Solaris Services

Sebastian Pasewaldt,
Martin Koeleman,
Robert Reicherdt

Agenda

- **Allgemeine Einführung**
  - Bootvorgang bis Solaris 9
  - Service Management Facility (SMF)
- Komponenten der SMF
  - Konfigurationsdatenbank
  - *Restarter*
  - *Contracts*
- Beispiel
  - Einbindung eines Dienstes in die SMF

Bootvorgang bis Solaris 9

- beim boot wird `/sbin/init` ausgeführt
- `/etc/inittab` enthält Einträge der Form: `id:runlevel:action:process`
- actions sind: `sysinit, powerfail, wait, respawn`
- Enthält das *default runlevel*
  - Action: `initdefault`
- Init führt jeden Eintrag aus, der im runlevel-Feld das zu startende runlevel enthält

/etc/inittab – Solaris 9

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
ap::sysinit:/sbin/soconfig -f /etc/sock2path
fs::sysinit:/sbin/rcS sysinit >/dev/msglog 2<>/dev/msglog </dev/console
is:3:initdefault:
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/msglog
sS:s:wait:/sbin/rcS      >/dev/msglog 2<>/dev/msglog </dev/console
s0:0:wait:/sbin/rc0      >/dev/msglog 2<>/dev/msglog </dev/console
s1:1:respawn:/sbin/rc1   >/dev/msglog 2<>/dev/msglog </dev/console
s2:23:wait:/sbin/rc2     >/dev/msglog 2<>/dev/msglog </dev/console
s3:3:wait:/sbin/rc3      >/dev/msglog 2<>/dev/msglog </dev/console
s5:5:wait:/sbin/rc5      >/dev/msglog 2<>/dev/msglog </dev/console
s6:6:wait:/sbin/rc6      >/dev/msglog 2<>/dev/msglog </dev/console
fw:0:wait:/sbin/uadmin 2 0   >/dev/msglog 2<>/dev/msglog </dev/console
of:5:wait:/sbin/uadmin 2 6   >/dev/msglog 2<>/dev/msglog </dev/console
rb:6:wait:/sbin/uadmin 2 1   >/dev/msglog 2<>/dev/msglog </dev/console
sc:234:respawn:/usr/lib/saf/sac -t 300
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "´uname -n´ console login: " \
   -T sun -d /dev/console -l console -m ldterm,ttcompat
```

/sbin/rc* - Run Control Scripts

- Init führt `/sbin/rc`***n*** für jeweiliges runlevel aus
- `/sbin/rc`***n*** führt alle Skripte in `/etc/rc`***n***`.d` aus
- Namensgebung in `/etc/rc`***n***`.d` nach dem Schema `[SK]`*m*`<servicename>` (0<*m*<100)
- `K` → „kill“; `S` → „start“
- Ausführungsreihenfolge ergibt sich aus *m*, dadurch Beachtung der Abhängigkeiten

/etc/rc2.d/

```
$ ls /etc/rc2.d
K03samba          S72autoinstall          S99dtlogin
K05appserv        S73cachefs.daemon
K05volmgt         S74capture_uptime
K06mipagent       S81dodatadm.udaplt
K07dmi            S89bdconfig
K07snmpdx         S89PRESERVE
K15cswopenldap    S90webconsole
K15imq            S90wbem
K16apache         S91afbinit
K27boot.server    S91gfbinit
README            S91ifbinit
S10lu             S91jfbinit
S20sysetup        S91zuluinit
S40llc2           S94ncalogd
S42ncakmod        S95cst
S47pppd           S95IIim
S70sckm           S98deallocate
S70uucp           S99audit
```

Ab Solaris 10: Service Management Facility (SMF)

- Erweiterung der UNIX *start-up scripts*, *init runlevels* und Konfigurationsmöglichkeiten
- Services werden als Objekte behandelt
- Verwaltung der Service-Objekte durch Anzeige- (`svcs`) und Konfigurationstools (`svccfg`, `svcadm`)
- Konkrete Angaben zum Grund eines Service-Ausfalls erleichtern Fehlersuche
- Automatische Sicherungskopien (*Snapshots)* erlauben *rollback* zu einer funktionierenden Konfiguration
- *dependency statements* beschreiben Abhängigkeiten
- Paralleler Start von unabhängigen Services beschleunigt Bootvorgang

Services vs. Serviceinstanzen

- Grundlegende Administrationseinheit ist die service instance
- Instanzen sind spezielle Konfigurationen eines Services
- Instanzen sind Kind-Objekte des jeweiligen Service-Objekts
- Fault Management Resource Identifier (FMRI)
  (svc|lrc):/<category>/<service_name>:<instance>
  Beispiel: `svc:/network/login:rlogin`
- Categories: device, application, network, milestone, system, platform
- Milestone – Service, der durch eine Menge von Abhängigkeiten die Verfügbarkeit bestimmter Dienste sicherstellt

Änderungen durch SMF

- `/etc/inittab` ist leerer

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
sp::sysinit:/sbin/soconfig -f /etc/sock2path
smf::sysinit:/lib/svc/bin/svc.startd    >/dev/msglog
  2<>/dev/msglog </dev/console
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog
  2<>/dev/msglog
```

