appzero.com

Virtualize-2-Modernize

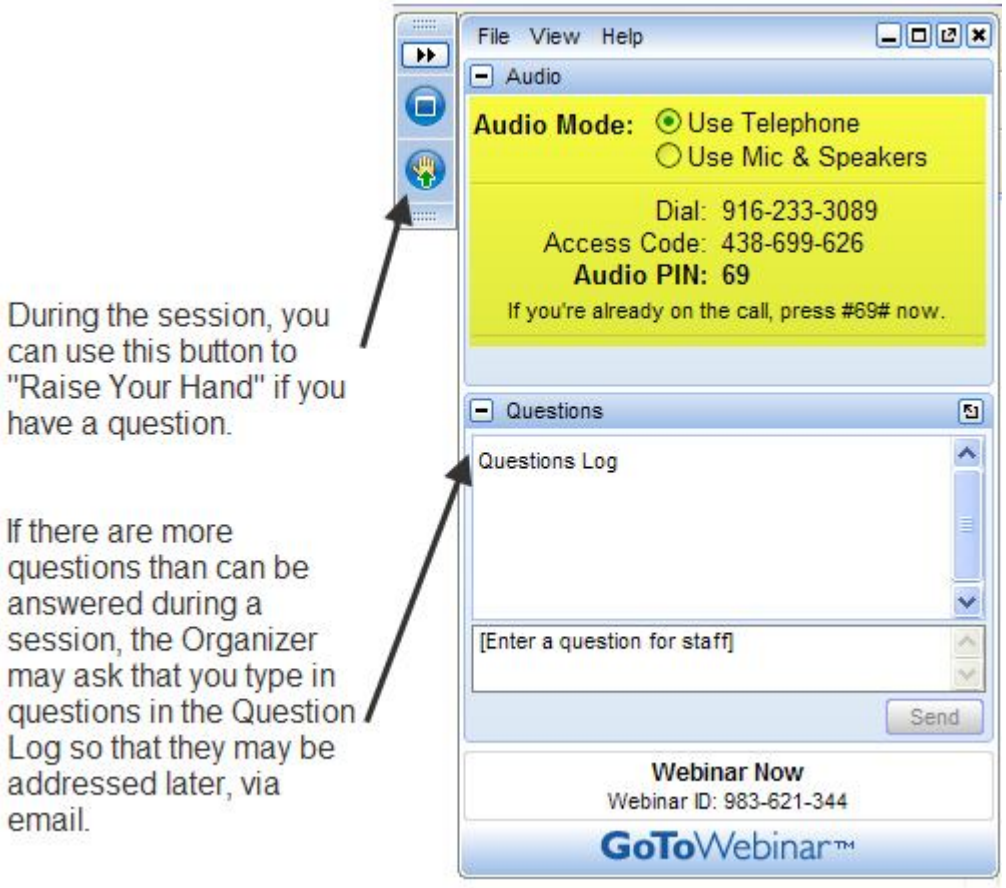
Technical drill down

Solaris 2.6 + 7 applications run unchanged on 9 + 10

appzero.com

appzero

Housekeeping



- Please Participate
- Use the GoToWebinar controls to ask questions
- We'll take most questions at the end

Virtualize-2-Modernize Introductions

- AppZero – application virtualization software vendor
  - Server applications (Solaris, Linux, Windows) and Windows desktop
  - Founded in 2006 – Solaris application virtualization first product
  - Patented and field-proven software
    - Greg O'Connor – CEO; serial innovator; co-inventor of the ESB Sonic Software
    - Bill Cullen – CTO; co-inventor of ESB Sonic Software
    - Ernesto Benedito – Lead Architect UNIX platform
    - Isaac Rozenfeld – Principal Sales Consultant at Oracle Corporation
- Virtualize-2-Modernize technical drill down – 60 minutes total
  - Review: value proposition = modernizing Solaris 2.6 and 7
  - Solution components and considerations
  - Legacy apps running on modern OS and hardware
  - AppZero's Application Assessment
  - Q&A and FAQs

What we hear about Solaris 2.6 and 7 apps

- Applications sit in the corner with a “Don’t Touch” sign
  - Undocumented, unsupported, and fragile
  - “Don’t break it because we don’t know how to fix it.”
- Not mission critical, but “pretty darn important”
  - Departmental solution
  - Important component of a solution delivery value chain
  - Not ‘worth’ re-engineering; too valuable to decommission
- Running on hardware that is no longer available
  - Ebay and Craigslist exhausted; closets empty
  - Next time is the big one
  - Management and LOB in denial; IT always makes it work
- It’s not “if”, it’s “when”

Solaris 2.6 and 7 applications

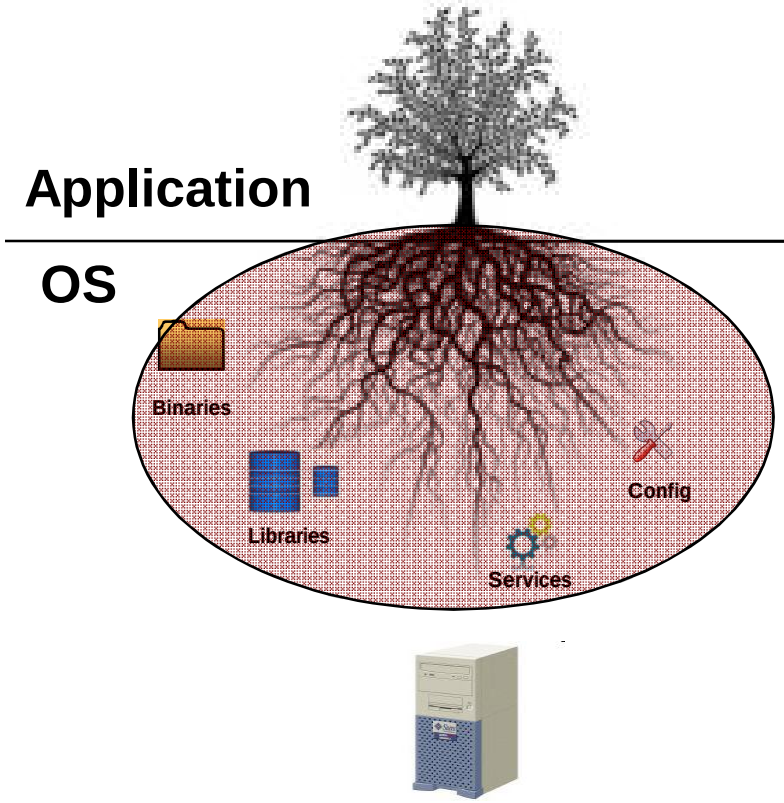
Today

- Apps run on unsupported OS that can not run on new hardware
- Old hardware with poor reliability; increasingly difficult to resuscitate
- Server sprawl
- Frozen/inflexible
- Risk increases daily

With AppZero

- Run legacy apps unchanged on Solaris 10
- Run on reliable, efficient hardware with standard maintenance
- Server consolidation ratio 15-30 to 1
- Mobility: datacenter/cloud
- Predictable

AppZero packages applications/dependencies in VAA*



=

*** Virtual Application Appliance (VAA)**

The AppZero Virtual Application Appliance VAA)

The () VAA:

- Packages all the elements required by the application to run:

  Executables, libraries, services, scripts, configuration files, network identity, host identity.

