
Integration eines Ticketsystems am Beispiel OTRS

Oldenburg, 2008-06-10

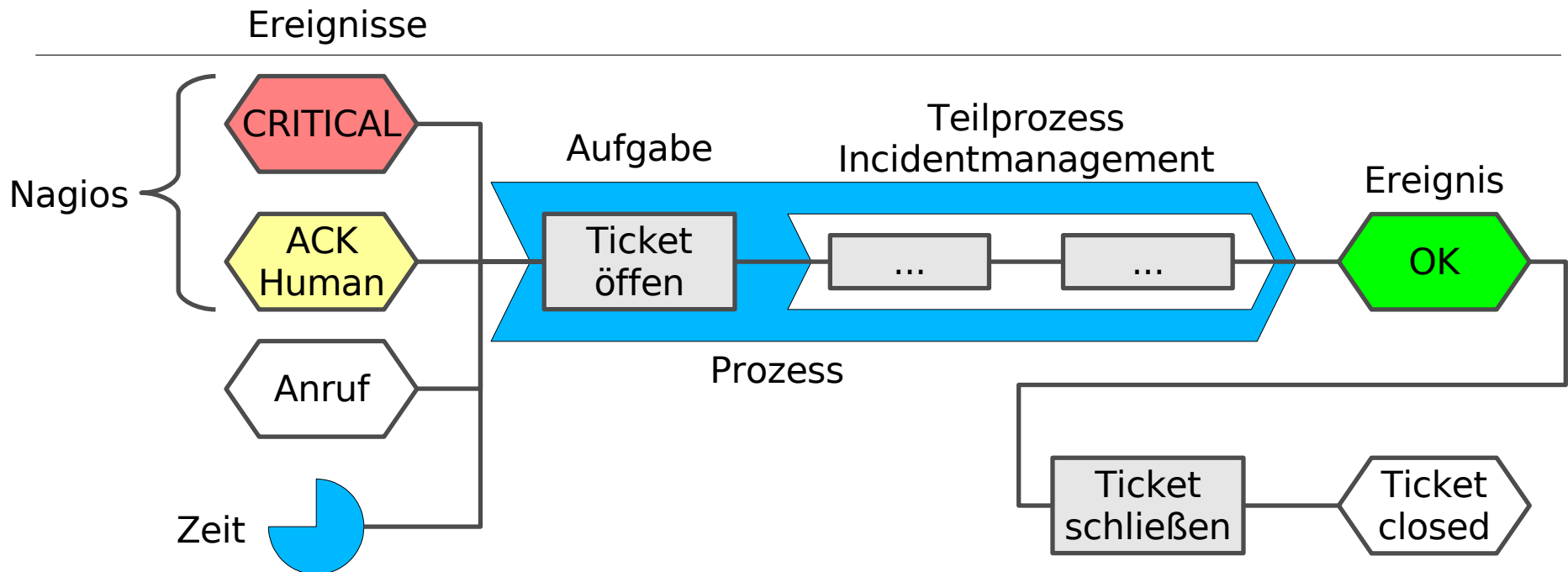
Motivation

- Mini-Workflow in Nagios mit Kommentaren möglich-> "Notizbuch"
- ein Ticketsystem bietet mehr:
 - Delegieren von Aufgaben
 - Kommunikation (meist per E-Mail)
 - automatische Zuordnung von Rückfragen
 - Zeiterfassung
 - Protokollfunktion
 - Auswertungsmöglichkeiten
 - ...