- `/etc/rc*.d`-Verzeichnisse sind leerer
- Nicht-SMF-Services (*legacy services*) bleiben funktionsfähig, sind aber nur rudimentär administrierbar
- Weniger Nachrichten beim Bootvorgang
  - Umleitung der Nachrichten einzelner Dienste in Logfiles
- *„Service refuses to die“*
  - *Automatischer Neustart von Diensten*

Agenda

- Allgemeine Einführung
  - Bootvorgang bis Solaris 9
  - Service Management Facility (SMF)
- **Komponenten der SMF**
  - Konfigurationsdatenbank
  - *Restarter*
  - *Contracts*
- Beispiel
  - Einbindung eines Dienstes in die SMF

Service Configuration Repository

- Zentrale Datenbank zur Speicherung aller zur Diensteverwaltung relevanten Daten
  - gespeichert in 2 SQLite Datenbanken
  - Später: Speicherung im Netzwerk möglich
- Nur *svc.configd* erlaubt Zugriff (*single point of access*)
- Persistente & transaktionsbasierte Registrierung
  - Ermöglicht Wiederherstellung des Systems
  - Bei Fehlkonfiguration sind „undo“-Aktionen möglich
- Im/Exportierung als XML-File möglich (SMF-Manifest)
  - plattformunabhängige Konfiguration
  - Transportabilität