- Requires a compatible OS on the target machine
- Is isolated from the underlying OS and from other VAAs.
- Sees a configurable, merged view of its own files and those of the OS
- Makes no changes to the underlying OS
- Can be managed independently and run without conflict.
- Enables old applications to be migrated to new OS and hardware
- Complements Oracle Solaris Zones and Oracle VM Server for SPARC

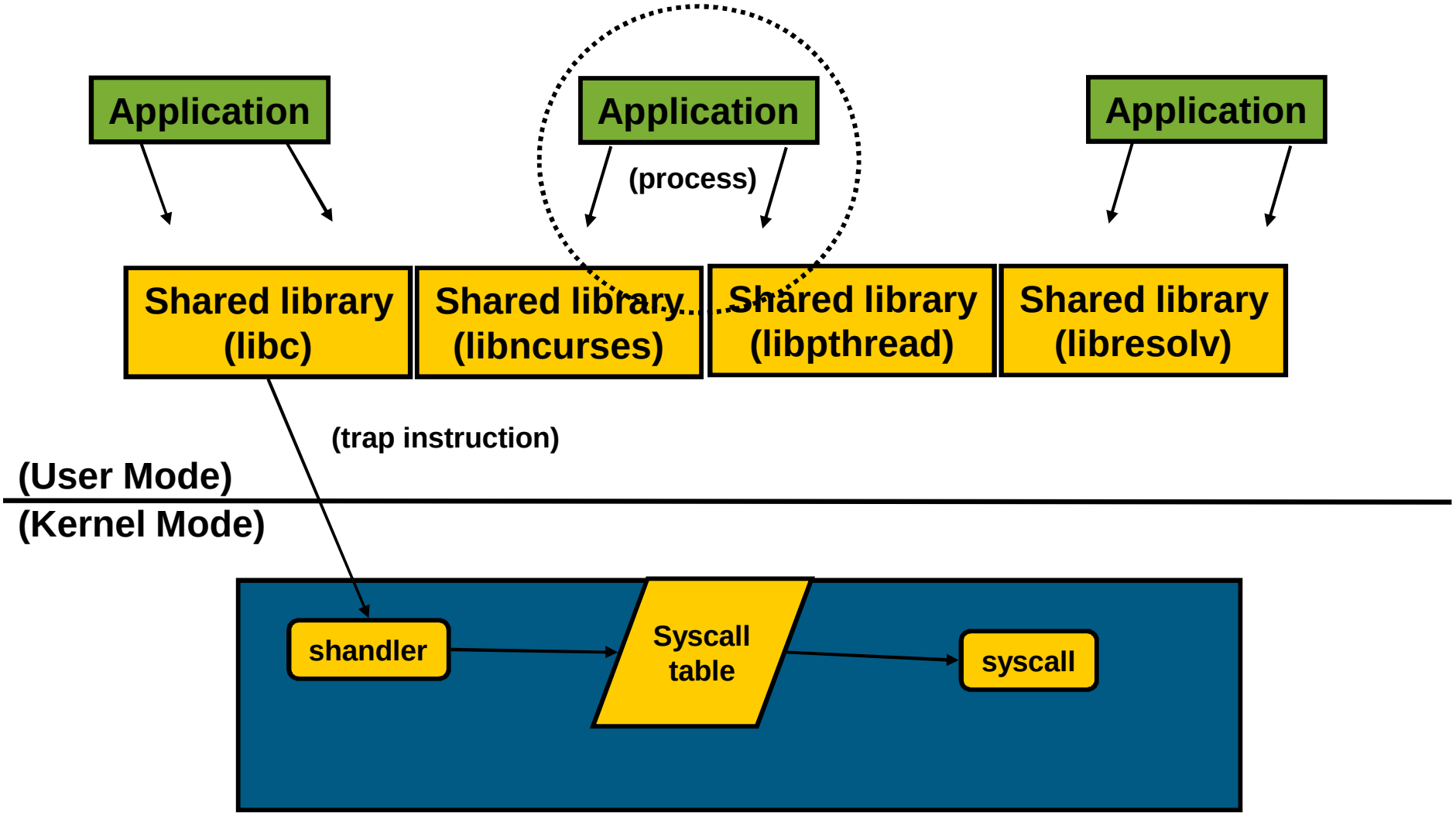
How does AppZero work under the hood?

- Appzero uses “intercepts”
  - A point of control inserted between an application and the OS
- Appzero intercepts certain system calls
  - Virtualization logic in both the kernel and userland space
- Executes when the application executes
  - Not a background process or “agent”

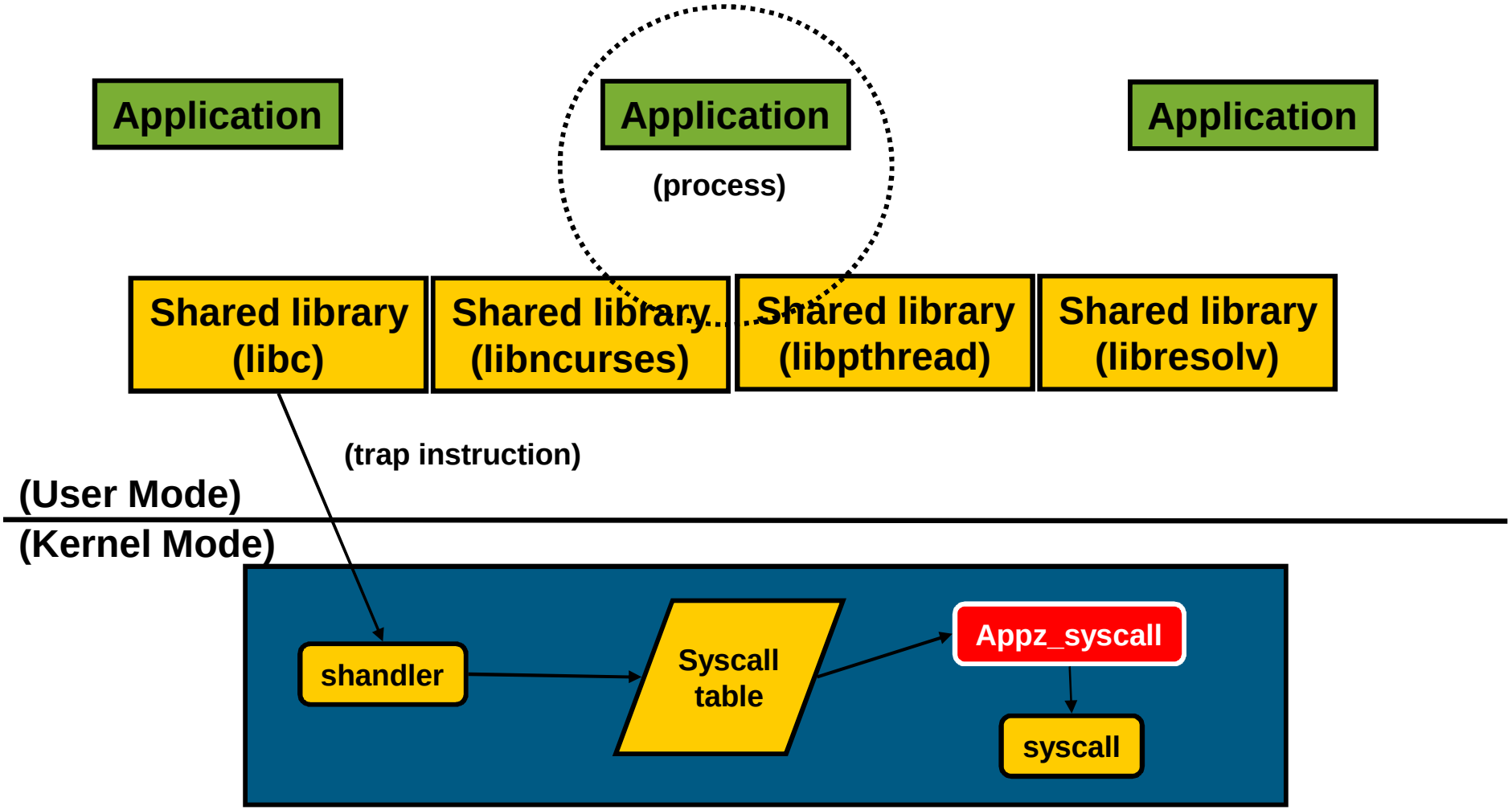
What do AppZero intercepts do?

