



**ORACLE®**

## **Automatisierte Installationen mit Oracle Solaris 11**

**- Eine Einführung**

**Detlef Drewanz**

**Principal Sales Consultant, EMEA Server Presales**

**2011**  
**DOAG**  
Konferenz + Ausstellung

# Agenda

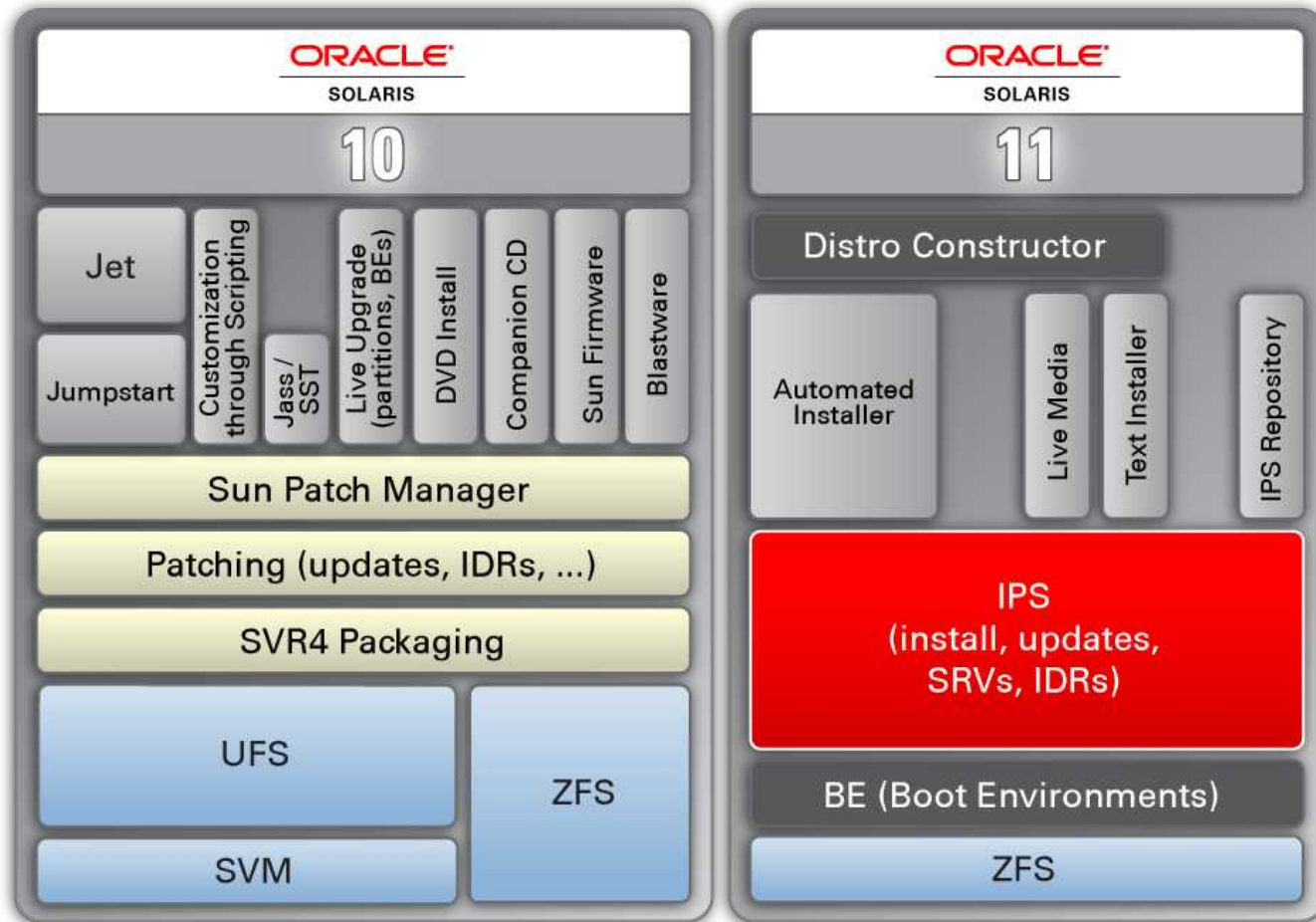
- Oracle Solaris 11 Lifecycle Management
- Das Image Packaging System (IPS)
- Automated Installer (AI)
- 