Überblick

- Grundlagen
 - Prozesse, Benachrichtigungen, Event Handler
- Ticket-Integration im Detail
 - Ticket erzeugen
 - Ticket schließen
- OTRS::SystemMonitoring
 - OTRS-Grundlagen
 - Einfache Ticket-Handling per E-Mail

Prozesse

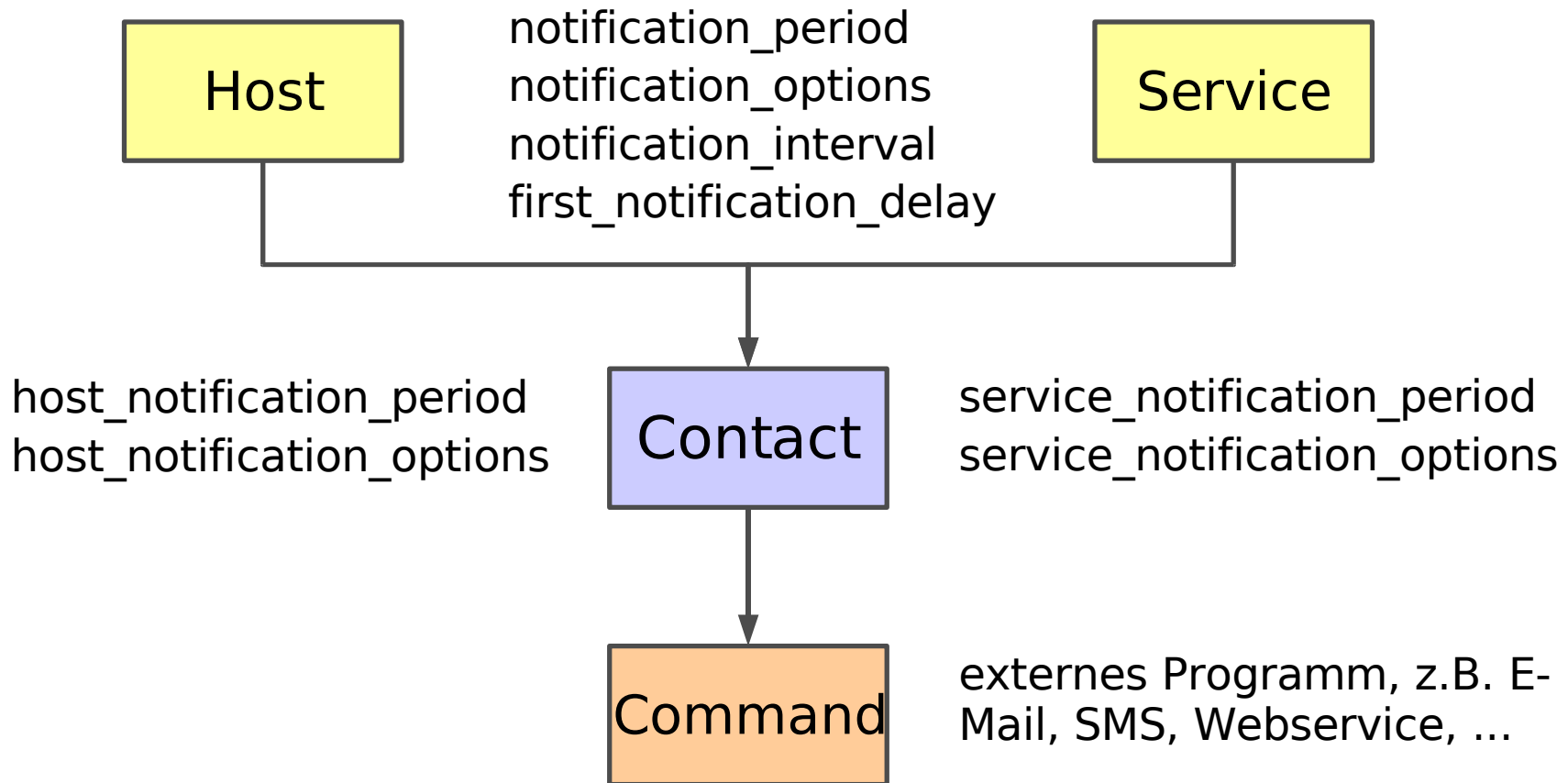


- Ein Prozess ist eine durch 1-n Ereignisse angestoßene Aufgabenkette, die 1-n Ereignisse und 1-n Leistungen hervorbringt.
- Starterereignis: Problem aufgetreten
- Zielereignis:
 - Problem beseitigt, *oder*
 - Ticket geschlossen