- Process tracking
  - fork, exit, wait
  - If parent is in appliance context then child is in capsule
- Monitor
  - read, write, send, recv, sendto, recvfrom
  - Track H/W resource usage; CPU, memory, network & disk
- Parameter Spoofing
  - uname, ioctl, getmsg, bind, sendmsg,
  - CFM: open, stat, fstat, etc.
  - Modify a parameter sent to or received from the kernel

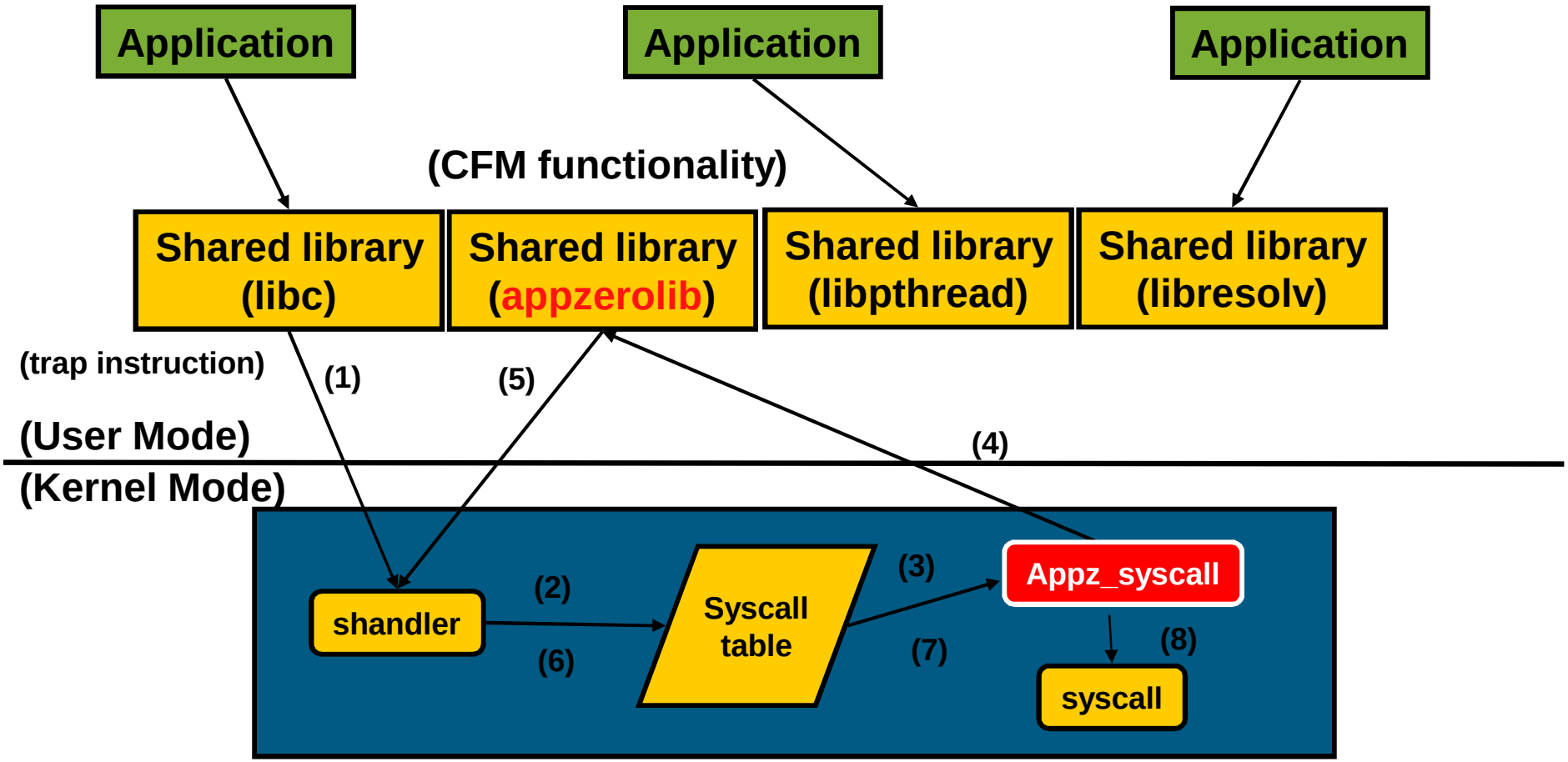
The OS model


Application
Application
(process)
Application
Shared library (libc)
Shared library (libncurses)
Shared library (libpthread)
Shared library (libresolv)
(trap instruction)
(User Mode)
(Kernel Mode)
shandler
Syscall table
syscall

AppZero Controller: Intercept



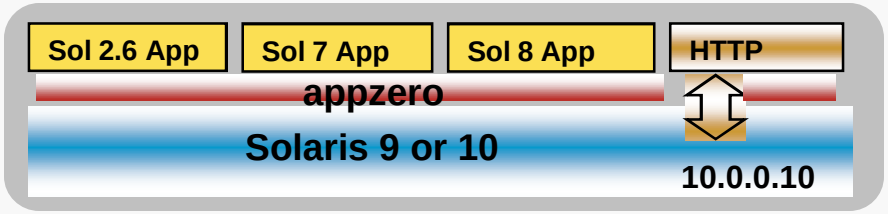
AppZero: Round trip



AppZero Solaris elements

Appzero director: orchestrates interaction of application and underlying OS

(Solaris 9, 10)



- **Appzero creator: packages application and dependencies into VAA**

(Solaris 9, 10)

- **Virtual Application Appliance (VAA) = container = capsule**

(Solaris 2.6, 7, 8, 9 and 10)

- *Note: There is an administrative console for managing VAAs*

appzero

Analyze

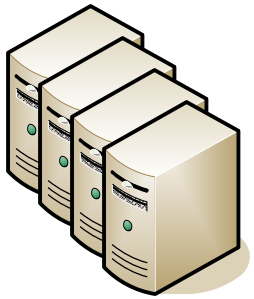
- Analyze if the application has:
  - Dependencies on kernel drivers deployed by the application.
  - Special **hardware** requirements available/installed only on the source server, such a dedicated backup ***hardware*** device.
- Source System:
  - Review file system: i.e. mount points, total disk space needed, special asynchronous vs. synchronous I/O requirements.
  - Review services: determine which services are required for the app that will end up in the appliance
  - Review kernel parameters: many applications require specific kernel parameters (e.g. #semaphores, shared memory, etc)
- Complete the application profile
- Complete recipe for validation (UAT)

Create

- A creator build machine
  - Standalone or part of the target system
- Schedule maintenance window
  - Application services on the source system are down
- Remotely capture files from the source system.
  - Options are exporting/mounting, tar or ssh
  - AppZero screate (source create) automates moving source to target
- Start AppZero creator tool
  - Learn Mode, Copy Mode, Install Mode (from VAA shell)
- Final format can be uncompressed or compressed & encrypted

AppZero Creator – creating a VAA

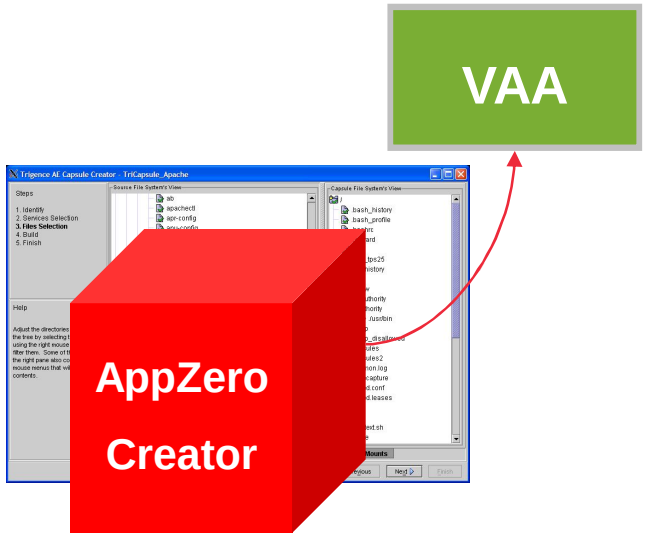
Source



Physical

Data Center

Virtualize to VAA



Approaches for creating a VAA:

1. **Learn mode**
2. **Fresh Install on empty VAA**
3. **Whole HD capture**

Output

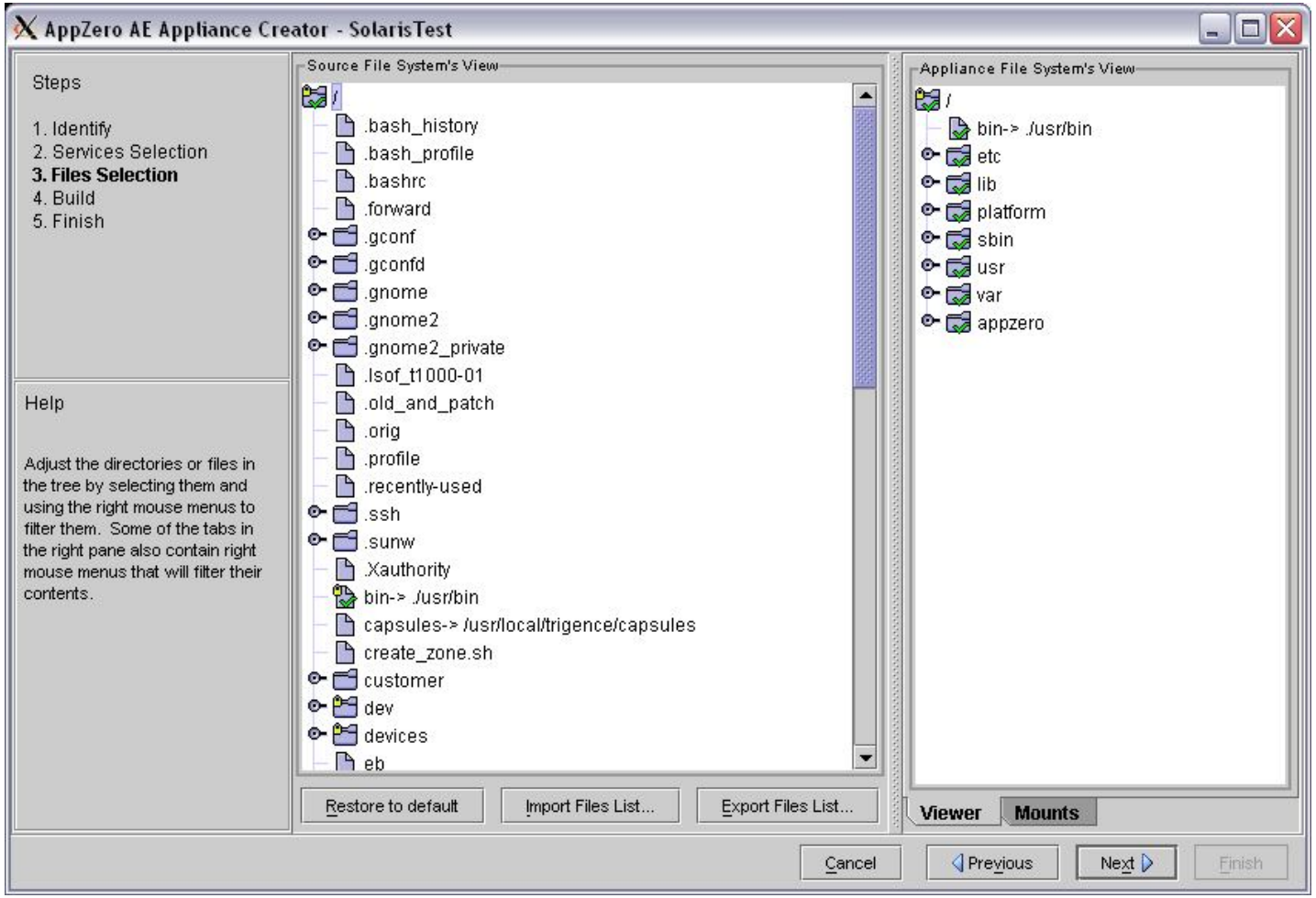
VAA

Portable

Final format can be uncompressed or compressed & encrypted file

AppZero Creator

GUI tool on Solaris 10 that allows you to create appliances



Encapsulating a legacy system

Move to create machine

- **tar.gz file**
- **Export/share/mount**
- **AppZero - screate**

Install

Target

Solaris 10

Modern Solaris 10 server

Source

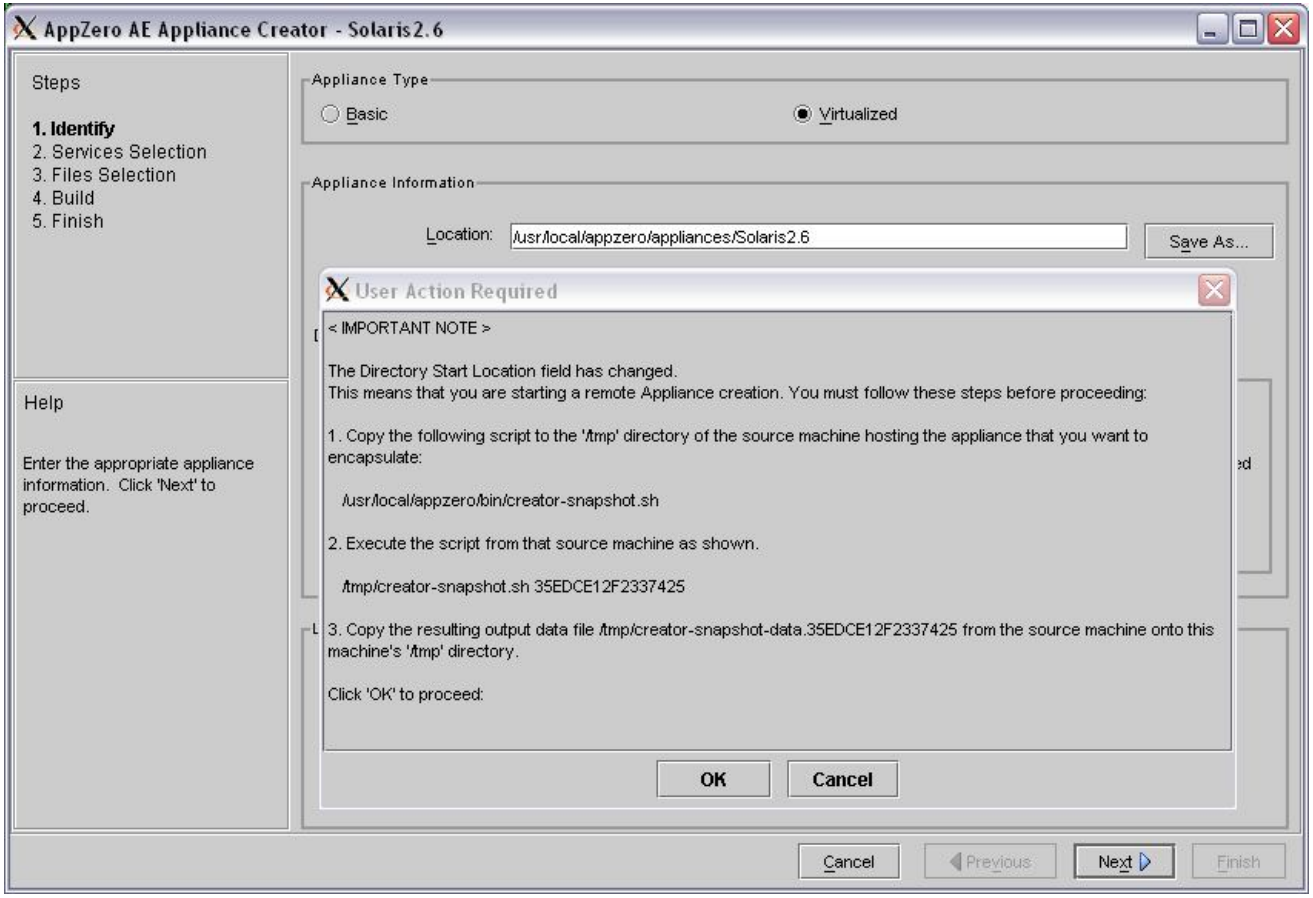
Solaris 2.6 with Application

Ultra10

Note: Creation of a VAA requires access to the files in the legacy system:

AppZero Creator: Identify files

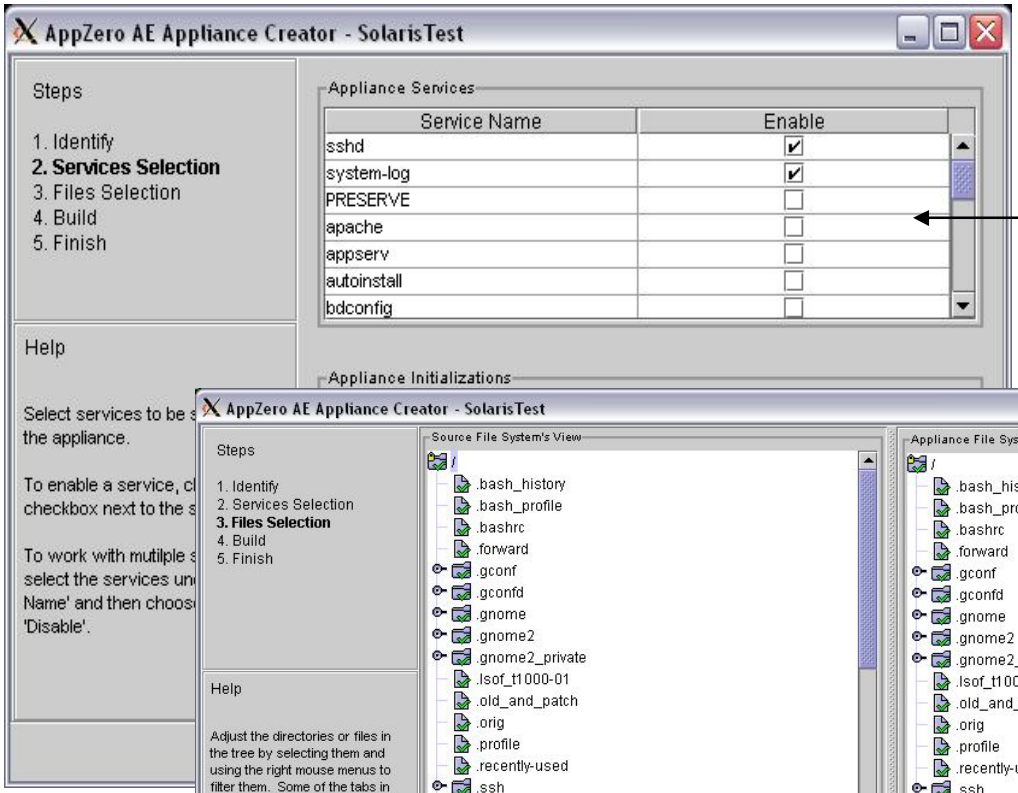
Point the creator tool to the files from the old system