# LifeCycle Management in Oracle Solaris 11

- Installation, Konfiguration und Update vereinfacht
- Weniger planned Downtime bei Installation und Patch
- Neues Package Management (IPS)
- Überarbeitete Systemkonfiguration
- Automated Installer (AI)
- Distribution Constructor (DC)

# Installation und Konfiguration vereinfacht



# Das Image Packaging System (IPS)

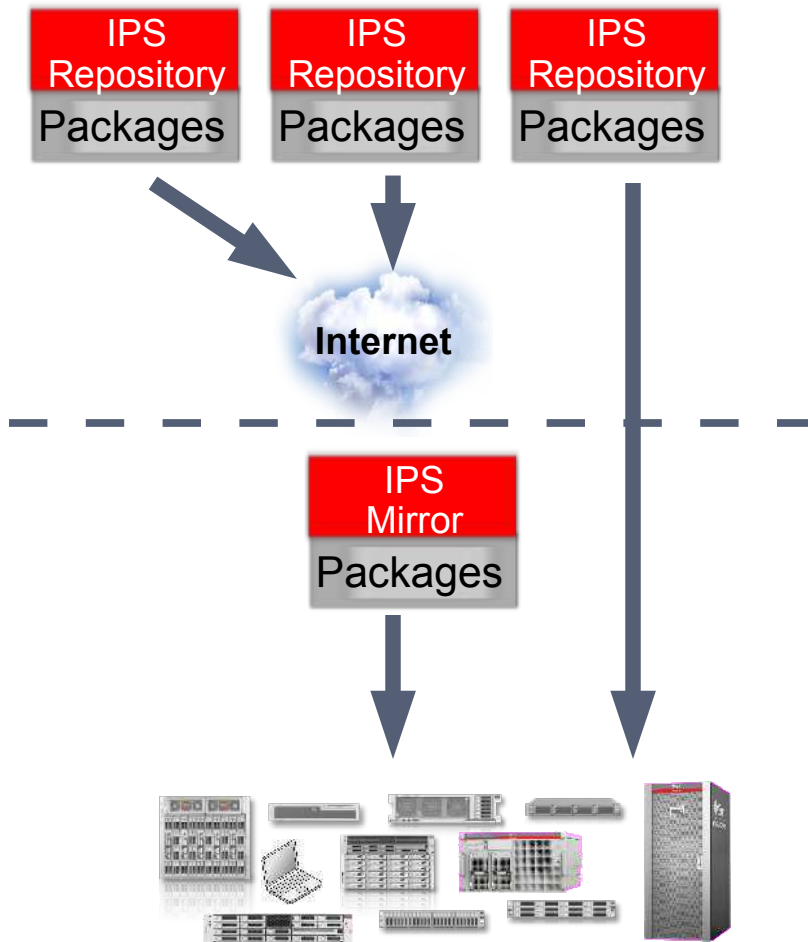
- Netzwerk-zentrisches Package Management
  - Manifeste und Package Content in Repositories
  - Lokale Repositories oder Zugriff per HTTP/HTTPS
  - Effektive Nutzung der verfügbaren Bandbreite
- Package Management
  - Installation, Update, Löschen, Status
  - Ohne Pre/Post Remove/Installation Scripting in packages
- Automatische Verfolgung von Abhängigkeiten
- Patches sind neue Versionen von Packages

# IPS Terminologie

- Package
  - Manifest und Content
  - `pkg://{publisher}/{package name}@{version}`
  - Example: `pkg://solaris/package/pkg@0.5.11,5.11-0.151:20101027T054323Z`
- Boot Environment (BE)
- Repository
  - `http://pkg.oracle.com/solaris/release`
    - Default repository für Oracle Solaris 11
  - `https://pkg.oracle.com/solaris/support`
    - Support repository
    - SRU = Support Repository Updates

