jbox

A Pure Java Cluster Node

JAOO 2002

Michael Ringgaard (mringgaa@csc.com)
Bjarne Hansen (bhansen4@csc.com)

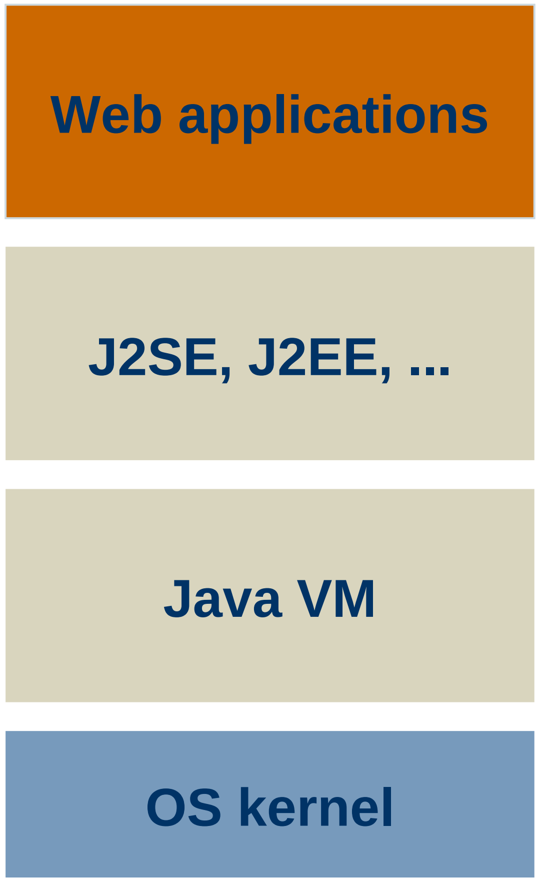
Computer Sciences Corporation

Welcome & Agenda

- Welcome
  - Bjarne Hansen, bhansen4@csc.com
    Computer Sciences Corporation
  - Michael Ringgaard, mringgaa@csc.com
    Computer Sciences Corporation
- Agenda
  - What is a jbox?
  - Why would we want a jbox?
  - What can a jbox be used for?
  - How to build a jbox

What is a jbox?

- A server appliance for Java programs
  - Requires only power and a network connection
  - No monitor, keyboard, or mouse
- Built for standard Intel based PC
  - Cheap, simple and powerful
- Runs only one process: the Java VM
  - Specifically the HotSpot Java VM for Windows
  - Relies on a small and efficient kernel
- Transforming application servers to appliances



Appliances

- Characteristics of appliances
  - Unpack, connect, use...
  - Can’t rely on experts to operate ..
  - Must require just about zero maintenance
- Would be nice characteristics for an IT business system!
- By the way...next generation of home appliances: Broadband router, DHCP, DNS,...