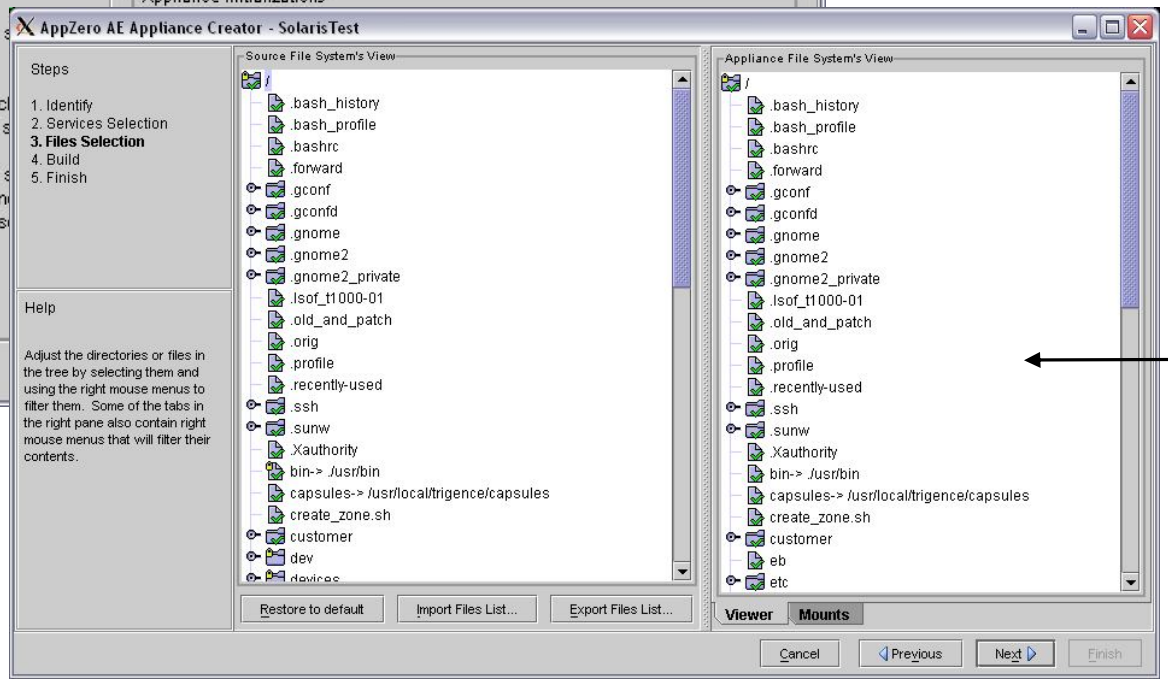


Move and run a small script to the source

appzero

AppZero Creator: Capture files, refine properties





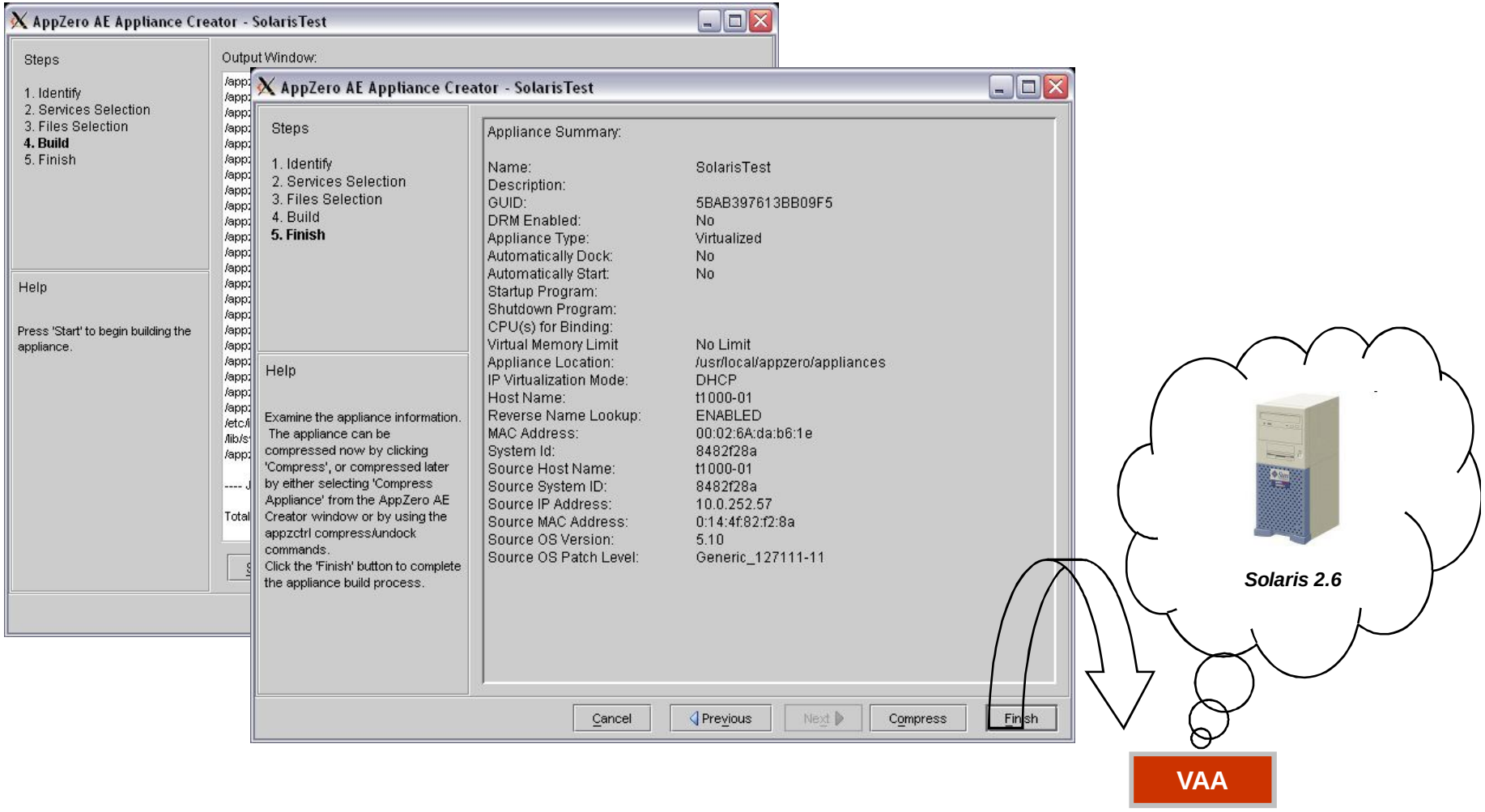
- **Select services**
- **Resource utilization**
- **Start up mode**