# Das Image Packaging System (IPS)

## Netzwerk-Architektur

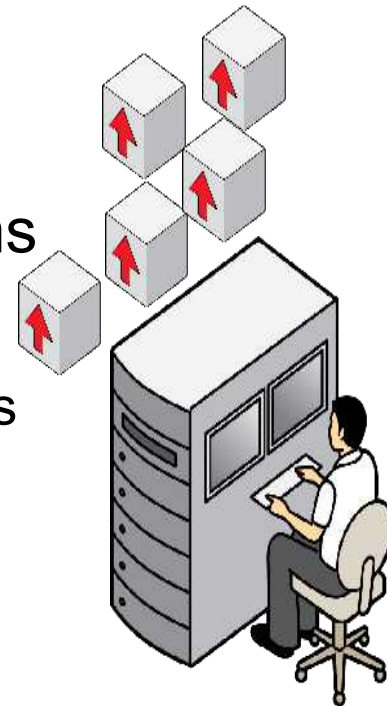


- Ein oder mehrere Repositories
- Mit lokalen Spiegel oder Kopien
- Zugriff per HTTP/HTTPS
- Installation und Upgrade
  - pkg ( 1 )
  - updatemanager
  - packagemanager ( 1 )

# Automated Installer (AI)

## Überblick

- Automatisierte Installation von Oracle Solaris 11
  - Schnelle, einfache, einheitliche, automatisierte Installation
- Einfach zu benutzen
  - Ein Kommando (installadm)
  - Erweiterbare Konfigurationsdateien
- Konfiguration der Installation und des Systems
- Unterschiedliche Quellen der Packages
  - Netzwerk (LAN, WAN), Lokal, Verteilte Repositories
- Post-Installation via first-boot-SMF-services