Nagios ruft OTRS

- Wie teilt Nagios OTRS mit, dass ein Ereignis stattgefunden hat?
 - Benachrichtigungen (Notifications)
 - Event-Handler

Benachrichtigungen I



Benachrichtigungen II

- Verschiedene Steuerungsmöglichkeiten:
 - ob: contact (contact_group)
 - wann: _period, _interval, _delay
 - was: _options
- ACHTUNG:
 - _period bei Host/Service \neq _period bei Contact

Benachrichtigungen III

Stolperstellen

- nur wer eine Problem-Notification bekommt, erhält später eine Recovery-Notification
- Verschiedene Mechanismen unterdrücken Benachrichtigungen:
 - Dependencies
 - Acknowledgements
 - Scheduled Downtime
 - Flapping

Event Handler

- Werden ausgeführt bei:
 - jedem Soft-State
 - neuem Problem-Hard-State
 - jedem Recovery (Soft- und Hard-Recovery)
- Steuerung über:
 - `$SERVICESTATE$` (OK, WARNING, ...)
 - `$SERVICESTATETYPE$` (SOFT, HARD)
 - `$SERVICEATTEMPT$`
 - dito für Hosts
- Finetuning über Skript (nicht über Nagios-Konfiguration)

Ticket erzeugen I

Notification

- Steuerung über Contact
- automatisch:
 - notification_period = 7x24
 - > kein zeitabhängiges Queuing
 - > Andere Benachrichtigungsarten nicht frei konfigurierbar
- Acknowledgement:
 - Idee: Ticket erst bei ACK erzeugen (nicht bei jedem Problem)
 - kein _options-Flag für ACK
=> Skriptabfrage

Ticket erzeugen II

Event Handler

- unabhängig von Benachrichtigungen
- muss bei Host/Service aktiviert werden
 - keine Steuerung über Contact
- evtl. Kollision mit anderen Event-Handler-Aufgaben
 - z.B. Reset eines Dienstes
- Event-Handler-Skript versendet E-Mail an OTRS

Ticket erzeugen III

External Command File

- Selbstprogrammierte Webanwendung
- unabhängig vom Zustand eines Hosts oder Services
- Ergänzende Aufgaben implementierbar
 - Acknowledgement
 - Kommentar
 - Benachrichtigungen
- Berechtigungen müssen manuell gesetzt werden
- URL-Einbindung: Action-URL oder CGI patchen (originales ACK ersetzen)

Ticket schließen I

Benachrichtigung

- Problem-Nachricht muss versandt worden sein, sonst gibt es keine Recovery-Nachricht:
 - first_notification_delay problematisch
 - period muss 7x24h laufen!
 - Filterung Recovery wie Problem State!

Ticket schließen II

Event Handler

- Information über Recovery kommt immer an
- Ideal für automatisches Schließen!
- Event-Handler-Skript versendet E-Mail an OTRS

Ticket schließen III

Manuell in OTRS

- grundsätzlich möglich
- Problem offen, Ticket schließen:
 - anschließender Wechsel z.B. von WARNING auf CRITICAL -> neues Ticket
- mehrere Tickets zum gleichen Vorgang müssen manuell zusammengefasst werden
- Problem erledigt, Ticket schließen:
 - manuelle Endkontrolle erforderlich
 - sinnvoll u.U. für bestimmte Services (z.B. Filesystem-Auslastung)

Menü a la wob

- Services explizit auswählen
 - nicht alles immer ins Ticketsystem schicken
- Ticket erzeugen via Acknowledgement
 - damit man das Problem nicht vergisst
- Ticket schließen per Event-Handler
- Einige Service-Tickets nur manuell schließen:
 - "Filesystem full"

ACK -> Ticket öffnen

```
#!/bin/bash
# notify-otrs.sh
...
case "$NAGIOS_NOTIFICATIONTYPE" in
    ACKNOWLEDGEMENT*)
        # mailtext zusammenbauen
        ...
        printf "%s\n" $body | /usr/bin/mail ...
        ;;
    *)
        exit 0
        ;;
esac
```