Review files in the VAA

Tweak mount points

appzero

AppZero Creator: Build VAA file set and/or file



VAA -Properties

```
root@tps31 /]$ appzctrl hr1texas
               Capsule Name (cn) = hr1texas
                       GUID (id) = 9B19D6C659560C69
                       Type (ty) = VIRTUALIZED
              File Revision (rv) = 26
                 Namespaces (dp) = [9B19D6C659560C69 (/),0000000000000001 (/kernel:/dev:/devices:/proc:/etc/mnttab:/opt/app/)]
           Intercept Version (iv) = 3.2.0 (build cc09w35c)
                  Creator ID (ci) = nFE+2IfqNlU0xEw0liWK6+P9vd1Ctuec
                Description (dc) = Inhouse HR system, database and app logic
                DRM Enabled (de) = N
             Creator Version (cv) = 3.0.2
         Create in Namespace (cr) = 9B19D6C659560C69
               Create Filter (cf) = /home/export/
             Whiteout Bypass (wb) =
  Executable ns r/w override (eo) = Y
          T1 lib redirection (ut) =
             Namespace order (no) = [9B19D6C659560C69,0000000000000001]
           Auto-dock On Boot (ad) = N
        Auto-startup On Dock (as) = Y
              Startup Script (ss) =
             Shutdown Script (ds) =
                 CPU Binding (cb) = [0,1]
        Virtual Memory Limit (vl) = 0
                       DHCP (dh) = STATIC
                 IP Address (ip) = 10.7.21.51
         Reverse Name Lookup (nl) = DISABLED
                   Hostname (hn) = hr1t
                MAC Address (ma) = 00:02:6A:f1:7f:0c
                  System ID (sy) = 8305648d
             System Patch ID (pl) =
             Source Hostname (sh) = hr1t
          Source MAC Address (sa) = 8:0:20:da:66:e8
            Source System ID (si) = 80da66e8
          Source IP Address (sp) = 192.168.3.175
      Source controller GUID (sg) = foBE586DF39494D450
           Source OS Version (sv) = 5.8
[root@tps31 /]$
```

Merge of file systems – what the VAA can see

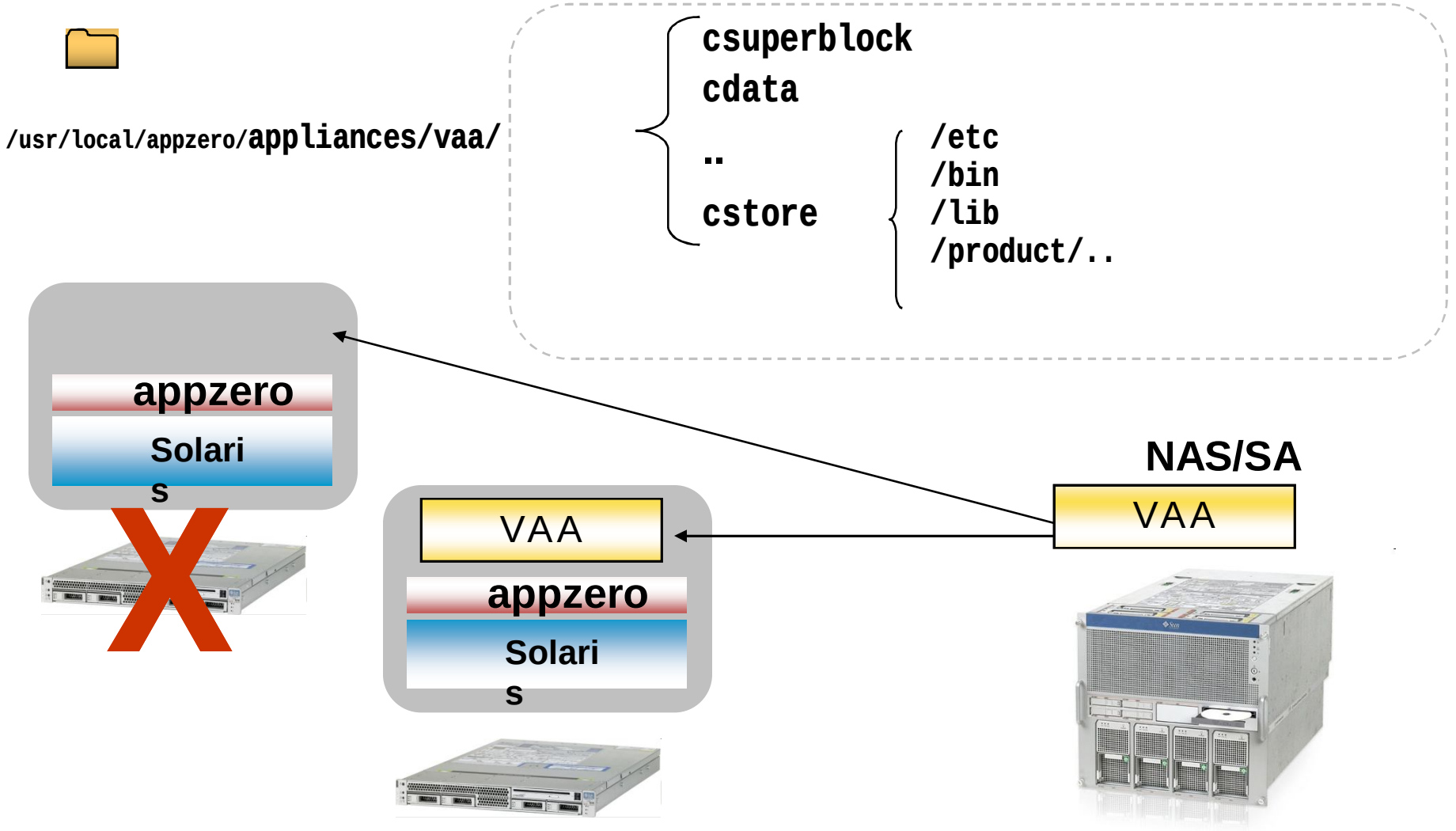
Where files will be created by default

Search order for files

Resource management

Network identity

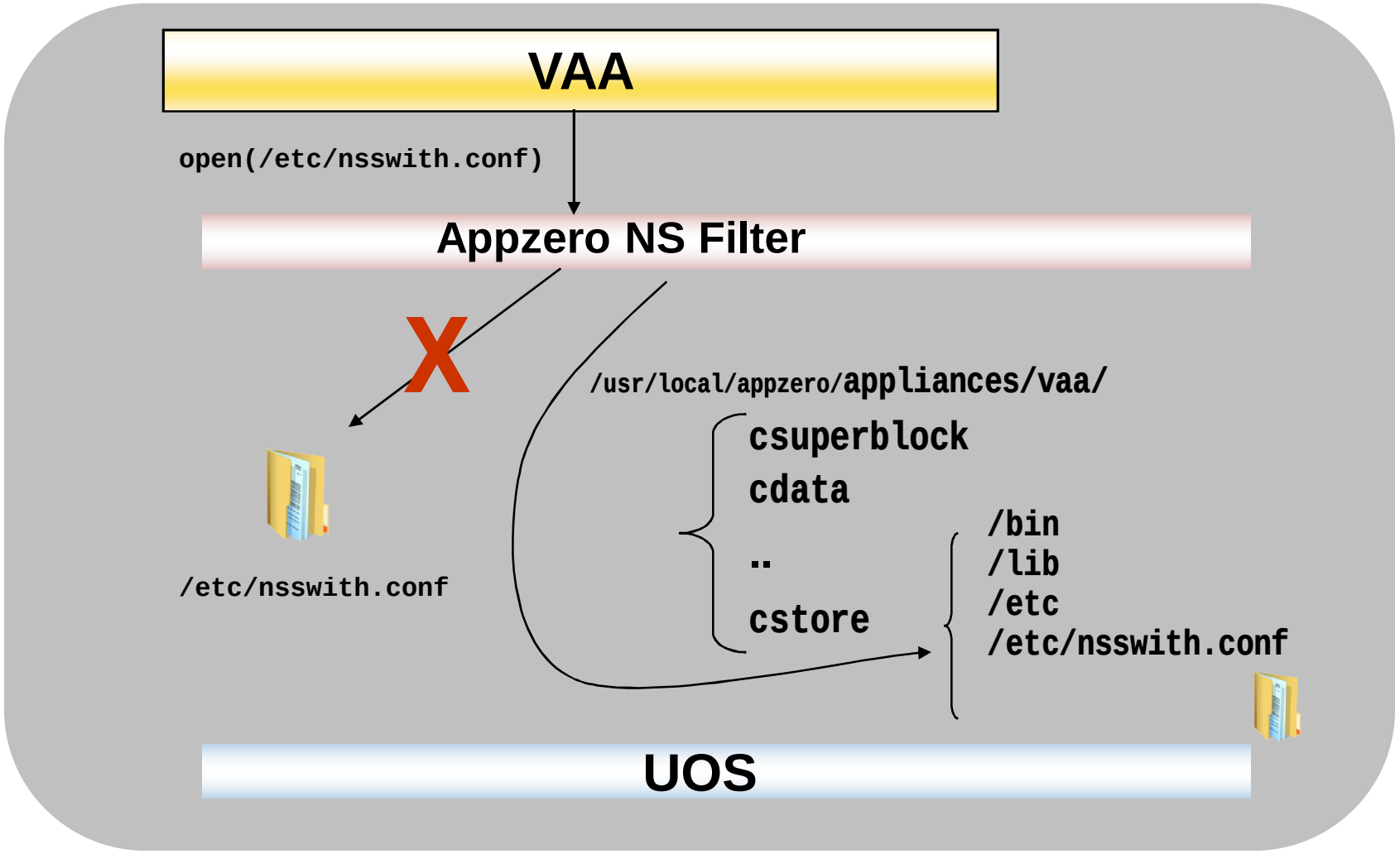
Where do the VAA files reside?