jbox.dk

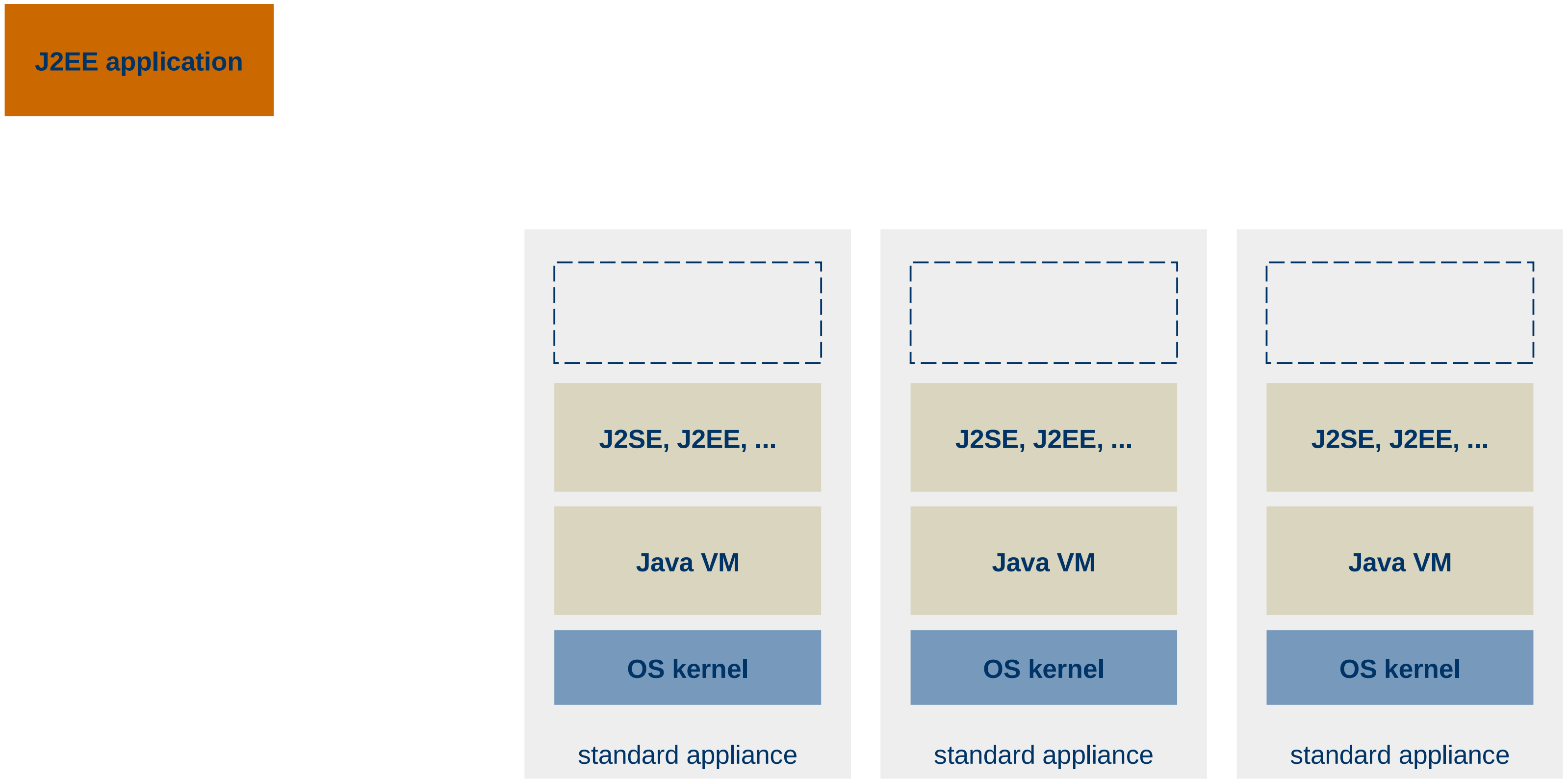
Why a *Java* Appliance?

- Need effective development and execution platform
  - Hardware:
    - Before: Exotic processor/hardware
    - Now: Complies with PC specification
  - Development platform
    - Before: C, C like variant, or assembler
    - Now: OO, VM, garbage collection
- Cost effective
  - Extremely cheap hardware
  - Develop on PC, execute on appliance
  - Wide selection of development environments, tools, utilities...
  - No specialized developers
- *Java is a powerful and rich environment yet simple enough to use in an appliance*

What can a jbox be used for?

- What would we like to achieve?
  - Apply the virtues of traditional appliances to IT business systems
  - Apply the effective software development tools, utilities, and methodologies to appliance development
- As a Java server appliance
  - Ideal development environment to develop, deploy and maintain software for appliances
- As a Java cluster node
  - It’s better to own 100 appliances than 100 application servers

Just add water...



Ready in just 5 minutes!!!

