

# open



USE



IMPROVE



EVANGELIZE

## Application Integration with OpenSolaris SMF

Sonny Hastomo

IT Architect

Sun Microsystems Indonesia

開  
放  
的  
열린  
مفتوح  
libre  
मुक्त  
ಮುಕ್ತ  
livre  
libero  
ముక్త  
开放的  
açık  
open  
nyílt  
:::  
πικρ  
オープン  
livre  
ανοικτό  
offen  
otevřený  
öppen  
открытый  
வெளிப்படை



# Agenda

- **CoolStack and OpenSolaris**
  - What is CoolStack ?
  - CoolStack Installation
  - Deploy SugarCRM in CoolStack
- **Developing SMF Service**
  - General Concept
  - Writing Service Manifest



## What is CoolStack ?

**Collection of software stack that  
optimized for Sun Solaris  
Operating System**



## CoolStack Packages

- **CSKruntime**
- **CSKamp**
- **CSKlibsbundle**
  - CSKlibsphp
  - CSKtds
  - CSKncurses
- **CSKmysql**
- **CSKperl**
- **CSKruby**
- **CSKmemcached**
- **CSKsquid**
- **CSKtomcat**
- **CSKlighthttpd**
- **CSKpython**
- **CSKngnix**



## Advantages of CoolStack

- Best level performance of your system in Solaris Operating System
- Reducing Time-to-Service



# Step-by-step CoolStack in OpenSolaris

- Installing CSKruntime as a core component of CoolStack Software

```
pkgadd -d CSKruntime_1.3.1_i386.pkg
```

- Installing CSKamp (Apache, PHP, and MySQL 32-bit)

```
pkgadd -d . CSKamp_1.3.1_i386.pkg
```

- Installing CSKlibsbundle (CSKphplibs, CSKtds, and CSKncurses)

```
pkgadd -d . CSKlibsbundle_1.3.1.pkg
```



# Configure MySQL and Enable Service

- Make sure mysql user exist
- Install MySQL database

```
/opt/coolstack/mysql_install_db
```

- Change MySQL data directory owner for mysql user

```
chown -R mysql /opt/coolstack/data
```

- *Enable CoolStack MySQL service*

```
svcadm enable mysql:mysql32-csk
```



## Enable Apache Service

- Query CoolStack Apache Service

```
svcs -a | grep apache
```

- Enable CoolStack Apache Service

```
svcadm enable http:apache22-csk
```



# Deploy SugarCRM in CoolStack

- Copy SugarCRM application to *htdocs* directory under apache software stack directory
- Configure SugarCRM directory (cache, custom, data, modules) and config.php

*chown -R 777 cache (repeat command for each directory)*

- Configure PHP to Enable Multi-byte

*Uncomment extension-mbstring.so inside file php.ini*

- Configure PHP session.save\_path inside php.ini

*session.save\_path = "/tmp"*

# SugarCRM installation preview

Welcome to  
**SUGARCOMMUNITY EDITION.**

Please enter your user name and password:

User Name:

Password:

[Options](#)

Server response time: 0.14 seconds.  
 © 2004-2008 SugarCRM Inc. The Program is provided AS IS, without warranty. Licensed under [GPLv3](#).  
 This program is free software; you can redistribute it and/or modify it under the terms of the  
[GNU General Public License version 3](#) as published by the Free Software Foundation including the additional permission set forth in the source code header.





