL SAP MANAGEMENT)	9
6SAP MONITORING SHARED LIBRARY	11
7NAGIOS	23
8GLOSSAR	28

1 Einleitung

1.1 Motivation

Die Verwaltung verteilter Client-Server-Systeme stellt hohe Anforderungen an Systemadministratoren. Bei verteilten Systemen, die über die Mindestkonfiguration hinausgehen, sind Werkzeuge zur Unterstützung von Systemadministratoren bei dieser Aufgabe unentbehrlich.

Das Computing Center Management System (CCMS) bietet eigene Systemverwaltungswerkzeuge, die eine integrierte Überwachung und Verwaltung mehrerer SAP-Systeme und ihrer gesamten Infrastrukturen einschließlich SAP-Geschäftskomponenten im SAP Business Framework ermöglichen.

Oftmals existieren schon firmenweite Systemmanagement-Infrastrukturen. In diesen Fällen ist eine Integration der CCMS Monitoring Infrastruktur in die firmenweiten Infrastrukturen sinnvoller, als mehre Systemmanagement-Anwendungen parallel zu betreiben.

Dem Anwender wird dadurch eine homogene Infrastruktur zur Anzeige der nötigen Informationen geboten.

2 Aufgabe externer Schnittstellen

Externe Schnittstellen sollen es ermöglichen, SAP-Installationen durch Programme anderer Hersteller zu überwachen und zu administrieren.

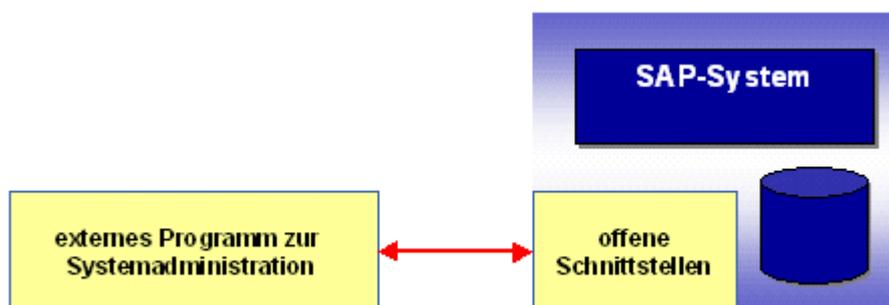


Abbildung 1: Externe Schnittstellen

Über externe Schnittstellen können SAP-Systeme sowohl in lokale Administrationsprogramme als auch in firmenweite Systemmanagement-Infrastrukturen integriert werden.

Das Ziel der Integration ist es, dem Anwender eine homogene Infrastruktur zur Anzeige der nötigen Informationen zu ermöglichen. Das Ziel der Schnittstellen ist es, den Informationsfluss zwischen SAP-

Systemen und externen Programmen zu erleichtern. So können externe Programme zusätzliche Flexibilität bei der Ergänzung grundlegender CCMS-Funktionen bieten.

3 Einführung in die Monitoring-Architektur

Es gibt heute kaum eine Organisation, Firma, Behörde oder Einrichtung mehr, die ihre Aufgaben ohne den massiven Einsatz von IT-Systemen bewältigen könnte. Sind diese Systeme oder einzelne Komponenten nicht verfügbar, kann dies erhebliche Folgen haben, so dass ein permanentes Management dieser Systeme unumgänglich ist. Fehlersituationen müssen frühzeitig erkannt werden; das Zusammenspiel der unzähligen Teilkomponenten muss organisiert und überwacht werden. Leistungsentpässe müssen analysiert und behoben werden.

Diese Aufgaben werden mit dem steigenden Grad der Verteilung solcher Systeme immer komplexer und sind nur noch mit leistungsfähigen Managementwerkzeugen zu bewältigen.

Für die Überwachung und Verwaltung mehrerer SAP-Systeme mitsamt ihrer Infrastrukturen bietet das Computing Center Management System (CCMS) eigene Systemverwaltungswerkzeuge. Somit bietet die Monitoring-Architektur des Computing Center Management System Systemadministratoren ein flexibles Werkzeug für die Überwachung und Verwaltung komplexer IT-Landschaften von SAP- und Nicht-SAP-Komponenten.

Die Monitoring-Architektur sammelt Informationen aus der gesamten SAP-Umgebung: dem SAP-System, der Datenbank, den Servern, auf denen das SAP-System läuft, und dem Netzwerk. Anhand des eingehenden Datenstroms präsentiert die Monitoring-Architektur einen klaren Überblick über den Zustand der SAP-Systeme und ihrer Umgebung. Systemadministratoren sehen mit einem Blick, ob ihre Systeme ordnungsgemäß laufen. Probleme hebt die Monitoring-Architektur durch Alerts hervor, und liefert die für deren Lösung erforderlichen Methoden und Daten.

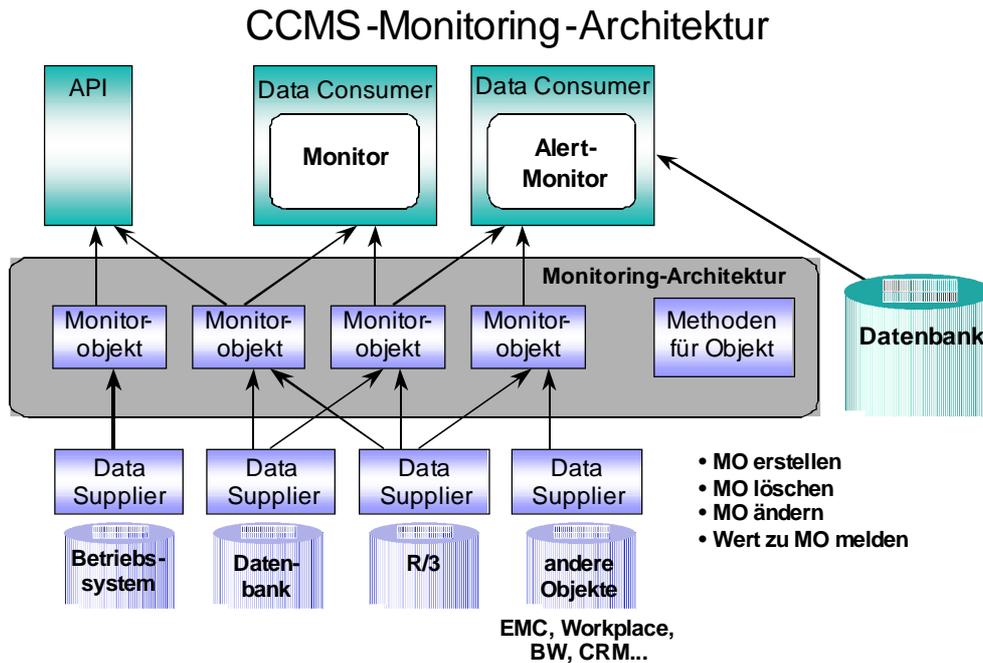


Abbildung 2: Schematischer Aufbau der Monitoring-Architektur

Die CCMS-Monitoring-Architektur ist kein monolithisches Überwachungs- und Verwaltungsprogramm. Vielmehr bietet sie einen flexiblen Rahmen, dem problemlos weitreichende Überwachungs- und Verwaltungsfunktionen hinzugefügt werden können. Die Bestandteile der Monitoring-Architektur arbeiten weitgehend unabhängig voneinander und können vor allem auch unabhängig voneinander weiterentwickelt und angepasst werden:

- **Data Supplier**

Ein Datasupplier ist ein Programm, das der Monitoring-Architektur Daten zu überwachten Objekten liefert. Die Monitoring-Architektur bietet eine Infrastruktur zum Erfassen und Verwalten von Systeminformationen an. Die Monitoring-Architektur vergleicht die Messwerte der Data Suppliers ständig mit Schwellenwerten und löst beim Über- oder Unterschreiten einen Alert aus.