jbox.dk

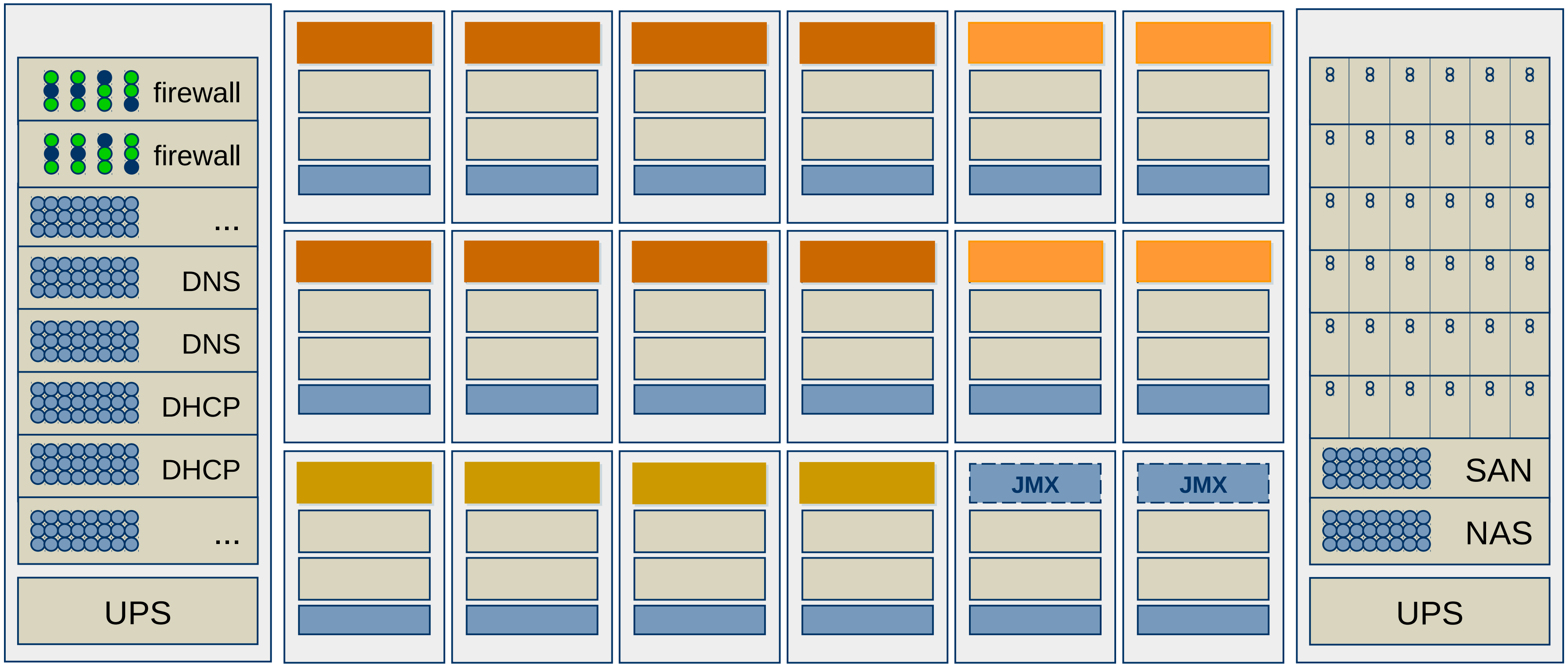
Clusters of Appliances

- Clustering support is a feature of specific J2EE server products
  - Focus on transparency (developer, user)
- Custom designed distributed architectures
  - J2EE +
  - Jini, JavaSpaces
  - P2P, JXTA, ...

Appliance Management

- Standalone appliances can be managed using browser interface
- Most J2EE servers has built-in web based management consoles
- Appliance clusters requires special attention on deployment and configuration issues
  - How do you deploy applications to many nodes
  - Centralized application and configuration repository (JMX).
- *Manage applications, not servers*

Application Clusters



jbox – A Pure Java Cluster Node

How to build a jbox...