## SMF Term and Definition

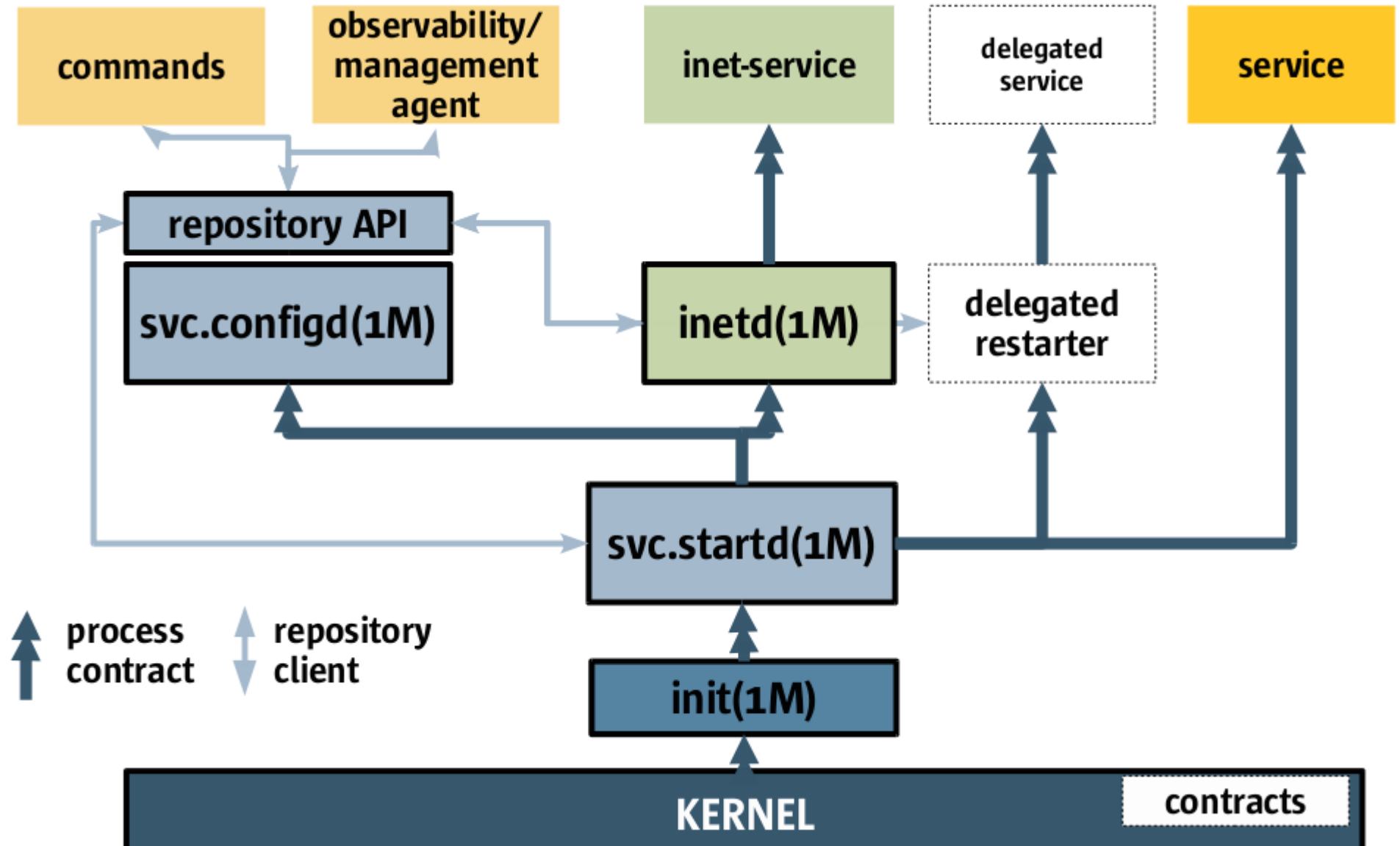
- A long lived software object with a well-defined state, error boundary, definition of start and stop, and relationship to other services. A service is often critical to operation of system or fulfillment of business objectives.
- Service Management Includes
  - Service Delivery
  - Administrative Interaction
  - Management of service by system



## Developing Services

- Unify Service delivery
- Single management for management, configuration and operation
- Enable service based resource management
- Simplify debugging of boot process

# Solaris Service Architecture





# Manifest Creation

- ***Name Your Service***
- ***Identify your service may have multiple instances***
- ***Identify your service model: contract, transient or wait***
- ***Identify how your service started/stopped***
- ***Determine fault to be ignore***
- ***Identify dependencies***
- ***Identify dependents***
- ***Insert your service into milestone***
- ***Create, if appropriate default instance***
- ***Create template information of your service***



# Service Relationship

- Restart relationships defined by dependency groups
  - ♦ grouping
    - require\_all: all services are running
    - require\_any: at least one service running
    - optional\_all: all services running, disabled, or maint
    - exclude\_all: all services disabled, maint, or absent
  - ♦ restart\_on
    - none: required only for startup
    - error: stop if dependency fails due to h/w or s/w error
    - restart: stop if dependency restarts for any reason
    - refresh: stop if dependency restarts or is refreshed



# Manifest Example

```
<service name='system/utmp' type='service' version='1'>
  <create_default_instance enabled='true' />
  <single_instance />
    <dependency name='milestone' grouping='require_all'
      restart_on='none' type='service'>
      <service_fmri value='svc:/milestone/sysconfig' />
    </dependency>
    <dependent name='utmpd_multi-user' grouping='optional_all'
      restart_on='none'>
      <service_fmri value='svc:/milestone/multi-user' />
    </dependent>

    <exec_method type='method' name='start'
      exec='/lib/svc/method/svc-utmpd' timeout='60' />
    <exec_method type='method' name='stop'
      exec=':kill' timeout='60' />
    <stability value='Unstable' />
    <template>
      <common_name><loctext xml:lang='C'>
        utmpx monitoring
      </loctext></common_name>

      <documentation>
      <manpage title='utmpd' section='1M'
        manpath='/usr/share/man' />
      </documentation>
    </template>
  </service>
```