VAA file mapping: Namespaces

- Applications running as VAAs “see” a merged file system:
  - The files from the appliance itself (namespace = VAA ID)
  - The files from the UOS file system (namespace =1)
- Can be ordered so UOS files seen first
  - Appliance NS is the cstore (container store) file system
    → “`/usr/local/appzero/appliance/VAA/cstore`”
  - UOS NS is the root file system of the target system
    → “/”
- File access is configurable and defined through “filters”
  - Include all except …
  - Deny all except …
- Example:
  - Cannot see “/etc” from UOS, but can see “/exports”
  - Cannot see “/home” from appliance, but can see it from UOS

VAA file mapping: search order and filter



Configure networking and test

VAA optionally keeps the network identity and hostid of the source system

AppZero Controller on target machine

SOURCE
Solaris 2.6

MAC = 08:00:20:b1:41:47
Client name = old app

IP = 10.0.0.6

DHCP server

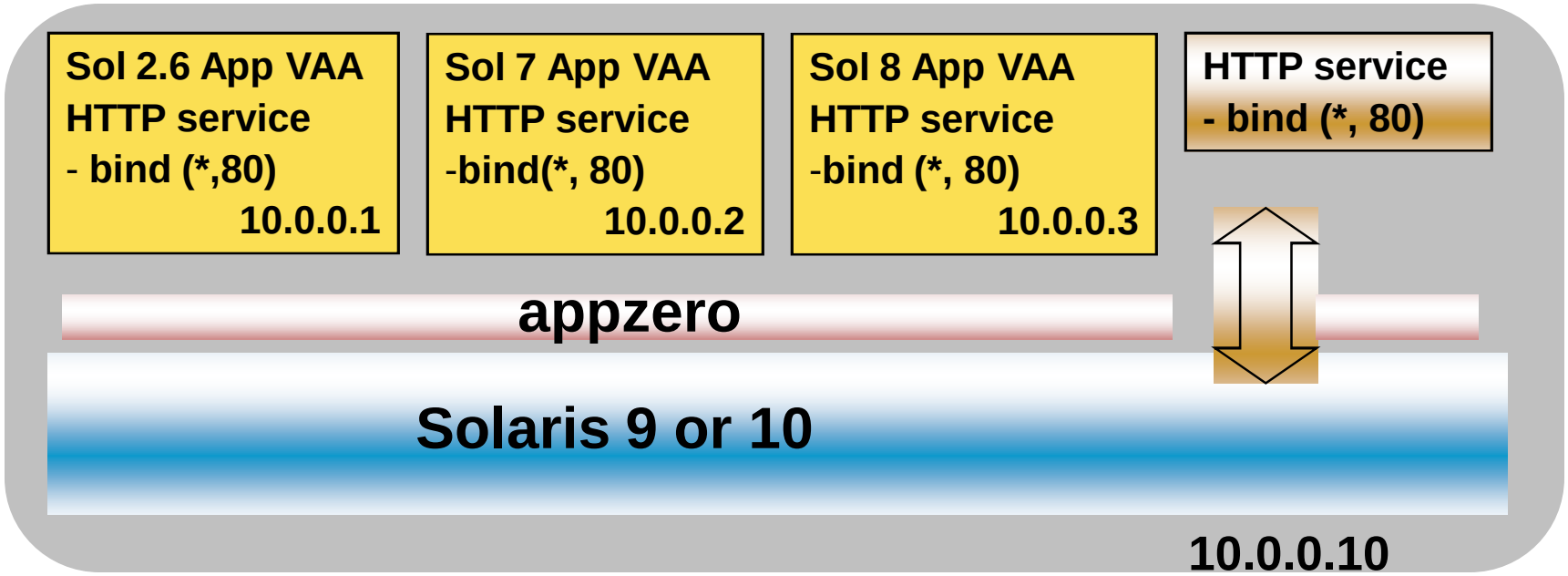
TARGET
Solaris 10

VAA - Solaris 2.6 App
appzero
solaris

IP = 10.0.0.6

IP = 10.0.0.10

Resource contention and isolation



AppZero Controller: managing VAAs

- [root@tps31 bin]# ./appzctrl <*cmd*>
  - *where CMD is:*

    - **list** - Display list of appliances

    - **dock** – Register an appliance
    - **undock** – Unregister an appliance

    - **startup** - Runs the defined startup script for an appliance
    - **shutdown** - Runs the defined shutdown script for an appliance
    - **run** - Runs an executable inside an appliance

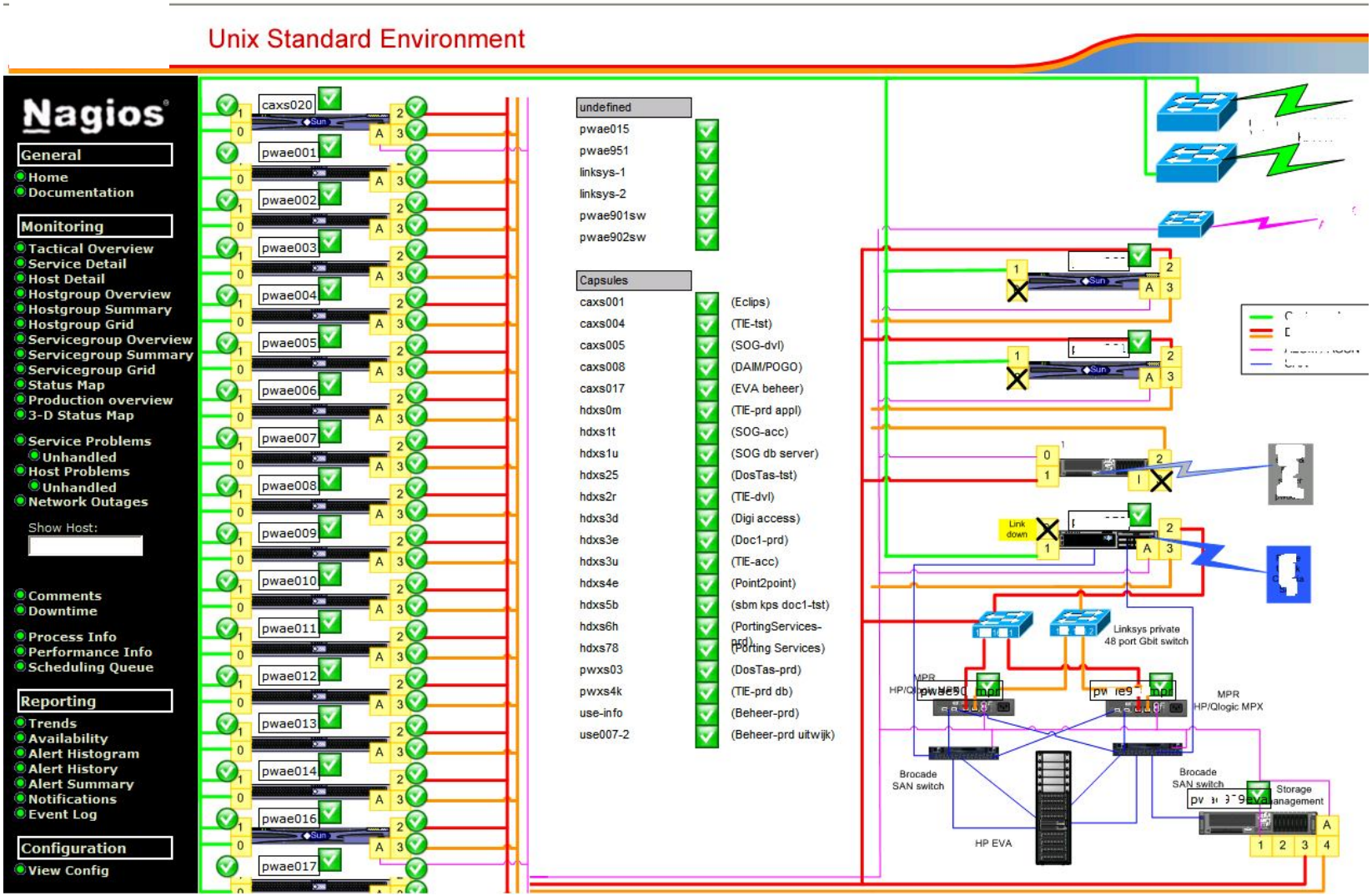
    - **procl** - Display appliance process list
    - **prop** - Display appliance property
    - **edit** - Edit appliance property