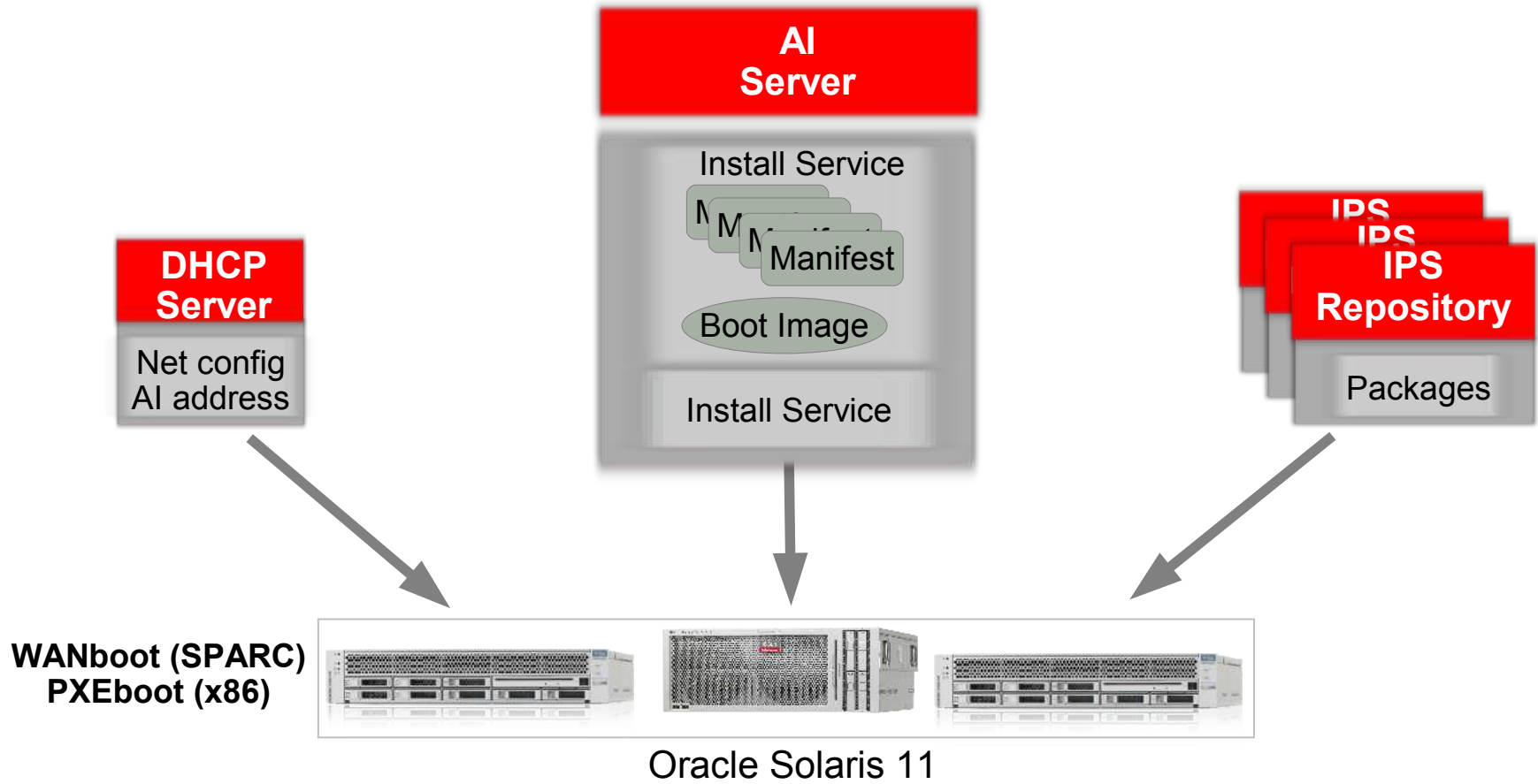


# AI Terminologie

- Client – Physische oder virtuelle zu installierende Maschine
- Manifest – XML Spezifikation der Installation
- Profile – SMF Profile zur pre-Konfiguration von Systemservices
- Service – AI Infrastruktur zum Boot eines Installations-Clients
- Criteria – Client Zuordnung zu Services, Manifesten, Profilen
- Repository – IPS package repository
- “Bootable AI” - Lokaler AI boot von DVD