Event -> Ticket schließen

```
#!/bin/bash
# otrs-event-ok.sh
...
case "$NAGIOS_SERVICESTATE" in
  OK)
    case "$NAGIOS_SERVICESTATETYPE" in
      HARD)
        # mailtext zusammenbauen
        printf "%s\n" $body | /usr/bin/mail ...
        ;;
      *) ;;
    esac
  *) ;;
esac
```

OTRS (1)

- E-Mail-zentriertes Ticket-System
 - fast alles über E-Mails steuerbar
 - !Security!
- Perl, Apache, Datenbank
- OpenSource
- ursprünglich entstanden bei SuSE
- Installation:
 - best fit: RPMs für SUSE, RedHat + MySQL
 - (k)ubuntu 8.04: aptitude install otrs2 (2.2.4)
 - Sourcen: tar.gz
- Homepage: www.otrs.org

OTRS (2)

- Begriffe:
 - User, Gruppen
 - Queues
 - Rollen
 - Mailadressen
 - Agent
 - Kunde
- Administration: SysConfig
- zusätzliche Pakete: Paketverwaltung
 - 'SystemMonitoring' für Nagios-Anbindung

OTRS::SystemMonitoring

[Paket Verwaltung]

Installieren:

Paket:

Online Repository:

Quelle: [-Master-]

Online Repository:

Name	Version	Anbieter	Beschreibung	Aktion
Benchmark	1.2.5	OTRS AG	Ein einfache Benchmark Applikation.	Installieren
Calendar	1.7.2	OTRS GmbH	Ein Web Kalender.	Installieren
FAQ	1.3.1	OTRS AG	Eine FAQ-/Wissensdatenbank.	Installieren
FileManager	1.3.1	OTRS GmbH	Ein Web Datei Manager mit download/upload moeglichkeit[..]	Installieren
Survey	1.0.17	OTRS GmbH	Ein Kundenumfragen Werkzeug.	Installieren
WebMail	0.11.3	OTRS AG	Ein Web Email (IMAP) Programm.	Installieren

Lokales Repository:

Name	Version	Anbieter	Beschreibung	Status	Aktion
LinkObject2	1.0.5	OTRS AG	Das LinkObject2 Paket.	installiert	Deinstallieren
SystemMonitoring	1.0.2	OTRS GmbH	Einfache Email Schnittstelle zur System Monitoring Suit[..]	installiert	Deinstallieren
SystemStatus	1.5.1	OTRS GmbH	Ein Web System-Status-Monitor.	installiert	Deinstallieren

- ftp-Download: ftp://ftp.otrs.org/pub/otrs/packages
- lokale Installation oder via Online-Repository