jbox.dk

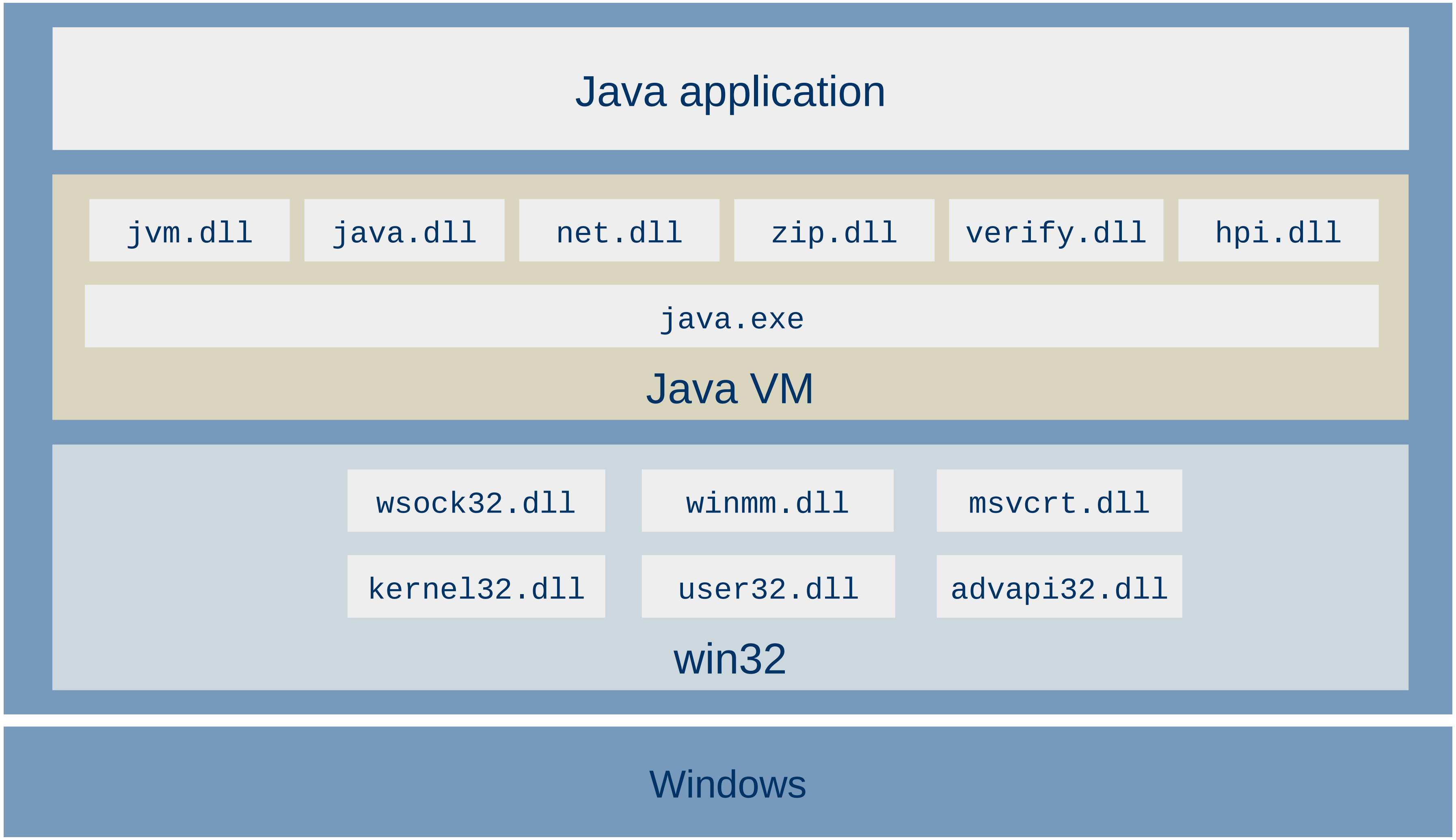
Java and the operating system

- What is actually going on under the hood when you run a Java application?
- How is the JVM using the operating system?
- What features of the operating system are used by a Java server application?
- Do you really need an operating system?

What is an operating system?

- Hardware Abstraction Layer
- Resource Manager
- Bootstrap Loader
- Application Programming Interface
- Virtual Machine Implementation
- Utility Collection
- One-stop-shopping User Entertainment System

Java VM on Windows



194 Windows API calls used

KERNEL32
CloseHandle
CreateEventA
CreateFileA
CreatePipe
CreateProcessA
CreateSemaphoreA
DebugBreak
DeleteFileA
DisableThreadLibraryCalls
DuplicateHandle
EnterCriticalSection
FindClose
FindFirstFileA
FindNextFileA
FlushFileBuffers
FormatMessageA
FreeLibrary
GetCurrentDirectoryA
GetCurrentProcess
GetCurrentThread
GetCurrentThreadId
GetEnvironmentVariableA
GetExitCodeProcess
GetFileAttributesA
GetLastError
GetLogicalDrives
GetModuleFileNameA
GetNumberOfConsoleInputEvents
GetProcAddress
GetStdHandle
GetSystemDirectoryA
GetSystemInfo
GetSystemTime
GetSystemTimeAsFileTime
GetTempPathA
GetThreadContext
GetThreadLocale
GetThreadPriority
GetThreadTimes
GetTimeZoneInformation
GetVersionExA
GetWindowsDirectoryA
InitializeCriticalSection
InterlockedDecrement
InterlockedIncrement
IsDBCSLeadByte
LeaveCriticalSection
LoadLibraryA
PeekConsoleInputA
PeekNamedPipe
QueryPerformanceCounter
QueryPerformanceFrequency
ReleaseSemaphore
RemoveDirectoryA
ResetEvent
ResumeThread
SetConsoleCtrlHandler
SetEndOfFile
SetEvent
SetFileAttributesA
SetFilePointer
SetFileTime
SetHandleInformation
SetThreadContext
SetThreadPriority
Sleep
SuspendThread
SystemTimeToFileTime
TerminateProcess
TlsAlloc
TlsGetValue
TlsSetValue
VirtualAlloc
VirtualFree
VirtualQuery
WaitForMultipleObjects
WaitForSingleObject
WideCharToMultiByte