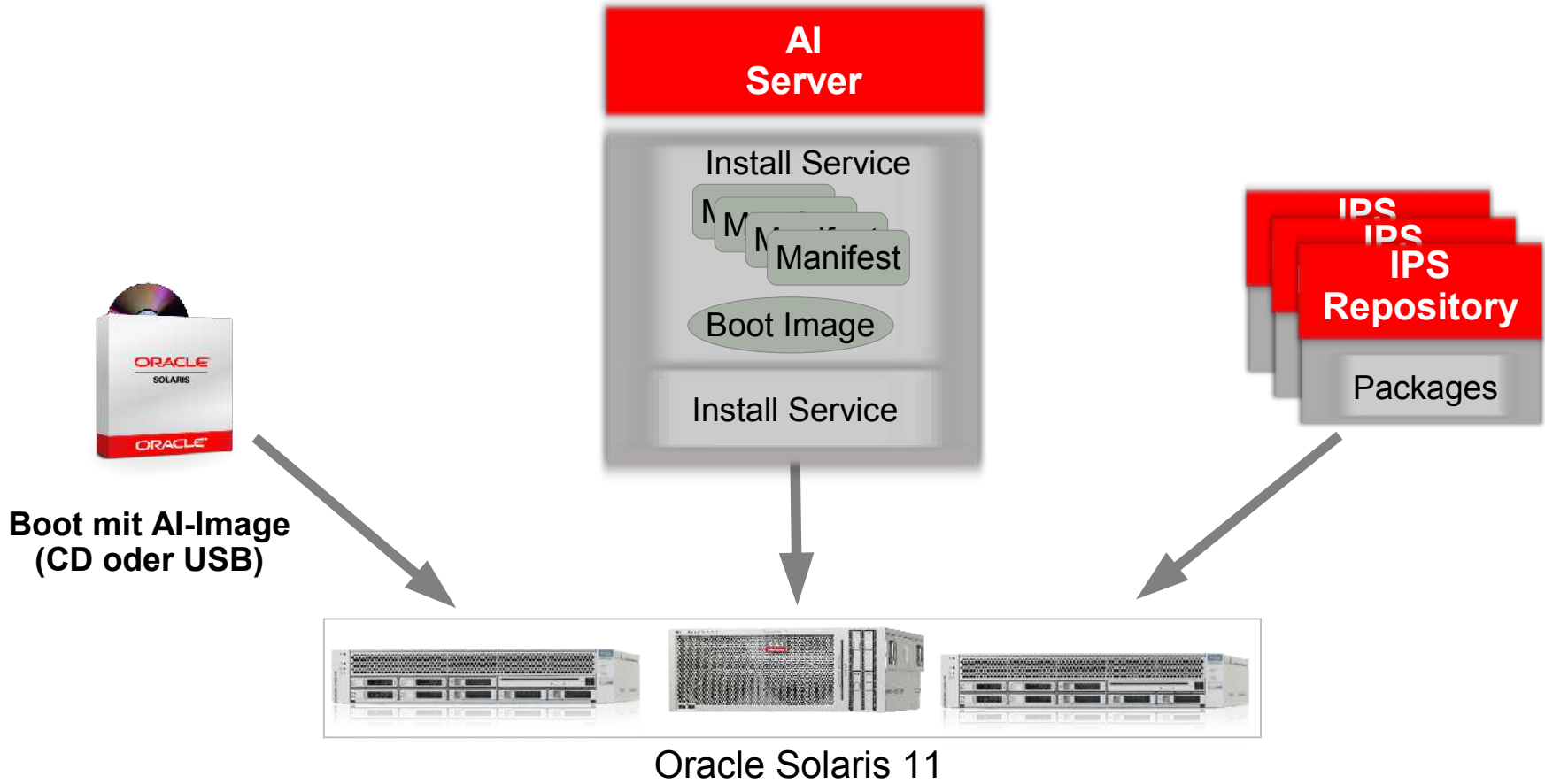
# Automated Installer

## Funktionalität



# Automated Installer

Funktionalität ohne Boot via Netzwerk



# Bestandteile AI-Server

- AI-Software
  - IPS Package installadm and dependencies
  - Services: DHCP, TFTP, HTTP/S, optional: mDNS
  - `installadm(1M)` als zentrales Kommando
- Manifeste
  - Konfiguration für Client und Installation
- IPS-Repositories
  - Quelle der Software Packages
- AI-media
  - miniroot für netboot
  - Konfigurierbar mit Distribution Constructor (DC)

# AI-Server and Manifests

- AI-Manifest
  - Define the Installation
  - Disk, Partition, IPS-Repositories, Packages
- SC-Profile
  - Define the configuration of the system
  - Timezone, root-Password, preconfigured user
- Criteria-Manifest
  - Assign AI- and SC-manifest to Client
  - Value or Area
    - ARCH/CPU/IPV4/MAC/MEM/PLATFORM

# Default AI Manifest

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd">
<auto_install>
  <ai_instance name="default">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@latest</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

# Static Manifests



- Default manifest provided with service
  - Installs solaris-large-server package set from Oracle's Solaris repository to firmware-designated boot disk
  - Sysconfig invoked automatically at first boot to interactively configure basic system
- Manifest specifies:
  - Package repositories and lists; major group packages: solaris-small-server, solaris-large-server, solaris-desktop
  - Target disk: choose by device path, volume id, type, vendor, size, container/receptacle/occupant (CRO) label; ZFS configuration
  - Locales are installed/removed using package facets; all locales are installed by default

# Derived Manifests

- Dynamically generate manifest in a script
- Scales AI management by reducing number of manifests maintained by administrators
- Most effective model is to load template manifest, modify specific elements
- Script uses the **aimanifest (1M)** command as interface to generate AI manifest
- Generated manifest located on the client at:  
    /system/volatile/manifest.xml



# System Configuration Profiles

- Common parameters available in Solaris 11:
  - User account, including RBAC roles, profiles and **sudo**
  - Root user: password, role/normal
  - Timezone, locale
  - Hostname
  - Console terminal type, keyboard layout
  - IPv4 and/or IPv6 interface, default route
  - DNS, NIS, LDAP clients
  - Name service switch