- **Data Consumer**

Ein Data Consumer ist ein Programm, das Daten aus der Monitoring-Architektur liest; es zeigt die von den Data Suppliers an die Monitoring-Architektur übergebenen Informationen an.

- **Monitorobjekte und -attribute**

Ein Monitorobjekt repräsentiert eine zu überwachende Komponente in der IT-Umgebung, z. B. die CPU-Auslastung eines Servers, das Dialogsystem oder die Hintergrundverarbeitung. Monitorattribute sind Werte, Status oder Texte, die zu diesem Objekt gemeldet werden, etwa die CPU-Auslastung oder die Antwortzeit im Dialogsystem.

4 Konzept der externen Schnittstellen

4.1 BAPI – Grundlagen

BAPIs (Business Application Programming Interface) sind SAP-Standard-Schnittstellen. Sie spielen eine wesentliche Rolle bei der technischen Integration und dem betriebswirtschaftlichen Datenaustausch zwischen SAP-Komponenten untereinander und zwischen SAP- und Nicht-SAP-Komponenten.

BAPIs ermöglichen eine Integration auf betriebswirtschaftlicher und nicht auf technischer Ebene. Somit wird eine größere Stabilität der Kopplung und eine Unabhängigkeit von der eingesetzten Kommunikationstechnologie gewährleistet.

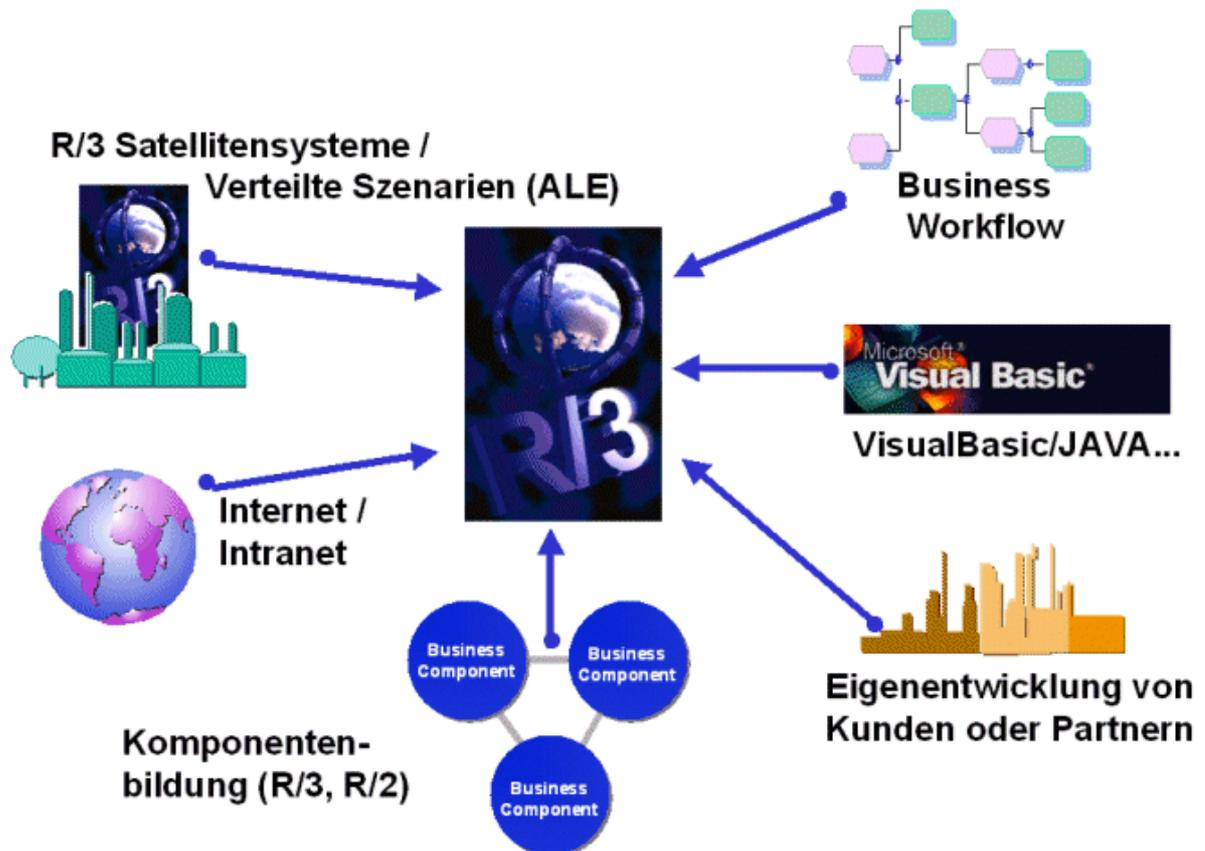


Abbildung 3: Anbindung verschiedener Anwendungen an das SAP-System durch BAPIs

Abbildung 3 zeigt, wie BAPIs als Schnittstellen die Anbindung verschiedenster Anwendungen an das SAP-System ermöglichen:

- Implementierung von verteilten R/3-Szenarien
- Anbindung des R/3-Systems an das Internet
- Anbindung von neuen SAP-Komponenten, von Fremdsoftware und Legacy-Systemen
- Realisierung von Workflow-Anwendungen, die über Systemgrenzen hinweg kommunizieren
- Verwendung von PC-Programmen als "Frontend" für das R/3-System. Diese können z.B. mit Visual Basic oder Java entwickelt werden.
- Eigenentwicklungen von Kunden und Partnern

Objektorientierte Technologien haben sich immer mehr zum Standard bei der Kommunikation zwischen unterschiedlichen Softwareprodukten etabliert. Dazu wurden bei SAP Business-Objekte eingeführt, die die Daten und Prozesse nach betriebswirtschaftlichen Kriterien strukturieren. Mit Hilfe der Business-Objekte wird folglich das SAP-System in kleinere, disjunkte Einheiten zerlegt, wodurch die Strukturierung verbessert und die Komplexität reduziert wird.

Die BAPIs werden als Methoden an den Business-Objekten definiert. Mit Hilfe dieser objektorientierten Schnittstellen können andere Komponenten somit direkt auf die Anwendungsschicht eines SAP-Systems zugreifen, ohne Implementierungsdetails kennen zu müssen.

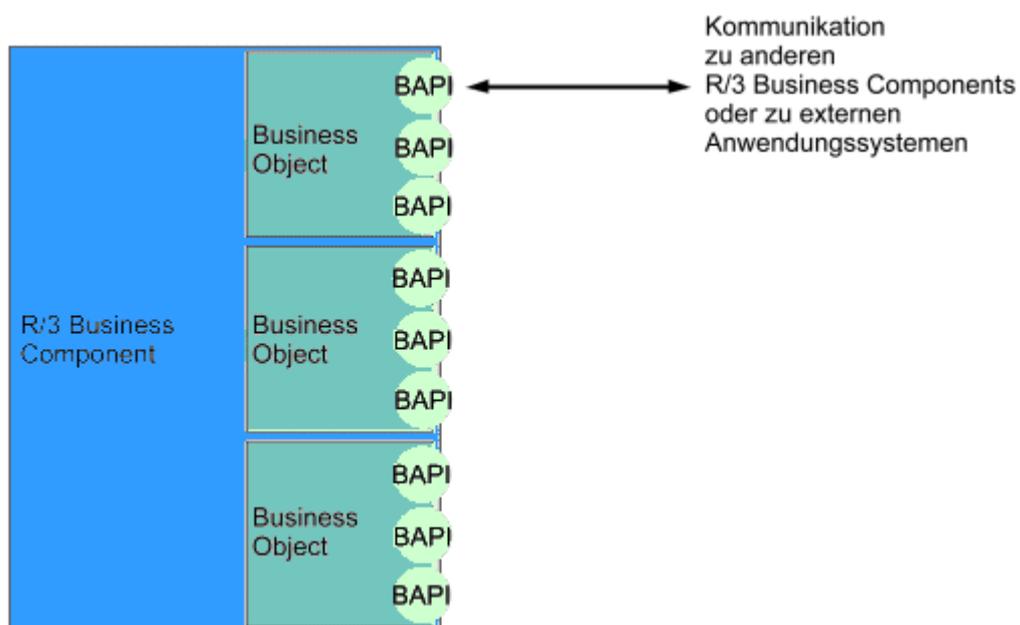


Abbildung 4: Business Objekte – BAPI

Technisch sind BAPIs als remotefähige Funktionsbausteine implementiert.