    - **cprop** - Display controller properties
    - **cedit** - Edit controller properties
    - **compress** - Compress an expanded fileset into a .con file
    - **uncompress** - Expand an appliance's .con file
    - **upgrade** - Upgrade an appliance
    - **sysversion** - Displays controller and kernel module version

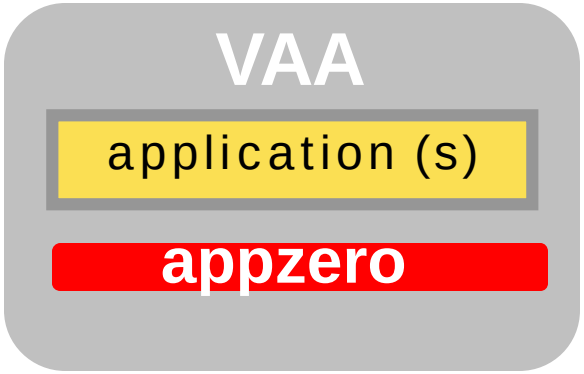
Appzctrl list

```
[root@server bin]# ./appzctrl  list
Name                                 Type        Expanded State
==================================== =========== ======== ====================
hostfs                               HostFS               Docked
app1s26                              Virtualized     Y    Started
app1s26.con                          Virtualized     N    Undocked
app2_1                               Virtualized     Y    Undocked
app2_1.con                           Virtualized     N    Needs Upgrading
osstst02.con                         Virtualized     N    Needs Upgrading
s26test                              Virtualized     Y    Stopped
s26test.con                          Virtualized     N    Needs Upgrading
Sol26Ultra1-08                       Virtualized     Y    Undocked
Sol26Ultra1-08.con                   Virtualized     N    Needs Upgrading
sol7virt                             Virtualized     Y    Stopped
sol7virt.con                         Virtualized     N    Undocked
WebApp_CC                            Virtualized     Y    Undocked
WebApp_CC.con                        Virtualized     N    Needs Upgrading
[root@server bin]#
```

Integrations



What type of application can I put on a VAA?



- Custom and proprietary applications
- Of the shelf applications.
- A simple “hello world” type of application
- A Web server
- A database (DBMS)
- A complex ERP systems
- Etc……..

Do I have to change or recompile my app? → **NO**

Note: apps with proprietary kernel drivers require specific conditions

Legacy apps on Oracle Solaris 10 on Oracle SPARC

Immediately:

- Reduce disaster scenario risk for valuable application assets that are no-longer supportable
- Enjoy performance and longevity gains of modern Oracle SPARC systems
- Increase efficiency with Oracle VM Server for SPARC
- Improve auditability by having access to a wider array of Oracle SPARC systems available today (as opposed to older, hard-to-get systems)
- Leverage Oracle Solaris 10 innovations such as Fault Management Architecture, Service Management Facility, ZFS, Resource Management and scheduling, security, zones
- Reduce risk of running unsupported versions of applications on an unsupported version of OS
- Enable business consolidation; reduce data center carbon foot-print

Oracle's Lifetime Support

- Oracle offers Lifetime Support – for Software, Hardware and Operating Systems
  - Three stages: Premier, Extended, Sustaining
  - For details, please see:

http://www.oracle.com/us/support/lifetime-support/index.html

VAAs and Oracle Solaris Zones

- Basic VAA
  - No virtual networking
  - Doesn't have Identity. UOS identity inherited
  - Globally compared with chroot environment. (but is not a chroot environment)
  - Application isolation
- Virtualized Network
  - Manually set network identity
  - Basic capsule capabilities plus identity mapping
    - Own Identity IP address, hostname, MAC, system ID are virtualized with values defined in the appliance properties
- AppZero Controller installed on the Global Zone
  - VAA managed across Zones

You're probably wondering about ….

- Level of effort: no re-engineering
  - Familiarize staff with AppZero solution
  - Virtualize/encapsulate application and test
  - Implementation is a copy, not an installation
- Time frame: a day to a week depending on complexity
- Skill level: application-knowledgeable admin or analyst
- Cost: one time costs
  - Creator $20K, VAA $7,500, Annual maintenance 20%
  - VAA can be used in dev/test/pre-production/production/cloud/DR concurrently

And maybe ….

- Where it won't work:
  - Solaris on x86 platforms, Solaris 2.5, Solaris to/from Linux, Windows, or UNIX
- What about applications with kernel drivers?
  - Doable under specific conditions
- Performance overhead: a non-issue
  - Application performance tends to improve running on current OS and new hardware
    - (%0 pure CPU, %3 business apps,%10 compilers.)
- Reliability: years of production use at large accounts

AppZero OS platform support

Solaris (SPARC)

Solaris 2.6, 2.7, 9, 10 VAA Support 32/64 bit

Windows

Windows XP	
Windows 2003	32/64 bit
Windows 2008	32/64 bit
Windows 2008 R2	64 bit
Windows 7	32/64 bit

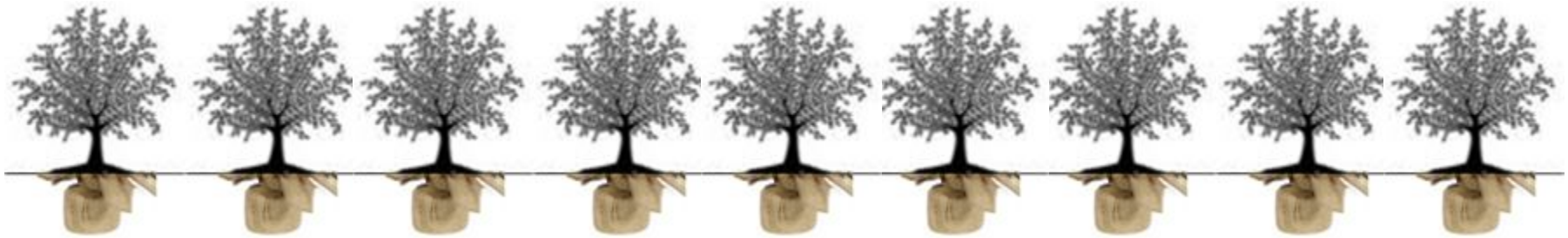
Linux

Linux 2.6 kernel:

REL 5.x,	32/64 bit
CentOS 5.x	32/64 bit

AppZero application virtualization summary ….

- Package once, run everywhere
- Eliminate application/application conflicts (sandbox)
- Run application on new hardware -- better performance/lower cost
- Server consolidation
- Continuity and disaster recovery



Next steps….

- No-charge “AppZero Application Assessment ”
  - Technical analysis of your application environment
    - App components
    - Services
    - 3rd party dependencies
    - Kernel modules
    - Fit with AppZero
- Connect directly with us
  - Greg O’Connor gregо@appzero.com
  - Valerie O’Connell voconnell@appzero.com
- Share this recorded session with your colleagues at www.appzero.com
- For more information or product download visit : www.appzero.com

appzero.com

Questions?

appzero.com

AppZero Solaris in action ….

- Large Telecommunications Provider
  - Large database used in development and support of hardware offering
  - Evolved over time…faced a choice: start over or use AppZero
  - AppZero within full production environment: “It just runs…no need to manage”
- Financial Investment Firm
  - Legacy customer information and data required for audit purposes
  - Application and database virtualized with AppZero in production
  - Originally intended to be an interim solution… now, no plans to change
- Systems Integrator
  - Providing support for large telecom company in EMEA
  - Single instance of application cloned or duplicated…then deployed
- Software Vendor
  - Build customer environment internally to replicate issues and provide technical support
  - Support time reduced from weeks down to days/hours

Oracle Solaris Binary Application Guarantee Program

- Guarantees portability within guidelines
- Tool available to inspect an app
- Can require analysis and recompilation
- In practice, many apps not eligible
  - C/C++ only, no assembly language
  - Third party libraries
  - Deprecated interfaces
  - Compiler dependencies
  - Statically linked libraries