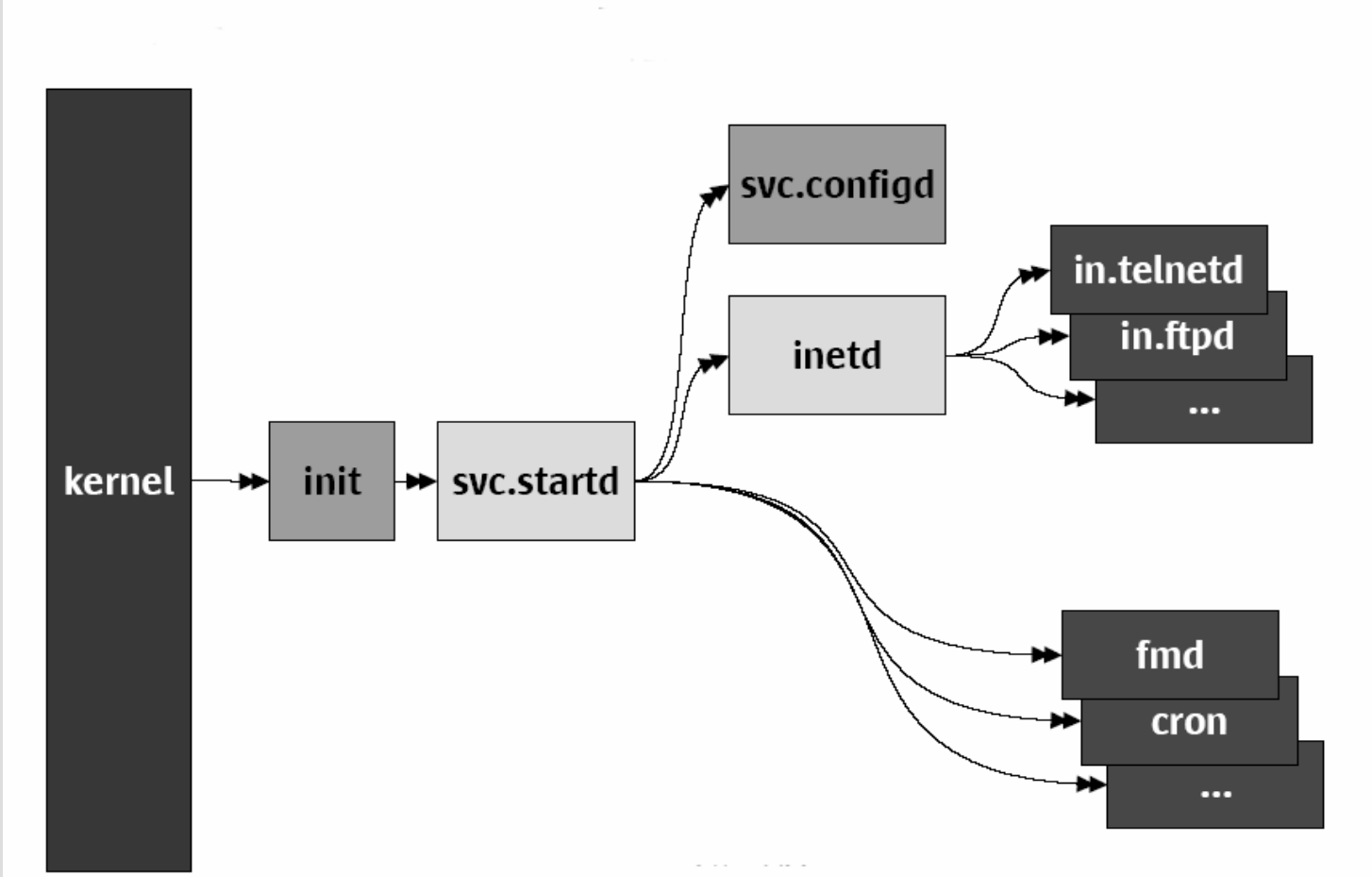
svc.startd

- Dienst zum Starten und Beenden von Services
  - *Master Restarter Daemon*
- Zuständig für runlevel Management
- Überwacht Zustand der Dienste
  - z.B. *online, legacy_run, maintenance*
- Regelt Abhängigkeiten von Services
- Startet bei Bedarf *Delegated Restarter*
- Schreibt *contracts*, reagiert auf *contract events*

Delegated Restarter

- Unterschiede zum Master-Restarter:
  - Nur zuständig für bestimmte Services
  - Feinere Granularität
  - Fehler werden innerhalb des Restarters behoben
  - Kann auch nicht-prozessbasierte Services verwalten
  - Beispiel: inetd
    - Zuständig für Internetservices

Restart Relationship



Quelle: http://mediacast.sun.com/share/lianep/t-smf-general-march-2005.pdf

Fehlerisolierung & Contracts

- Was passiert wenn ein Fehler im Speicher auftritt ?
- Bei Solaris 9:
  - Hardwarefehler wird von Kernel registriert:
    - Wenn behebbar → Userprozesse registrieren Fehler nicht & Kernel behebt Fehler
    - Wenn nicht behebbar → Kernel beendet den Prozess in dem der Fehler auftrat
  - Problem:
    - Solaris 9 weiß nichts über die Abhängigkeiten zwischen Prozessen
    - Daher: Reboot des gesamten Systems nach einem Hardwarefehler
      - Vermeidung weiterer Fehler
      - Erreichung eines definierten Systemzustands
  - Lösung:
    - Contracts !

Fehlerisolierung & Contracts

- Solaris 10 führt Contracts Konzept ein
- Contracts:
  - „Container“ zur Prozessausführung
  - Ereignisse, die die Prozesse in einem *Contract* betreffen, werden an den Ersteller des *Contracts* weitergeleitet
  - SMF Restarter können auf diese Ereignisse reagieren
- Vorteile:
  - Dienst-Abhängigkeiten + Contracts → gezielter Neustart von den Diensten, die von einem Fehler betroffen sind

Fehlerisolierung & Contracts

- SMF Restarter schreiben *Contracts* beim Starten von Prozessen

```
$ ptree -c `pgrep sendmail`
     [process contract 1]
         1 /sbin/init
           [process contract 4]
               7 /lib/svc/bin/svc.startd
                  [process contract 513]
                     18676 /usr/lib/sendmail -Ac -q15m
                     18678 /usr/lib/sendmail -bd -q15m
```