5 XM – Schnittstellen (eXternal SAP Management)

Die XM-Schnittstellen (eXternal SAP Management) sind ein Schnittstellenpaket für externe Systemmanagement-Programme. SAP-seitig bestehen diese Schnittstellen aus einer Sammlung von BAPIs. Externe Programme können die Bausteine via RFC (Remote Function Call) aufrufen.

Das Schnittstellenpaket besteht aus folgenden Schnittstellen:

- **XMI** externe Monitoring-Schnittstelle für die Protokollierung
- **XAL** externe Schnittstelle für Alert-Management
- **XBP** externe Schnittstelle für die Hintergrundverarbeitung
- **XBR** externe Schnittstelle für Backup und Recovery
- **XMB** externe Schnittstelle für die Grundlagen des Monitoring
- **XOM** externe Schnittstelle für Output-Management

Bei den XM-Schnittstellen spielt die XMI-Schnittstelle eine wesentliche Rolle:

Die XMI-Schnittstelle ist Teil einer gemeinsamen Schicht, die sich alle externen Schnittstellen teilen. Aus Sicherheitsgründen werden die Aktivitäten, die ein externes Programm im SAP-System durchführt protokolliert. Die Aufgabe der Protokollierung übernimmt die XMI-Schnittstelle.

Bevor man eine externe Schnittstelle (z.B. die XAL-Schnittstelle) benutzen kann, muss zuerst eine XMI-Session für diese Schnittstelle geöffnet werden. Dabei wird der Name des externen Programms gesichert und die Versionsnummer, die das externe Programm bei der Schnittstelle voraussetzt, überprüft.

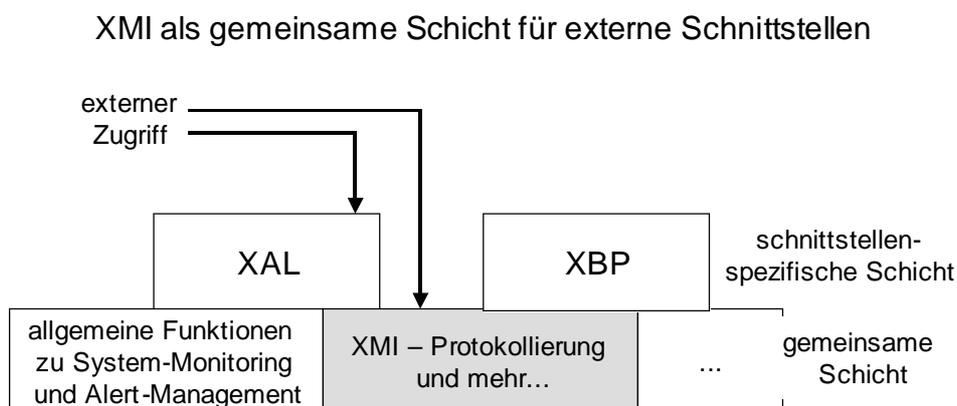


Abbildung 5: XM-Schnittstellen

5.1 Remote Function Call (RFC)

RFC (Remote Function Call) ist eines der wichtigsten SAP-Schnittstellenprotokolle. Damit wird die Programmierung von Kommunikationsabläufen zwischen Systemen wesentlich vereinfacht. Mit RFCs können vordefinierte Funktionen auf einem entfernten System - oder innerhalb des gleichen Systems - aufgerufen und ausgeführt werden.

Funktionsweise:

Bei externen Systemmanagement-Programmen ist das Programm gegenüber einem SAP-System ein Klient, dem ein SAP-Applikationsserver einen Service zur Verfügung stellt. Dieser Service umfasst z.B. die Lieferung von internen Systeminformationen oder die Ausführung einzelner Systemmanagement-Funktionen.

RFC arbeitet in einzelnen Sessions - der Anwender öffnet eine Session, führt die entsprechenden Aufgaben durch und schließt die Session wieder. Die Öffnung erfordert dabei eine reguläre Anmeldung an dem SAP-System mit Benutzer und Kennwort, mit dem er vom SAP-System erkannt wird. Dieser Benutzer muss also innerhalb des SAP-Systems existieren und für die gewünschten Aktivitäten der RFC-Session über die notwendigen Berechtigungen verfügen.

Eine Session zwischen einem SAP-System und einem externen Systemmanagement-Programm besitzt folgenden Aufbau: Eine RFC-Session umfasst eine oder mehrere XMI-Sessions. Jede XMI-Session umfasst eine Reihe von Funktionsaufrufen der Funktionsgruppe der jeweiligen Schnittstelle:

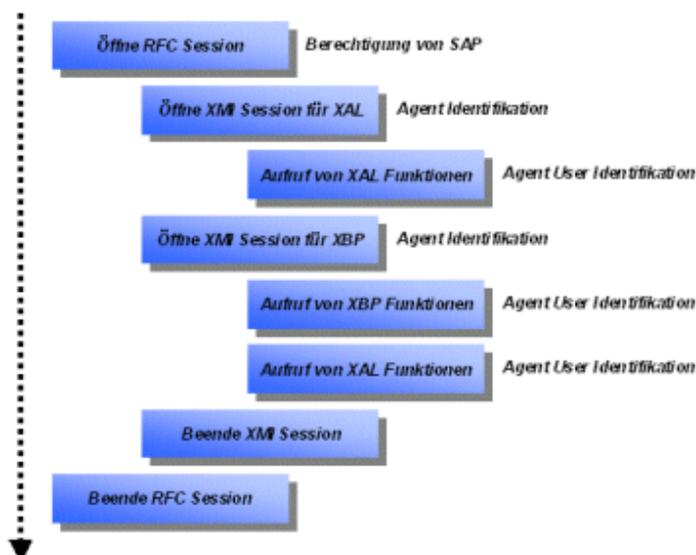


Abbildung 6: Session zwischen einem SAP-System und externen Systemmanagement-Programm

6 SAP Monitoring Shared Library

Teil der Aufgabenstellung war die Erstellung einer Funktionsbibliothek (Shared Library), mit deren Hilfe man komplexe Systemlandschaften über ein einfaches API überwachen kann.

Die, im Rahmen dieser Arbeit entwickelte, SAP Monitoring Shared Library bietet eine einfache Schnittstelle zur Benutzung der CCMS-Daten und somit Informationen zur Überwachung komplexer Systemlandschaften. Sie stellt die Informationen über wenige, einfache Calls zur Verfügung und zeigt sie losgelöst von SAP-Internen, wie interne Tabellen, MTEs, Kontexten ... an.

Technisch gesehen ist die SAP Monitoring Shared Library ein Wrapper um die XAL-Schnittstelle und RFC-Verbindung.