USER32
MessageBoxA

ADVAPI32
GetUserNameA
RegCloseKey
RegEnumKeyExA
RegOpenKeyExA
RegQueryInfoKeyA
RegQueryValueExA

WSOCK32
__WSAFDIsSet
accept
bind
closesocket
connect
gethostbyaddr
gethostbyname
gethostname
getprotobyname
getsockname
getsockopt
htonl
htons
ioctlsocket
listen
ntohl
ntohs
recv
recvfrom
select
send
sendto
setsockopt
shutdown
socket
WSACleanup
WSAGetLastError
WSAStartup

MSVCRT
new
delete
__dllonexit
__mb_cur_max
_access
_adjust_fdiv
_assert
_beginthreadex
_CIfmod
_close
_control87
_endthreadex
_errno
_except_handler3
_finite
_fstati64
_ftol
_fullpath
_get_osfhandle
_getdcwd
_getdrive
_initterm
_iob
_isctype
_isnan
_lseeki64
_mkdir
_onexit
_open
_open_osfhandle
_pctype
_purecall
_read
_setjmp3
_setmode
_stat
_stati64
_strdup
_vsnprintf
_write
abort
atof
calloc
exit
fclose
fflush
fgets
fopen
fprintf
fputc
free
getc
getenv
isalnum
isspace
longjmp
malloc
memmove
printf
putchar
qsort
raise
realloc
rename
signal
sprintf
sscanf
strchr
strerror
strncmp
strncpy
strrchr
strstr
strtol
toupper
vfprintf
vsprintf

WINMM
timeEndPeriod
timeBeginPeriod
timeGetTime

jbox.dk

OS Services

Kernel context

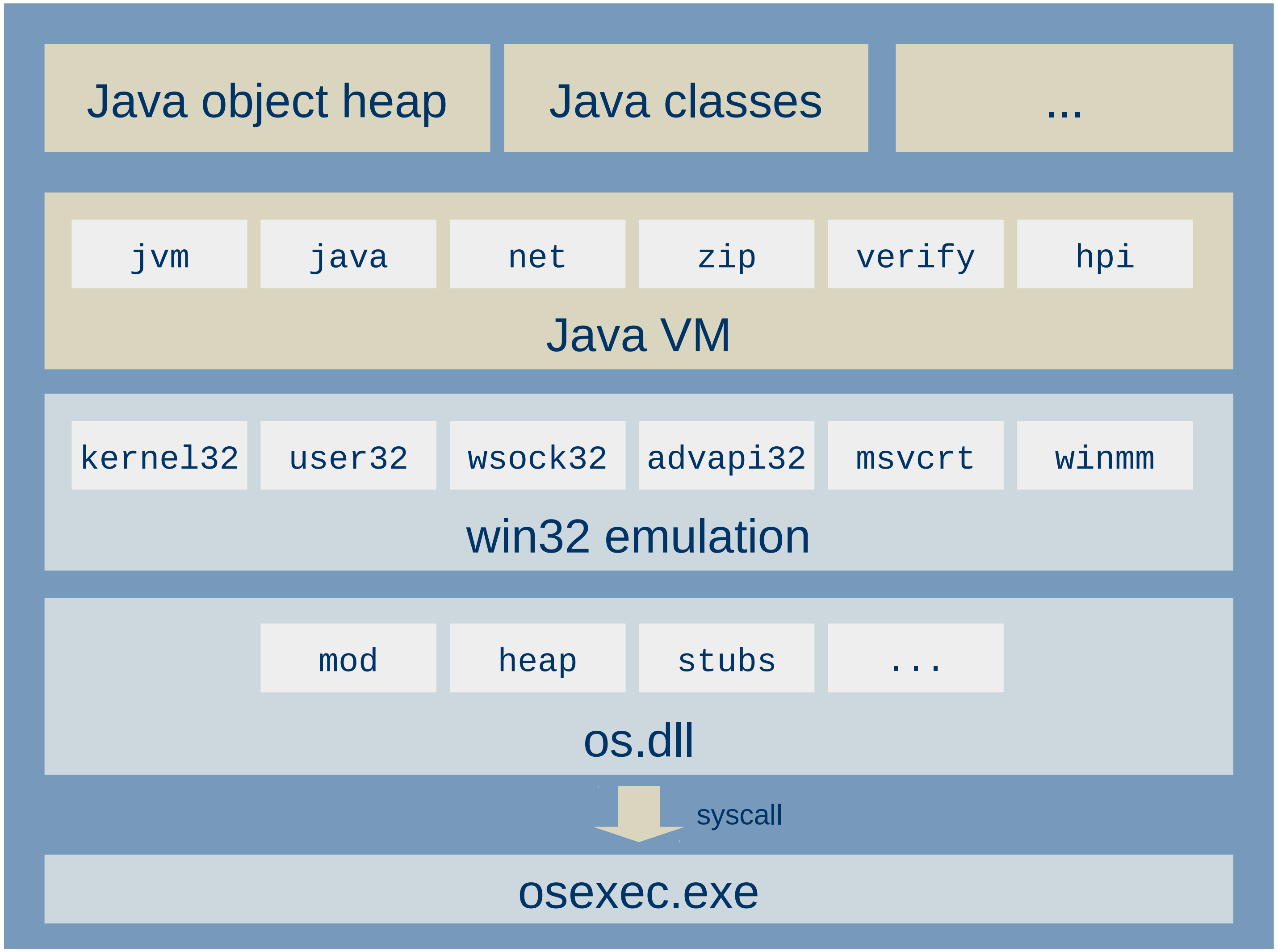
- file
- network
- virtual memory
- threads
- synchronization
- time