OTRS::S.M. Konfiguration

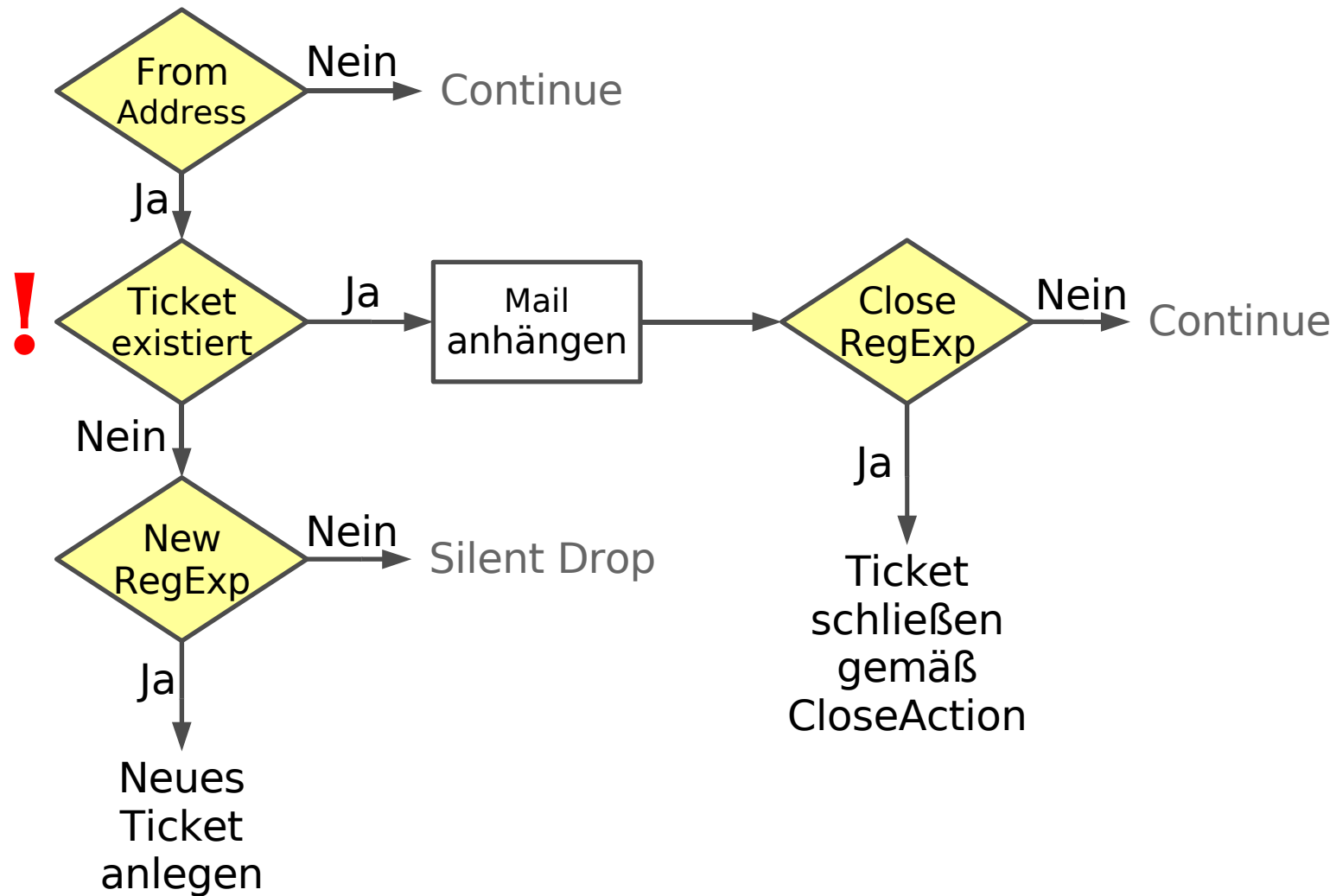
Config Einstellungen: SystemMonitoring -> Core::PostMaster

PostMaster::PreFilterModule###1-SystemMonitoring:
Einfache Email Schnittstelle zu System Monitoring Suites

- ArticleType
- FreeTextHost
- FreeTextService
- Regexp aus E-Mail:
 - Host, Service
 - State
 - FromAddress
- Regexp aus State:
 - NewTicket, CloseTicket

Schlüssel	Inhalt	
ArticleType	email-external	Löschen
CloseActionState	closed successful	Löschen
ClosePendingTime	172800	Löschen
CloseTicketRegExp	OK UP	Löschen
DefaultService	Host	Löschen
FreeTextHost	6	Löschen
FreeTextService	7	Löschen
FromAddressRegExp	nagios@(elix01.eli mhkna	Löschen
HostRegExp	\s*Host:\s+(\S+)\s*	Löschen
Module	Kernel::System::PostMast	Löschen
NewTicketRegExp	CRITICAL DOWN WARNI	Löschen
SenderType	system	Löschen
ServiceRegExp	\s*Service:\s+(.*)\s*	Löschen
StateRegExp	\s*State:\s+(\S+)	Löschen
Neu		

OTRS::S.M. Logik



OTRS::PostMaster

- Filter für Queue-Zuordnung:

Bearbeiten:

Filtername:

Treffer:

Kopf 1: Wert:

Kopf 2: Wert:

Kopf 3: Wert:

Kopf 4: Wert:

Setzen:

Kopf 1: Wert:

Kopf 2: Wert:

Kopf 3: Wert:

Kopf 4: Wert:

Übermitteln



Live Demo