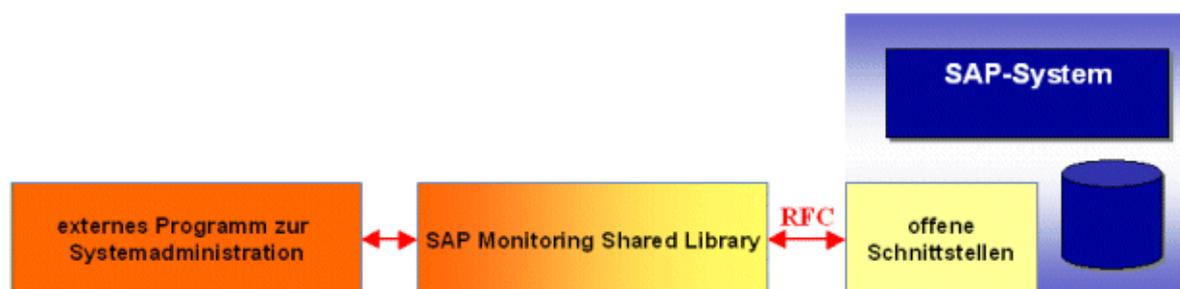


Abbildung 7: SAP Monitoring Shared Library

Die SAP Monitoring Shared Library übernimmt den Verbindungsaufbau zum SAP-System und liefert die gewünschten Informationen. Der Entwickler eines externen Programms zur Systemadministration (Agent) muss sich nicht mit den Details der RFC-Verbindung auseinandersetzen und kann die Monitoring-Daten mit einer einfachen API abfragen.

Die SAP Monitoring Shared Library stellt für Entwickler von Agenten eine extreme Arbeitserleichterung dar, da sie nur noch die Daten spezifizieren müssen, welche sie sammeln wollen und sich nicht mit den eigentlichen Sammelprozess auseinandersetzen müssen.

6.1 Datensammlung von Systemlandschaften

Um Informationen einer Systemlandschaft darstellen zu können, müssen die Informationen der einzelnen Systeme gesammelt werden. Die SAP Monitoring Shared Library übernimmt gegenüber dem SAP-Applikationsserver die Rolle des Klienten. Der SAP-Applikationsserver stellt der SAP Monitoring Shared Library einen Service zur Verfügung, der die Lieferung der Systeminformationen umfasst.

Dabei unterscheidet man zwei Datensammlungs-Ansätze:

- Zentrale Datensammlung von Systemlandschaften
- Dezentrale Datensammlung von Systemlandschaften

6.1.1 Zentrale Datensammlung von Systemlandschaften

Abbildung 8 zeigt das Schema der zentralen Datensammlung von Systemlandschaften:

In der CCMS-Monitoring Infrastruktur sammelt jede Komponente seine Überwachungsdaten selber und speichert sie lokal im Hauptspeicher – dem Monitoring Segment – ab.

Bei der zentralen Datensammlung von Systemlandschaften erfolgt die Datensammlung über ein zentrales Monitoring-System:

- Ein SAP-System der Systemlandschaft wird als zentrales Monitoringsystem ausgewählt.
- Das zentrale Monitoringsystem sammelt die Überwachungsdaten der Komponenten.
- Aus Performance-Sicht fällt für das zentrale Monitoring-System nur unerheblich mehr Last an, weil die Ermittlung der Überwachungsdaten generell dezentral erfolgt und nur die gesammelten Daten angefordert werden.

Der Agent kommuniziert nur mit dem zentralen System, das die anderen Systeme überwacht. Der Agent loggt sich in das zentrale System ein und sammelt die Daten von allen Systemen, wobei das zentrale System in SAP konfiguriert sein muss.

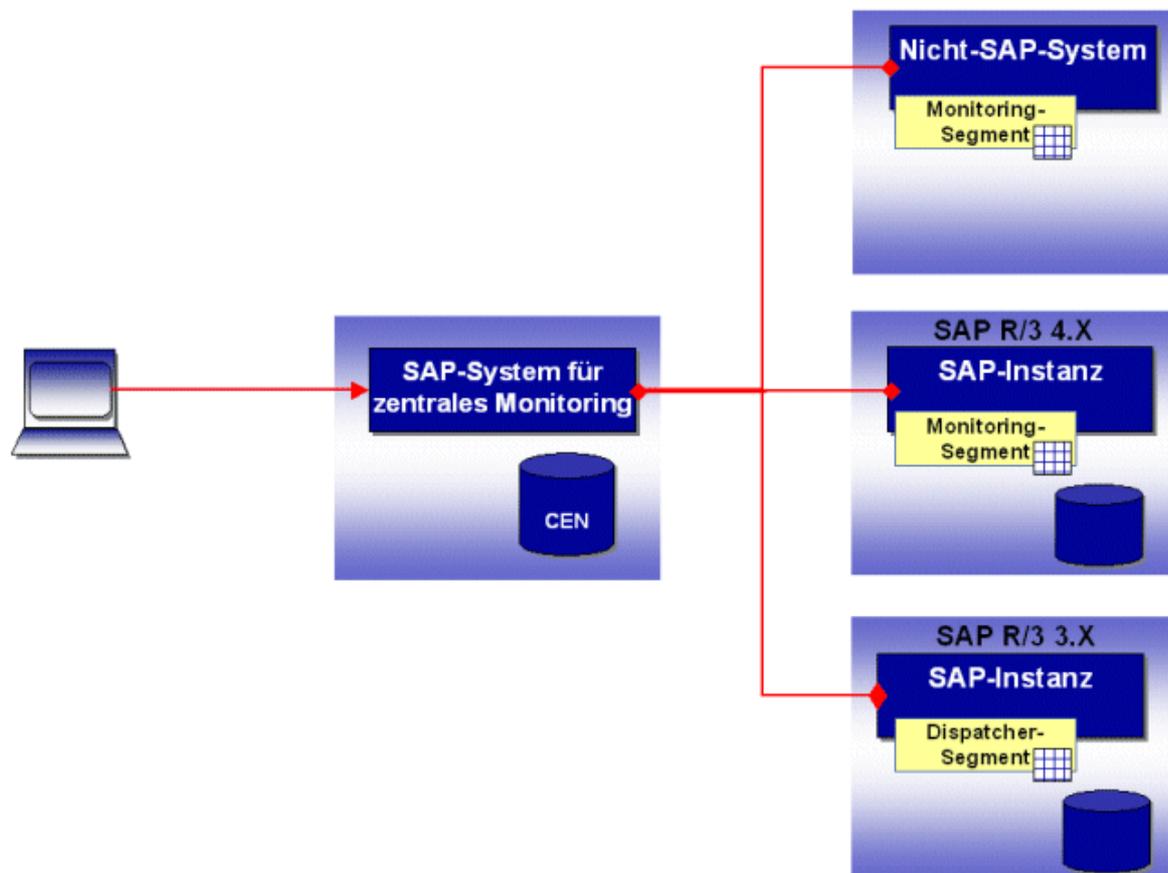


Abbildung 8: Zentrale Datensammlung von Systemlandschaften

Vorteile:

- Zentrale Datensammelstelle: Der Agent muss nur mit dem zentralen System kommunizieren.
- SAP übernimmt Datensammlung der gesamten Systemlandschaft.
- Bessere Performanz durch weniger Overhead (Verbindsaufbau und Datensammlung in nur einem System).
- Nutzung der Konfiguration in anderen Anwendungen (z.B. Solution Manager).

Nachteile:

- Konfiguration des zentralen Systems in SAP

6.1.2 Dezentrale Datensammlung von Systemlandschaften

In Abbildung 9 wird die dezentrale Datensammlung von Systemlandschaften gezeigt:

Auch hier sammelt jede Komponente ihre Überwachungsdaten selber, doch wird auf das zentrale Monitoring-System verzichtet.

Bei diesem Ansatz sind alle Systeme gleichberechtigt, der Agent muss sich in jedes System einloggen, die Daten sammeln und wieder ausloggen. Für jedes System müssen Anmeldedaten für den Agenten gepflegt sein.