User context

- resolver
- heap
- modules
- critical sections
- thread local storage

jbox.dk

JavaOS emulator



jbox.dk

Design objectives

- Build a kernel for executing Java server application on appliances
- Use existing HotSpot VM
- Small,simple, fast but complete kernel
- Run on standard PC hardware (IA-32)
- Develop using Microsoft Visual C
- Use standard PE (EXE/DLL) executables

How to build a JavaOS kernel

- ✓ Use the existing HotSpot VM
- ✓ Write stub DLLs for Win32 DLL
- ✓ Implement user mode components:
  - loader, heap, tls, critsect, resolver...
- Implement a boot loader to load kernel
- Implement kernel
  - memory management, thread control, device I/O and networking.

jbox.dk

Core kernel services

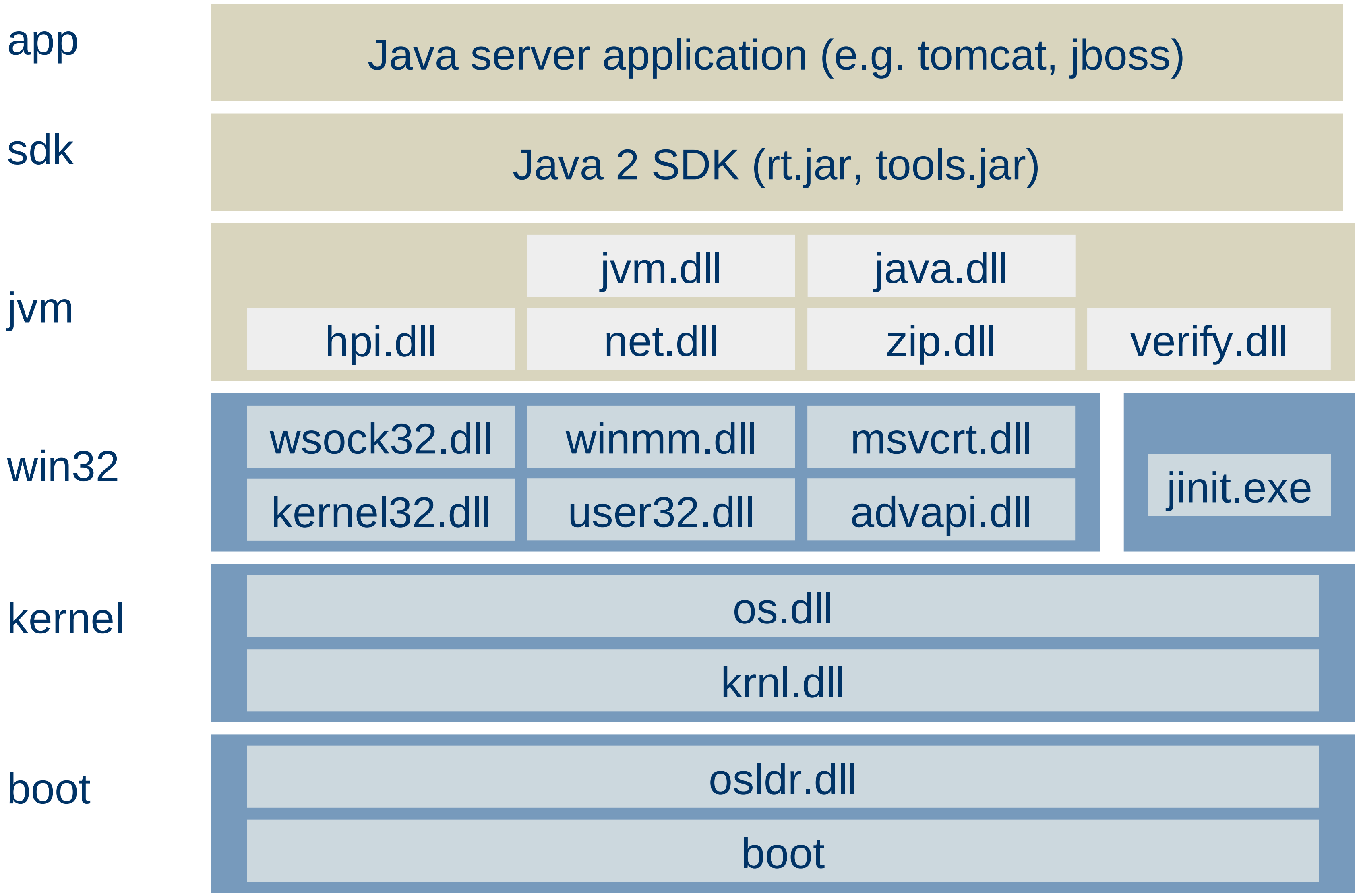
- System booting and application loading
- Memory Management
  - Virtual memory mapping
  - Physical memory allocation and paging
  - Heap allocation and module loading and linking
- Thread Control
  - Thread scheduling and trap handling
  - Thread context
  - Thread synchronization and timers
- I/O Management
  - I/O bus and unit enumeration
  - Block devices and file systems
  - Stream devices
  - Packet devices (NIC) and networking (TCP/IP)

jbox.dk

OS Development on the Internet

- There are lots of information and code on the internet on OS topics:
  - Linux kernel code (www.kernel.org)