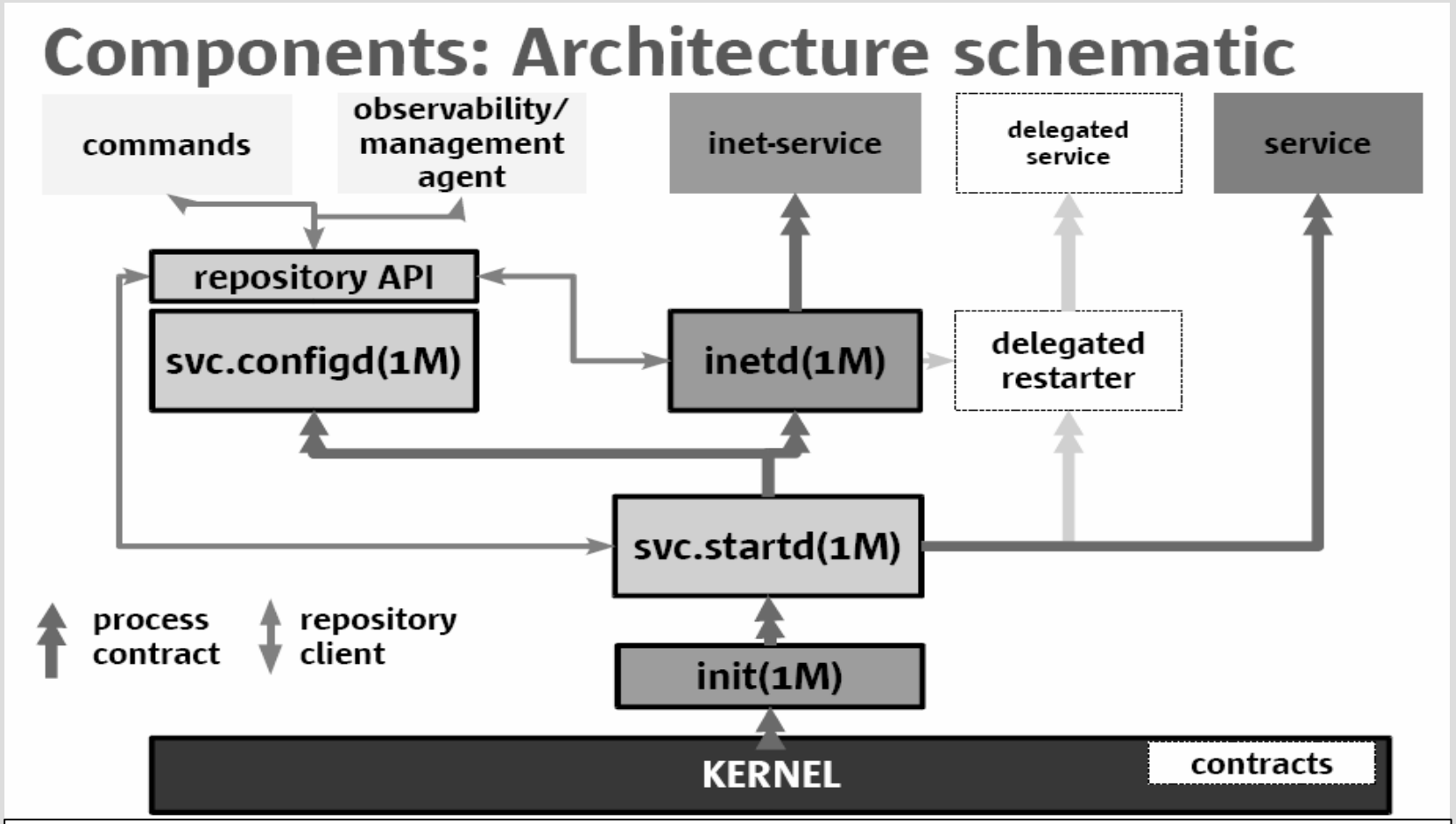
Fehlerisolierung & Contracts

```
$ ctstat -vi 513
CTID  ZONEID TYPE    STATE HOLDER EVENTS QTIME NTIME
513   0      process owned 7      0      -     -

      informative event set: none
      critical event set: hwerr empty
      fatal event set: none
      member processes: 18676 18678
```

- event sets geben Bedingungen an, wann der *Contract Holder* über einen Fehler benachrichtigt wird
- critical event set ist für SMF relevant

SMF Architektur



Quelle: http://mediacast.sun.com/share/lianep/t-smf-general-march-2005.pdf

Agenda

- Allgemeine Einführung
  - Bootvorgang bis Solaris 9
  - Service Management Facility (SMF)
- Komponenten der SMF
  - Konfigurationsdatenbank
  - *Restarter*
  - *Contracts*
- **Beispiel**
  - Einbindung eines Dienstes in die SMF

Übersicht: Konvertierung von *legacy services*

- Anlegen von Manifest-Datei in `/var/svc/manifest/<category>/<servicename>.xml`
  - Definition von Abhängigkeiten etc.
- Anlegen von Methoden-Datei in `/lib/svc/method/<servicename>`
  - Kopieren und Einbinden von `/lib/svc/share/smf_include.sh` in das Start-Skript aus `/etc/rc`***n***`.d` reicht meist
- Importieren des Manifests per `svccfg –v import /var/svc/manifest/<category>/<servicename>.xml`
- Überprüfen des Service-Status per `svcs -vx <servicename>`
- Bei Fehlern:
  - `svcadm -v disable <FMRI>`
  - `svccfg -v delete <FMRI>`
  - Erneuter Import des Manifests

Quellen

- Solaris System Administration Guide: Basic Administration Chapter 9, 14
  http://docs.sun.com/app/docs/doc/817-1985
- Liane Praza (Solaris Kernel Developer): Solaris 10 smf(5): Service Management Facility
  http://mediacast.sun.com/share/lianep/t-smf-general-march-2005.pdf
- Liane Praza's Weblog
  http://blogs.sun.com/roller/page/lianep/
- Peter's Solaris Zone: Setting up Postfix under Greenline (SMF)
  http://www.rfcgr.mrc.ac.uk/~ptribble/Solaris/smf.html
- Solaris 9 System Startup and Shutdown
  http://www.quepublishing.com/articles/article.asp?p=101659