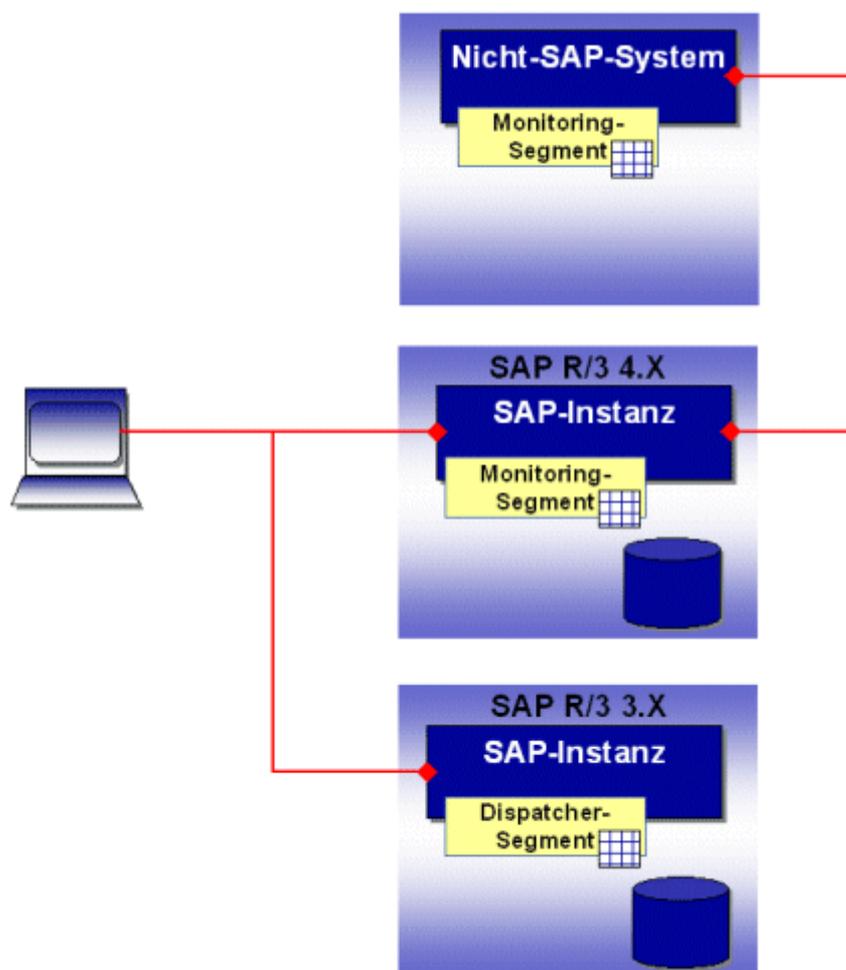


Abbildung 9: Dezentrale Datensammlung von Systemlandschaften

Nicht-SAP-Systeme werden durch SAP-Agenten überwacht, wobei ein Agent immer an ein SAP-System gebunden ist.

Vorteile:

- Direkt, ohne weitere Konfiguration, nutzbar.

Nachteile:

- Agent muss die Datensammlung der Systemlandschaft übernehmen.
- Schlechtere Performanz, da sich der Agent in jedes System einloggen, die Daten sammeln und wieder ausloggen muss.
- Konfiguration der Datensammlung ist nur vom Agenten nutzbar.

Beide Ansätze können mit der SAP Monitoring Shared Library genutzt werden, wobei die Ansätze auch kombiniert werden können. Welche Ansätze bei der Datensammlung verwendet werden, wird über die Konfigurationsdateien konfiguriert.

6.2 Konfigurationsdateien der SAP Monitoring Shared Library

Die SAP Monitoring Shared Library sammelt Systeminformationen kompletter Systemlandschaften. Sie ist eine universelle Bibliothek, mit der beliebige Systeminformationen aus beliebigen SAP-Systemen gesammelt werden können. Damit sie universell einsetzbar ist, muss sie konfiguriert werden. Dies geschieht über zwei Konfigurationsdateien, die üblicherweise in `/etc/sapmon` liegen:

- `agent.cfg`: Spezifikation, welche Informationen gesammelt werden.
- `login.cfg`: Spezifikation, aus welchem System die Informationen gesammelt werden.

6.2.1 `agent.cfg`

In der Konfigurationsdatei `agent.cfg` wird spezifiziert, welche Daten aus dem R/3-System übertragen werden.

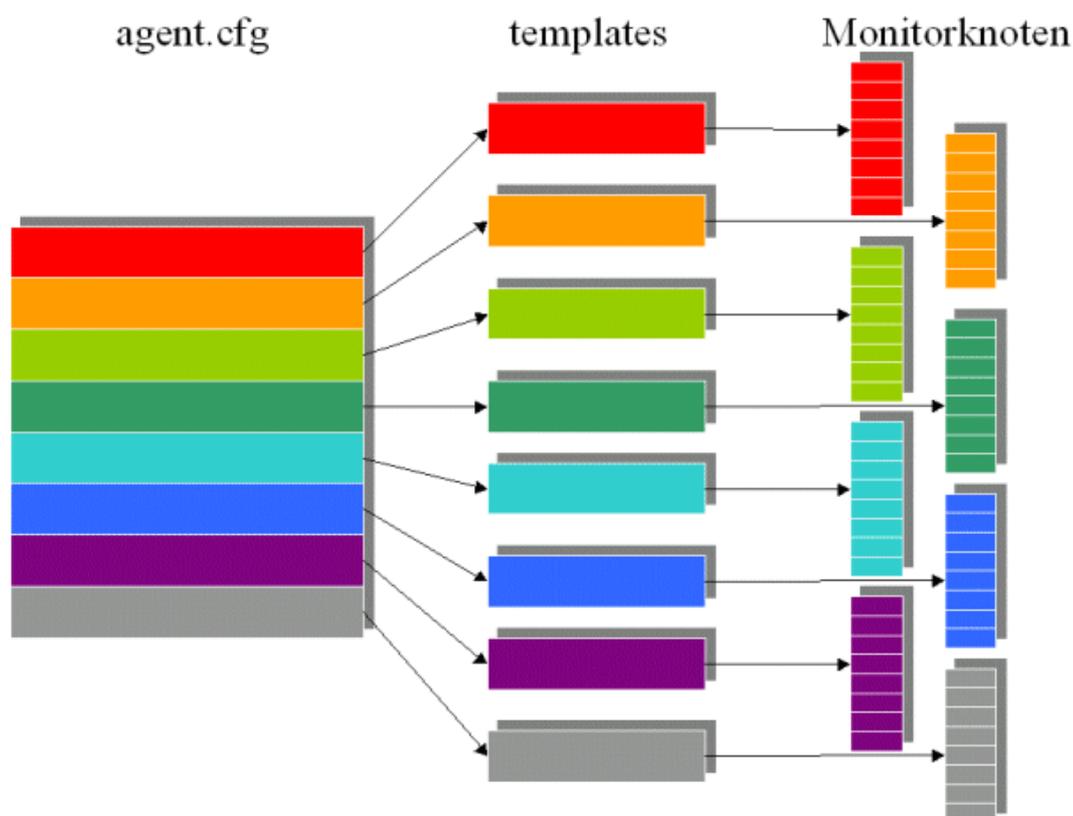


Abbildung 10: Aufbau der Konfigurationsdatei `agent.cfg`

Die Konfigurationsdatei `agent.cfg` besteht aus einem oder mehreren Templates. Jedes Template spezifiziert eine Beschreibung zur Übertragung verschiedener Monitorknoten. Mit Hilfe der Templates ist es möglich Monitorknoten zu einer sinnvollen Einheit zusammenzufassen, die in einem logischen Zusammenhang stehen. So kann man z.B. ein Template spezifizieren, welches die Dateisysteme aller Applikationsserver eines bestimmten Systems anzeigt.

Man unterscheidet zwei Typen von Monitor-Templates:

- einfache Monitortemplates
- erweiterte Monitortemplates

6.2.1.1 Einfache Monitortemplates

Das einfache Monitortemplate bietet eine Schnittstelle ohne direkten CCMS-Bezug.