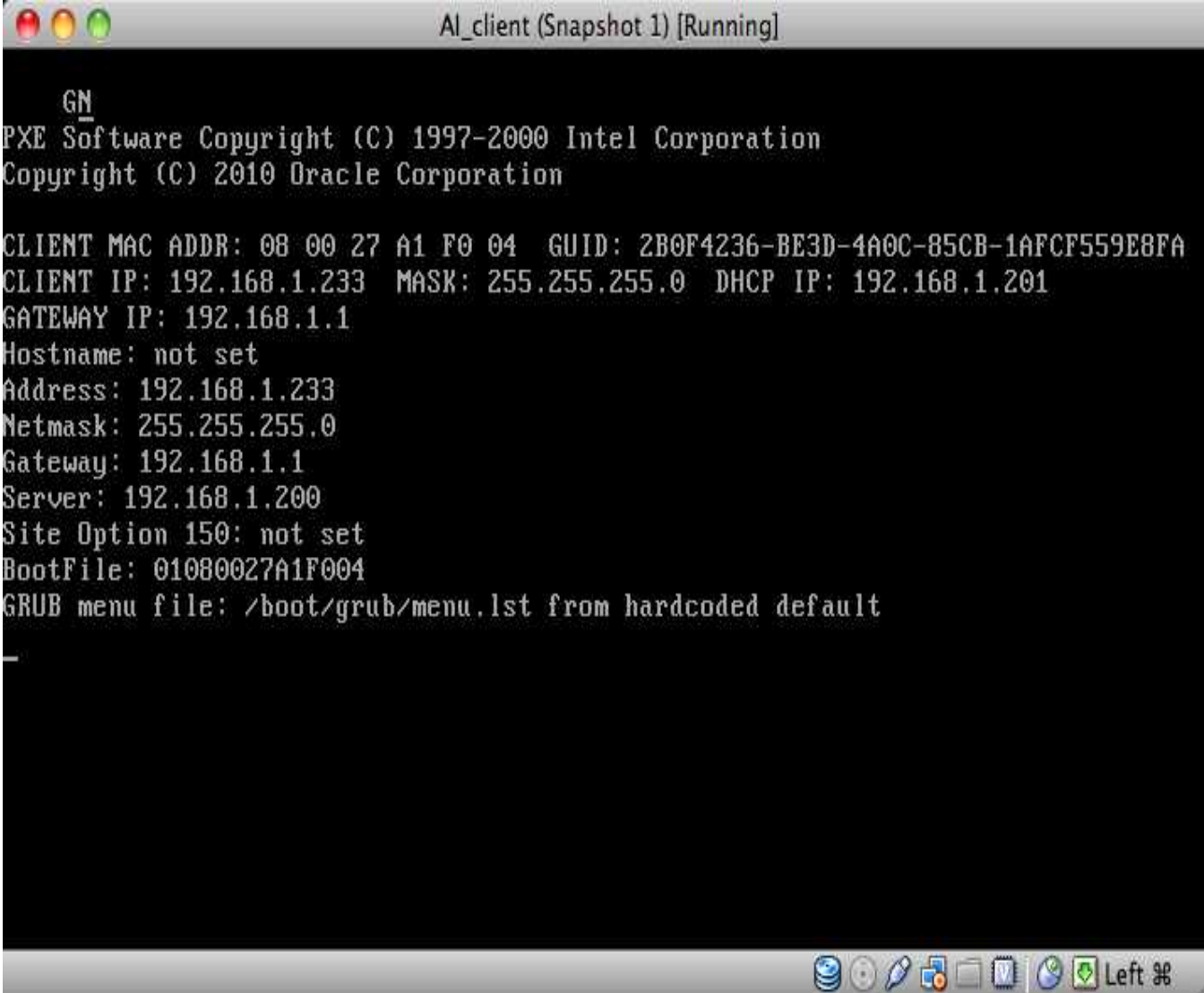
# Creating a Configuration Profile with sysconfig

- Easiest starting point

```
# sysconfig create-profile -o myprofile.xml
```

- Runs the sysconfig UI, places output into specified profile
- Edit further to add properties not configured by sysconfig