# Manifest Example



# Creating Service Manifest - 1

- **Create service manifest for Glassfish**

*/var/svc/manifest/glassfish/glassfish.xml*

- **Edit service manifest file**

```
<?xml version='1.0'?>
```

```
<!DOCTYPE service_bundle SYSTEM
```

```
'/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
```

- **Define service bundle name**

```
<service_bundle type='manifest' name='Glassfish:glassfish'>
```

- **Define single instance service**

```
<service
```

```
  name='application/glassfish'
```

```
  type='service'
```

```
  version='1'>
```

```
  <create_default_instance enabled='false' />
```

```
  <single_instance />
```



## Creating Service Manifest - 2

- **Define dependency services**

```
<dependency name='fs'  
  grouping='require_all'  
  restart_on='none'  
  type='service'>  
  <service_fmri  
    value='svc:/system/filesystem/local' />  
</dependency>
```

```
<dependency name='net'  
  grouping='require_all'  
  restart_on='none'  
  type='service'>  
  <service_fmri value='svc:/network/loopback' />  
</dependency>
```



# Creating Service Manifest – 3

- **Define start execution method**

```
<exec_method
  type='method'
  name='start'
  exec='/lib/svc/method/svc-glassfish start'
  timeout_seconds='-1'>
  <method_context>
    <method_credential user='root' group='root' />
  </method_context>
</exec_method>
```

- **Define stop execution method**

```
<exec_method type='method' name='stop'
  exec='/lib/svc/method/svc-glassfish stop' timeout_seconds='-1'>
  <method_context>
    <method_credential user='root' group='root' />
  </method_context>
</exec_method>
```



# Creating Service Manifest – 4

- **Define Restart execution method**

```
<exec_method type='method' name='stop'  
  exec='/lib/svc/method/svc-glassfish restart' timeout_seconds='-1'>  
  <method_context>  
    <method_credential user='root' group='root' />  
  </method_context>  
</exec_method>
```



# Creating Service Execution Script - 1

- **Create execution script under `/lib/svc/method/svc-glassfish`**

```
#!/usr/bin/sh
```

```
. /lib/svc/share/smf_include.sh
```

- ***Define start execution script for Glassfish Application Server***

```
case $1 in
```

```
'start')
```

```
    echo "starting glassfish.."
```

```
    /opt/glassfish-v2/glassfish/bin/asadmin start-domain domain1 & >/dev/  
null 2> /dev/null
```

```
;;
```



## Creating Service Execution Script - 2

- **Define stop execution script for Glassfish Application Server**

```
'stop')
```

```
    echo "stopping glassfish.."
```

```
    /opt/glassfish-v2/glassfish/bin/asadmin stop-domain domain1 & >/dev/  
null 2> /dev/null
```

```
;;
```

- **Define restart execution script for Glassfish Application Server**

```
'restart')
```

```
    stop
```

```
    sleep 100
```

```
    start
```

```
;;
```

```
esac
```



# Import Service Manifest to the Repository

- Change script file for execution mode

```
chmod 777 svc-glassfish
```

- Use `svccfg` command to import the manifest file

```
svccfg import glassfish.xml
```

- Enable newly created service

```
svcadm enable glassfish
```

- Test the Glassfish application result

```
http://localhost:4848
```

- Experiment to disable created service

```
svcadm disable glassfish
```



# Reference

- Solaris Service Management Facility – Service Development  
[http://www.sun.com/bigadmin/content/selfheal/sdev\\_intro.jsp](http://www.sun.com/bigadmin/content/selfheal/sdev_intro.jsp)
- CoolStack web site  
<http://cooltools.sunsource.net/coolstack/>
- SugarCRM web site  
<http://www.sugarcrm.com/crm/>
- Service Management Facility How to Guide  
<http://www.sun.com/software/solaris/howtoguides/servicemgmthowto.jsp>
- Service Management Facility: Administration and Development  
<http://mediacast.sun.com/users/~lianep/media/t-smf-devday-june-2005.p>

# open



USE



IMPROVE



EVANGELIZE

## Thank you!

“Albert Einstein : it's not that I'm so smart it's just that I stay with problems longer”

Sonny Hastomo

IT Architect

sonny@sun.co.id

<http://epuzzle.wordpress.com>

“open” artwork and icons by chandan:

<http://blogs.sun.com/chandan>

開  
放  
的  
열린  
مفتوح  
libre  
मुक्त  
ಮುಕ್ತ  
livre  
libero  
ముక్త  
开放的  
açık  
open  
nyílt  
•••••  
πικρ  
オープン  
livre  
ανοικτό  
offen  
otevřený  
öppen  
открытый  
வெளிப்படை