Ein einfaches Template besitzt folgenden Aufbau:

```
[TEMPLATE_<NAME>]
DESCRIPTION = Beschreibung des Templates           z.B. Einfaches Template
SYSTEM      = <SID>                                z.B. BCE | BC? | B*
APPL-SERVER = <APPL-SERVER>                        z.B. bcemain_BCE_26 | bce* | *
VALUE       = <VALUE>                              z.B. DIALOG_RESPONSE_TIME
```

z.B.

```
[TEMPLATE_TEST]
SYSTEM      = BCE
APPL-SERVER = *
VALUE       = DIALOG_RESPONSE_TIME
```

Das Template TEST liefert die Dialogantwortzeit aller Applikationsserver des Systems BCE.

6.2.1.2 Erweiterte Monitortemplates

Das erweiterte Monitortemplate bietet eine Schnittstelle, die der CCMS-Nomenklatur entspricht.

Die CCMS-Monitoring-Überwachungsdaten liegen im Monitoring-Segment des Systems. Die Anzeige der Überwachungsdaten geschieht über Monitore. Diese stellen eine Sicht auf das Monitoring-Segment dar.

Ein erweitertes Template besitzt folgenden Aufbau:

```
[TEMPLATE_<NAME>]
DESCRIPTION      = Beschreibung des Templates                z.B. Erweitertes Template
MONI_SET_NAME    = <Monitorsammlung>                        z.B. "SAP ADMIN TEMPLATE"
MONI_NAME       = <Monitor der Sammlung>                    z.B. "Operating System"
PATTERN_<m>     = SID\Monitorkontext\Monitorobjekt\Monitorattribut
                  z.B. BCE\bcemain_BCE_26\CPU\*
```

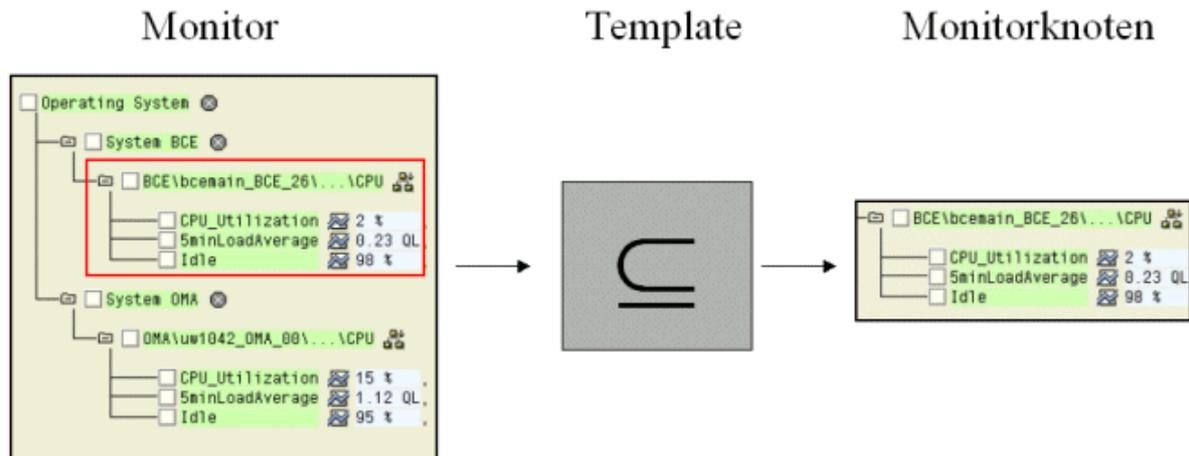


Abbildung 11: Erweitertes Template

Das erweiterte Template beschreibt eine Teilmenge eines SAP-Monitors.

Die Monitorknoten des Monitors MONI_NAME der Monitorsammlung MONI_SET_NAME werden selektiert. Da meistens nicht alle Knoten eines Monitors von Interesse sind, können Pattern spezifiziert werden, die die Auswahl einschränken (Jeder Monitorknoten ist eindeutig durch ein Viertupel bestehend aus System-ID\Monitorkontext\Monitorobjekt\Monitorattribut identifizierbar).

In unserem Beispiel enthält der Monitor Informationen zu CPU-Werten der Systeme BCE und OMA. Durch das Pattern `BCE\bcemain_BCE_26\CPU*` werden nur die Monitorknoten der CPU-Werte des Applikationsservers bcemain ausgewählt.

Die CCMS-Monitoring-Architektur erlaubt ein einfaches Anlegen eigener Monitore. Damit können die Monitore bereits auf SAP-Ebene eingestellt werden und auf das Spezifizieren der Pattern verzichtet werden.

6.2.2 login.cfg

Die Verbindung zum R/3-System erfolgt via RFC.

Die Eröffnung einer RFC-Session erfordert eine reguläre Anmeldung am SAP-System mit Benutzer und Kennwort. Der Benutzer muss innerhalb des SAP-Systems existieren und für die gewünschten Aktivitäten der RFC-Session über die notwendigen Berechtigungen verfügen.

Die System- und Logon-Daten werden in der `login.cfg` hinterlegt.

Die Konfigurationsdatei `login.cfg` besteht aus einem oder mehreren Templates. Jedes Template spezifiziert eine Verbindung zu einem SAP-System und besitzt folgenden Aufbau:

```
[LOGIN_<NAME>]  
LOGIN=<Argumente>
```

z.B.

```
[LOGIN_BCE]  
LOGIN=-d BCE -u user -p password -h bcemain -s 26
```

Das Template BCE enthält die Logon-Daten des users `user` mit Passwort `password` für das System BCE mit Systemnummer 26 auf Host `bcemain`.

Durch die Aufspaltung in zwei Konfigurationsdateien können die gleichen Monitoring-Templates für verschiedene Systeme verwendet werden, d.h. man kann mit einem Monitoring-Template die gleichen Daten aus verschiedenen Systemen anfordern.

6.3 SAP Monitoring Shared Library – API

6.3.1 Datentypen

`AlMoniHandle_T`: Definiert ein Handle, das ein Template in der `agent.cfg` repräsentiert. Über das Handle können Informationen zum Template und zu Monitorknoten angefordert werden.

`AlMoniNode_T`: Definiert einen Monitorknoten.
Die Struktur enthält folgende Informationen:

- Name des Systems
- Name des Monitorkontextes
- Name des Monitorobjekts
- Name des Monitorattributs

- Datum und Uhrzeit des Auftreten eines Fehlers
- Datum und Uhrzeit des zuletzt gemeldeten Wertes

- Aktuelle Alertschwere und Gewichtung

- Aktueller Wert

6.3.2 Funktionen

```
int sapmon_init (AlMoniHandle_T **MoniHandle, int *NrOfMoniTempl);
```

Liest die Konfigurationsdatei `agent.cfg` ein und legt für jedes Template ein `AlMoniHandle_T` an.

```
int sapmon_connect (char *Login_Template, RFC_HANDLE *rfc_handle);
```

Öffnet eine R/3-Verbindung zu dem in `Login_Template` spezifizierten System.

```
int sapmon_disconnect (RFC_HANDLE *rfc_handle);
```

Schliesst die R/3-Verbindung.

```
int sapmon_request(RFC_HANDLE *rfc_handle, AlMoniHandle_T *MoniHandle, int NrOfMoniTempl);
```

```
int sapmon_refresh(RFC_HANDLE *rfc_handle, AlMoniHandle_T *MoniHandle, int NrOfMoniTempl);
```

Mit `sapmon_request` werden die Monitorknoten aus dem R/3 geladen und die Werte gefüllt.

Über `sapmon_refresh` werden der Werte der Monitorknoten aktualisiert.

```
int sapmon_request_tree (AlMoniHandle_T MoniHandle, AlMoniNode_T **nodes, int *NrOfNodes);
```

Gibt Informationen zu den Monitorknoten eines Templates.

Für jedes Template bekommt man ein Array von Monitorknoten zurück.

```
char * sapmon_get_template_description(AlMoniHandle_T MoniHandle);
```

```
char * sapmon_get_template_name      (AlMoniHandle_T MoniHandle);
```

Geben Informationen zu den Templates.

```
int sapmon_sap_start_ssh(char *sid, char *hostname, char *nr);
```

```
int sapmon_sap_stop_ssh (char *sid, char *hostname, char *nr);
```

```
int sapmon_sap_start   (char *sid, char *hostname, char *nr);
```

```
int sapmon_sap_stop    (char *sid, char *hostname, char *nr);
```

Starten und Stoppen von R/3: Lokal oder via Secure Shell (ssh).