# Automated Installation: DHCP client



```
GN
PXE Software Copyright (C) 1997-2000 Intel Corporation
Copyright (C) 2010 Oracle Corporation

CLIENT MAC ADDR: 08 00 27 A1 F0 04  GUID: 2B0F4236-BE3D-4A0C-85CB-1AFCF559E8FA
CLIENT IP: 192.168.1.233  MASK: 255.255.255.0  DHCP IP: 192.168.1.201
GATEWAY IP: 192.168.1.1
Hostname: not set
Address: 192.168.1.233
Netmask: 255.255.255.0
Gateway: 192.168.1.1
Server: 192.168.1.200
Site Option 150: not set
BootFile: 01080027A1F004
GRUB menu file: /boot/grub/menu.lst from hardcoded default
-
```

# Automated Installation: Obtaining miniroot

```
AI_client [Running]
SunOS Release 5.11 Version snv_162 32-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
NOTICE: System detected 32 cpus, but only 1 cpu(s) were enabled during boot.
NOTICE: Use "boot-ncpus" parameter to enable more CPU(s). See eeprom(1M).
Hostname: ai-client4
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-04-20 02:30:49-- http://10.0.2.15:5555/export/aiserver/snv162x86/solaris
.zlib
Connecting to 10.0.2.15:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 114557440 (109M) [text/plain]
Saving to: `/tmp/solaris.zlib'

45% [=====>] 1 52,340,136 6.64M/s eta 12s
```

# Derived Manifest Example

```
#!/bin/ksh

aimanifest load /net/192.168.56.1/export/derived/target_frag.xml

if [[ $SI_NUMDISKS -gt 1 ]]; then
    aimanifest add disk/disk_name@name $SI_DISKNAME_1
    aimanifest add disk/disk_name@name $SI_DISKNAME_2
fi

aimanifest load -i /net/192.168.56.1/export/derived/software_frag.xml
aimanifest set origin@name http://ipkg.us.oracle.com/solaris11/dev
exit 0
```

# Derived Manifest Example (2)

```
# cat target_frag.xml

<!DOCTYPE auto_install SYSTEM"file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance>
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home" mountpoint="/export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
  </ai_instance>
</auto_install>
```

# Derived Manifest Example (3)

```
# cat software_frag.xml

<!DOCTYPE auto_install SYSTEM"file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://ivy2.us.oracle.com/nightly/" />
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

# AI Criteria

- Default manifest (if configured) is used when no criteria exist or match; similar to `any` rule in JumpStart

Criterion	Range	Equivalent JumpStart Rule
arch	No	karch
cpu	No	arch
ipv4	Yes	hostaddress
mac	Yes	hostname (with RARP & ethers)
mem	Yes	memsize
platform	No	model



# Deploying Zones with AI

- Zones can be specified in the AI manifest

```
<configuration type="zone" name="zone1"  
    source="http://server/zone1/config"/>  
<configuration type="zone" name="zone2"  
    source="file:///net/server/zone2/config"/>
```

- config file is the zone's configuration file as output from "zonecfg export"
- Automatically installed on first boot of the global zone  
svc:/system/zones-install:default

# Deploying Zones with AI (2)

- Use zonename criterion to associate manifests and profiles with a zone

```
# installadm create-manifest -n S11-x86 -f /tmp/zmanifest.xml  
-c zonename="zone1 zone2"
```

```
# installadm create-profile -n S11-x86 -f /tmp/zprofile1.xml  
-c zonename="zone1"
```

```
# installadm create-profile -n S11-x86 -f /tmp/zprofile2.xml  
-c zonename="zone2"
```

# Deploying Zones with AI (3)

- AI is also used when installing non-global zones from existing global zone
- Default manifest is
- Default profile enables interactive system configuration during first boot
- Provide alternate manifest and/or profile with