6.3.3 Beispielprogramm

```

int main(int argc, char **argv) {
    int i,j;

    RFC_HANDLE      rfc_handle;
    AlMoniHandle_T  *MoniHandle;
    int             NrOfMoniTempl;

    AlMoniNode_T    *nodes;
    int             NrOfNodes;

    if (argc < 2) {
        usage(argv[0]);
        return AG_INVALID_ARGUMENTS;
    }

    /* Read agent.cfg */
    if (sapmon_init (&MoniHandle, &NrOfMoniTempl) >= AG_FIRST_ERROR) {
        printf("Could not initialize INI-File\n");
        return AG_CALL_INVALID;
    }

    /* Log on to system specified in arguments */
    if (sapmon_connect (argv[1], &rfc_handle) >= AG_FIRST_ERROR) {
        printf("Could not log on\n");
        return AG_CALL_INVALID;
    }

    for (i = 0; i < NrOfMoniTempl; i++) {
        sapmon_request (&rfc_handle, &(MoniHandle[i]), 1 );
        sapmon_request_tree (MoniHandle[i], &nodes, &NrOfNodes);

        for (j = 0; j < NrOfNodes; j++) {
            printf("MO: %s, MA: %s, Value %s\n",
                nodes[j].Objectname, nodes[j].Mtnameshrt, nodes[j].currentValue);
        }
    }

    while(running) {
        for (i = 0; i < NrOfMoniTempl; i++) {
            sapmon_refresh ( &rfc_handle, MoniHandle, NrOfMoniTempl );
            sapmon_request_tree (MoniHandle[i], &nodes, &NrOfNodes);

            for (j = 0; j < NrOfNodes; j++) {
                printf("MO: %s, MA: %s, Value %s\n",
                    nodes[j].Objectname, nodes[j].Mtnameshrt,
                    nodes[j].currentValue);
            }
        }
        sleep(10);
    }

    /* Log off */
    sapmon_disconnect (&rfc_handle);

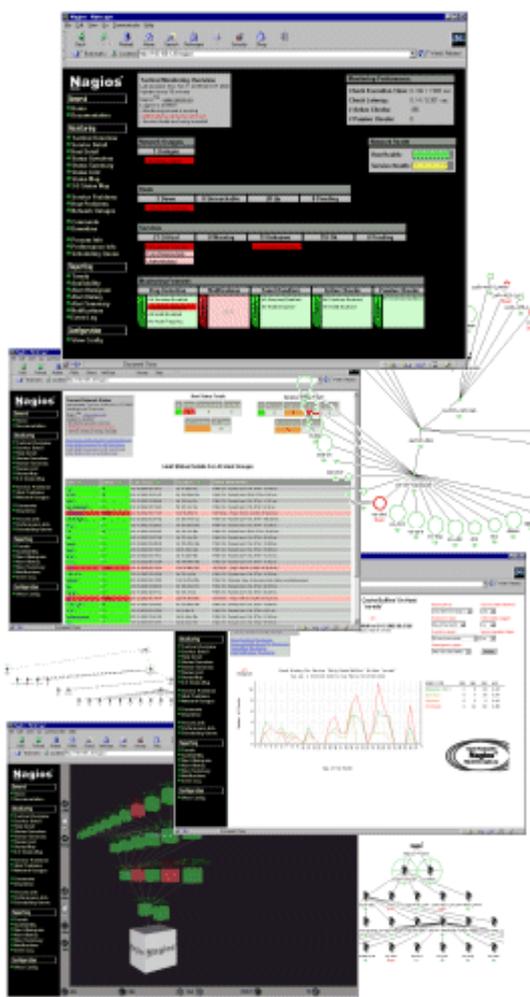
    return SAP_OK;
}

```

Das Programm liest die agent.cfg ein und baut eine Verbindung zum angegebenen System auf.

Für jedes Template werden die Monitorknoten aus dem SAP-System eingelesen und alle 10 Sekunden aktualisiert.

7 Nagios



Nagios ist eine generische, auf Open-Source basierende, Monitoring-Lösung. Die Software überwacht ausgewählte Rechner und deren Dienste. Wenn ein Dienst nicht mehr funktioniert, angegebene Schwellenwerte über- oder unterschritten werden oder Rechner nicht mehr erreichbar sind, können Meldungen (eMails, Pager, etc) versendet werden.

Unter Microsoft Windows steht mit der Microsoft Management Console ein etabliertes Monitoring-Tool zur Verfügung. Unter Linux und Unix hat sich noch kein universeller Standard herausgebildet.

Teil der Aufgabe war eine Beispiel-Implementierung der SAP Monitoring Shared Library. Das Ziel war nicht „yet another monitoring tool“ zu entwickeln, welches nur die Funktionalität der Monitoring Shared Library demonstriert, sondern eine existierende Lösung um die SAP Überwachungsfunktionalität zu erweitern.

Die Wahl fiel auf Nagios, da es die Möglichkeit eigene Plugins zu schreiben, große Funktionalität und Flexibilität bietet, eine ansprechende Oberfläche besitzt und Open-Source ist, was auch Modifikationen am Kernprogramm erlaubt.

7.1 Nagios – Plugins

Das Nagios-Kernprogramm übernimmt die administrativen Aufgaben wie das Starten und Stoppen von Datensammelmethode, Generierung und Archivierung von Störmeldungen und Versenden von Meldungen.

Die eigentliche Datensammlung wird von Plugins durchgeführt. Plugins sind eigenständige Programme, die Komponenten überwachen und ihre Überwachungsdaten an Nagios weitergeben.

Mit Hilfe der SAP Monitoring Shared Library wurden im Rahmen der Arbeit vier Nagios-Plugins geschrieben.

<pre>check_sap <Monitor-Template> <RFC-Template></pre>	Sammelt die Überwachungsdaten, die im <Monitor-Template> spezifiziert sind aus dem System, das in <RFC-Template> gepflegt ist. <code>check_sap</code> dient zur Anzeige eines einfachen Wertes.
<pre>check_sap_multiple <Monitor-Template> <RFC-Template></pre>	Sammelt die Überwachungsdaten, die im <Monitor-Template> spezifiziert sind aus dem System, das in <RFC-Template> gepflegt ist. <code>check_sap_multiple</code> dient zur Anzeige mehrerer Werte.
<pre>check_sap_system <SID> <RFC-Template></pre>	Listet die Applikationsserver des Systems <SID> und den Status der Applikationsserver auf.
<pre>check_sap_instance <Applikationsserver> <RFC-Template></pre>	Liefert Standard-Daten (Dialogantwortzeit, Anzahl der eingeloggt Benutzer) zum Applikationsserver.

Tabelle 1: Nagios Plugins

Um die Vorteile der Konsole zu nutzen, wurde zu jedem Plugin jeweils ein Programm mit dem Suffix „_cons“ erstellt. Diese Konsolen-Datensammler sammeln diesselben Daten, stellen sie aber für die Konsole aufbereitet dar.

Vorteile der Konsolen-Datensammler sind:

- Stand-alone Programme: Die Datensammler sind direkt, ohne eine Installation und Konfiguration von Nagios, nutzbar.
- Direktes Ausgabe: Die Datensammler werden direkt aufgerufen und unterliegen nicht dem Anstartverhalten von Nagios (periodisches Sammeln in bestimmten Intervallen).
- Höhere Geschwindigkeit und geringere Netzlast: Es werden nur die reinen Informationen gesendet (keine Layoutinformationen).
- Skriptfähig

7.2 Nagios – Core – Anpassungen

Nagios soll nicht nur ein Monitoring-Tool sein, welches Werte überwacht und Störmeldungen generiert, sondern ein System-Management-Tool, welches den Administrator aktiv bei seiner Aufgabe unterstützt. Deswegen wurde zusätzlich zu den Monitoring-Plugins das Nagios-Core-Programm um SAP-Funktionalität erweitert.