```
/usr/share/auto_install/manifest/zone_default.xml
```

```
# zoneadm -z <zone> install -m <manifest> -c <profile>
```

# Steps to Convert from Solaris 10 JumpStart

- Deploy S11 server instance to host AI service
  - Use as JumpStart server as well
- Translate rules to criteria
- Translate profiles to manifests
- Translate sysidcfg to SMF profile
- Publish manifests and profiles to AI service
- Convert finish scripts to SMF service(s)
- Publish SMF service package to IPS repository

# js2ai JumpStart to AI translation tool

- Automatically converts existing JumpStart rules, profiles, sysidcfg files to AI equivalents
- Conversion is best-effort, with instructions on issues that need manual resolution
- Result is a directory hierarchy with AI profiles, system config manifests, log of the tool's actions
- See **js2ai (1m)**

# JumpStart to AI Mapping

JumpStart	Automated Installation
setup_install_server	Installadm create-service
add_install_client	Installadm create-client
JumpStart profile & rules	AI manifest & criteria
sysidcfg file	SMF configuration profile
Begin script	Derived Manifests, custom images from Distribution Constructor
Finish script	pkg actions, First-boot SMF services

# Prepare to get ready for IPS and AI

- IPS is the new standard for Oracle Solaris 11
  - SVR4 packages still usable
- Begin to think in terms of Repositories for packages
  - “On disk” or “Over network”
  - Create your own repositories, use Oracle or some third party ?
- New paradigm, what does that mean ?
  - Packages without scripting
  - No patches
- Start thinking about adopting AI
  - What really needs to be “migrated” from Jumpstart to AI ?

# Summary

- Oracle Solaris 11 deployment is different from Solaris 10
  - Little required customization work to start deploying
  - Powerful, stable, supported capabilities for those who need to customize
- Transition documentation, tools are provided
- Feature set will expand & evolve
- Development of IPS and Installation are still in the open as OpenSolaris projects
  - Many opportunities for advice, feedback, discussion
  - <http://hub.opensolaris.org/bin/view/Community+Group+install/>



# Resources

- Oracle Solaris 11 Express Image Packaging System Guide
  - <http://download.oracle.com/docs/cd/E19963-01/pdf/820-6572.pdf>
- Managing Boot Environments With Oracle Solaris 11 Express
  - <http://download.oracle.com/docs/cd/E19963-01/pdf/820-6565.pdf>
- Oracle Solaris 11 Express Automated Installer Guide
  - <http://download.oracle.com/docs/cd/E19963-01/pdf/820-6566.pdf>

# Q & A



# **Hardware and Software Engineered to Work Together**

**ORACLE®**

# Rosetta Stone for Solaris 10 Users

## Software Installation Management



Solaris 10	Solaris 11 Express
SVR4 Packages	IPS Packages
Install DVD	Install CD + pkg repository
Live Upgrade	Boot Environments
Upgrade from installer	pkg(1), Update Manager
JumpStart	Automated Installer(AI)
JumpStart Profiles	AI manifests
Flash Install	No equivalent yet
Blueprints for custom DVD's	Distribution Constructor

## Overview

### Reduce Deployment Time with Training

- Expert-led training to support your adoption of Oracle Solaris 11
- Learning paths for new and experienced system administrators
- Flexible training formats to save you time and money: In-class, virtual-class or self-study



### Key Courses Available for:

- Transition to Oracle Solaris 11
- Oracle Solaris 11 System Administration
- Oracle Solaris 11 Advanced System Administration

[oracle.com/training/solaris11](http://oracle.com/training/solaris11)

### Exclusive Offer For Premier Support Customers:

- Save 20% on Training. Details at [education.oracle.com/renewaloffer](http://education.oracle.com/renewaloffer)

# Reduced Downtime, Reduced Risk

- Automated Installer and Image Packaging System make maintenance fool-proof and **safe**
  - Built-in dependency tracking prevents system modification if changes are incorrect
  - Upgrading in sand-boxed boot environments provides **easy roll-back** option
  - Ease of use through automatic dependency tracking
- Drastic Improvement of Reboot Performance
  - Streamlined shutdown
  - “Fast Reboot” **reduces time** by not testing hardware we already know to be good

6x faster Zone provisioning vs hypervisors

4x faster upgrades vs Red Hat

2.5x faster reboots vs Red Hat

When “Good Enough” Is Not Reliable Enough