7.2.1 Host-Commands-Menü

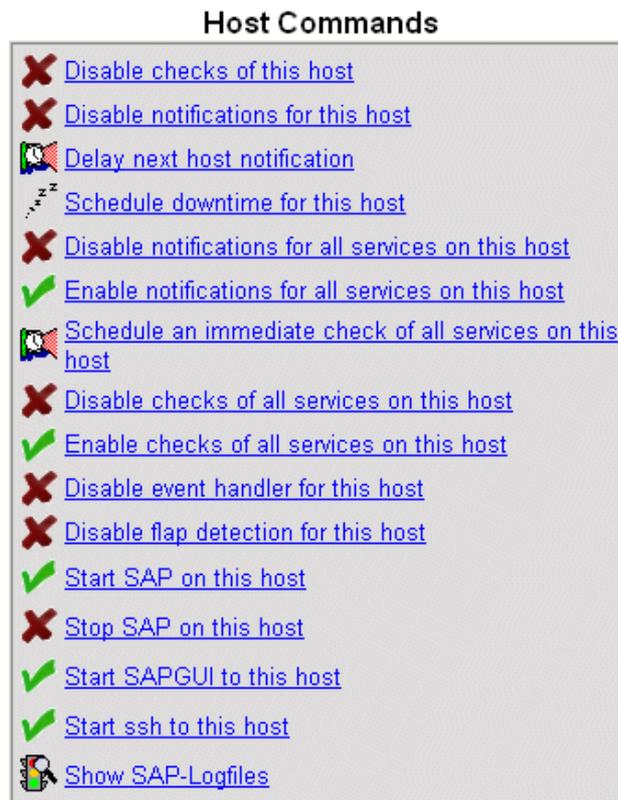


Abbildung 12: Host Commands Menü

Über das Host Commands Menü stehen für SAP-Systeme neue Funktionen zur Verfügung

Start SAP on this host	Startet das SAP-System auf dem Host
Stop SAP on this host	Stoppt das SAP-System auf dem Host
Start SAPGUI to this host	Öffnet eine SAPGUI zu dem SAP-System
Start ssh to this host	Öffnet eine Secure Shell zu dem Host
Show SAP-Logfiles	Zeigt die SAP-Logfiles an

Tabelle 2: Host Command Menü

7.2.2 Schwellenwertänderung

Nagios übernimmt die Alert-Einstufung direkt aus dem SAP-System, d.h. es werden die CCMS-Schwellenwerte verwendet.

Man kann direkt von Nagios aus die Schwellenwerte ändern; wodurch auch die die Schwellenwerte des CCMS geändert werden. Hiermit erhält man eine konsistente Alert-Einstufung.

Command Options

Threshold GREEN to YELLOW: msec	<input type="text" value="2000"/>
Threshold YELLOW to RED: msec	<input type="text" value="3000"/>
Threshold RED to YELLOW: msec	<input type="text" value="2000"/>
Threshold YELLOW to GREEN: msec	<input type="text" value="1000"/>
Node specific:	<input type="checkbox"/>
	<input type="button" value="Commit"/> <input type="button" value="Reset"/>

Abbildung 13: Schwellenwertänderung

8 Glossar

Alert	Störmeldung im CCMS-Alert-Monitor und in der Monitoring-Architektur. Alerts lenken die Aufmerksamkeit des Systemadministrators auf kritische oder potenziell gefährliche Situationen in einem SAP-System. Der CCMS-Alert-Monitor wertet die von den Data Suppliers gemeldeten Werte aus und vergleicht sie mit den vordefinierten Alert-Schwellenwerten. Wenn die eingehenden Daten gegen einen Schwellenwert verstoßen, löst der Alert-Monitor einen Alert aus.
Alert-Monitor	Standard-Data-Consumer, der in einer hierarchischen Struktur die Messwerte und die Alerts der überwachten Monitorobjekte und -attribute anzeigt.
BAPIs	Die B usiness A pplication P rogramming I nterfaces ermöglichen als Methoden an den Business-Objekte den objektorientierten Zugriff auf das SAP-System. Zusammen mit den Business-Objekte definieren und dokumentieren die BAPIs den Schnittstellenstandard auf betriebswirtschaftlicher Ebene.
Business-Objekte	Business-Objekte sind die Grundlage der objektorientierten Strukturierung des SAP-Systems. Ein einzelner Business-Objekttyp ist das Abbild eines betriebswirtschaftlichen Sachverhaltes. Er umfaßt sowohl die Funktionalität als auch die Daten dieses Sachverhaltes.
Data Consumer	Programm, das die Daten aus der Monitoring-Architektur verwendet und anzeigt (z. B. die CCMS-Monitore).
Data Supplier	Programm oder Transaktion, das Informationen über das SAP-System und seine Umgebung sammelt und diese in der CCMS-Monitoring-Architektur registriert.
Monitor	Sammlung von MTEs, die in einem hierarchischen Baum angeordnet sind.
Monitorattribut	Grundelement innerhalb des Monitorbaums. Beschreibt die Eigenschaften der Monitorobjekte; es gibt die folgenden Arten von Monitorattributen: <ul style="list-style-type: none">• Performance-Attribute• Statusattribute• Protokollattribute• Textattribute
Monitorbaum	Baumstruktur, die die von der Monitoring-Architektur überwachten Systemkomponenten zeigt. Je nach Definition kann ein Monitorbaum nur einen Teil oder alle der verfügbaren Knoten (oder MTEs) anzeigen.

Monitorbaumelement (MTE)	Knoten im Monitorbaum, von denen drei Arten existieren: <ul style="list-style-type: none">• Monitorattribute• Monitorobjekte• Sammelknoten
Monitorkontext	Gruppe von Monitorobjekten, die unter einer Überschrift zusammen in einem Baum angeordnet sind. Jeder unabhängige Baum bildet einen Monitorkontext innerhalb eines Monitorsegments.
Monitorobjekt	MTE, das verschiedene Monitorattribute enthält, die zu demselben realen Objekt gehören (z. B. Festplatte oder Komponenten im SAP-System).
Monitorsammlung	Reihe von Monitoren, die in einem hierarchischen Baum angezeigt werden. Mit Monitorsammlungen werden die einzelnen Monitore visuell gruppiert.
Monitoring-Segment	„Container“ mit allen aktiven Komponenten und Daten, die in der Monitoring-Architektur benötigt werden. Das Monitorsegment wird durch zwei technische Implementierungen realisiert: <ul style="list-style-type: none">• gemeinsamer Speicher (speicherorientierte Datei)• Reihe von Tabellen in der Datenbank
Performance-Attribut	Monitorattribut, das Performance-Werte sammelt und mittelt, die der Monitoring-Architektur gemeldet wurden. Wenn diese Werte gegen die eingestellten Schwellenwerte verstoßen, wird ein Alert ausgelöst.
Protokollattribut	Monitorattribut, das wie ein Statusattribut Nachrichten mit einer Alert-Bewertung enthält. Ein Protokollattribut kann mehrere Nachrichten enthalten. Es liefert das Mittel, um Protokoll- und Trace-Dateien für die Aufnahme in der Monitoring-Architektur zu instrumentieren.
Schwellenwert	Wert, dessen Über- oder Unterschreitung in einem Performance-Attribut einen Alert auslöst.
Statusattribut	Monitorattribut, in das jeweils ein Fehlermeldungstext und Alert-Status gemeldet werden kann.
Textattribut	Monitorattribut, in das ein Data Supplier technische Informationen melden kann. Der Text kann nach Bedarf aktualisiert werden. Es können jedoch keine Werte gemeldet und keine Alerts ausgelöst werden.