  - IA-32 Reference Manual (www.intel.com)
  - TCBs and u-kernels (Jochen Liedtke, i30www.ira.uka.de/teaching/coursedocuments/47/)
  - DNS Resolver (ISC BIND lwres, www.isc.org)
  - TCP/IP Stack (Adam Dunkels, www.sics.se/~adam/lwip/)
  - Heap Allocator (Doug Lea, http://gee.cs.oswego.edu/dl/html/malloc.html)
  - Bochs (bochs.sourceforge.net) and WMWare simulators (www.vmware.com)
  - IDE Disks (Hale Landis, www.ata-atapi.com)
  - ...

Architecture layers



jbox.dk

sanos API

file

canonicalize
chdir
chsize
close
dup
flush
format
fstat
fstatfs
futime
getcwd
getfsstat
ioctl
link
lseek
mkdir
mount
open
opendir
read
readdir
readv
rename
rmdir
stat
statfs
tell
umount
unlink
utime
write
writev

socket

accept
bind
connect
getpeername
getsockname
getsockopt
listen
recv
recvfrom
send
sendto
setsockopt
shutdown
socket

time

clock
gettimeofday
settimeofday
time

memory

mlock
mmap
mprotect
mremap
munlock
munmap

thread

beginthread
endthread
epulse
ereset
eset
getcontext
getprio
gettib
gettid
mkevent
mksem
resume
self
semrel
setcontext
setprio
sleep
suspend
wait
waitall
waitany

system

config
dbgbreak
exit
loglevel
panic
peb
syscall
syslog

critsect

csfree
enter
leave
mkcs

tls

tlsalloc
tlsfree
tlsget
tlsset

heap

calloc
free
mallinfo
malloc
realloc

module

exec
getmodpath
getmodule
load
resolve
unload

resolver

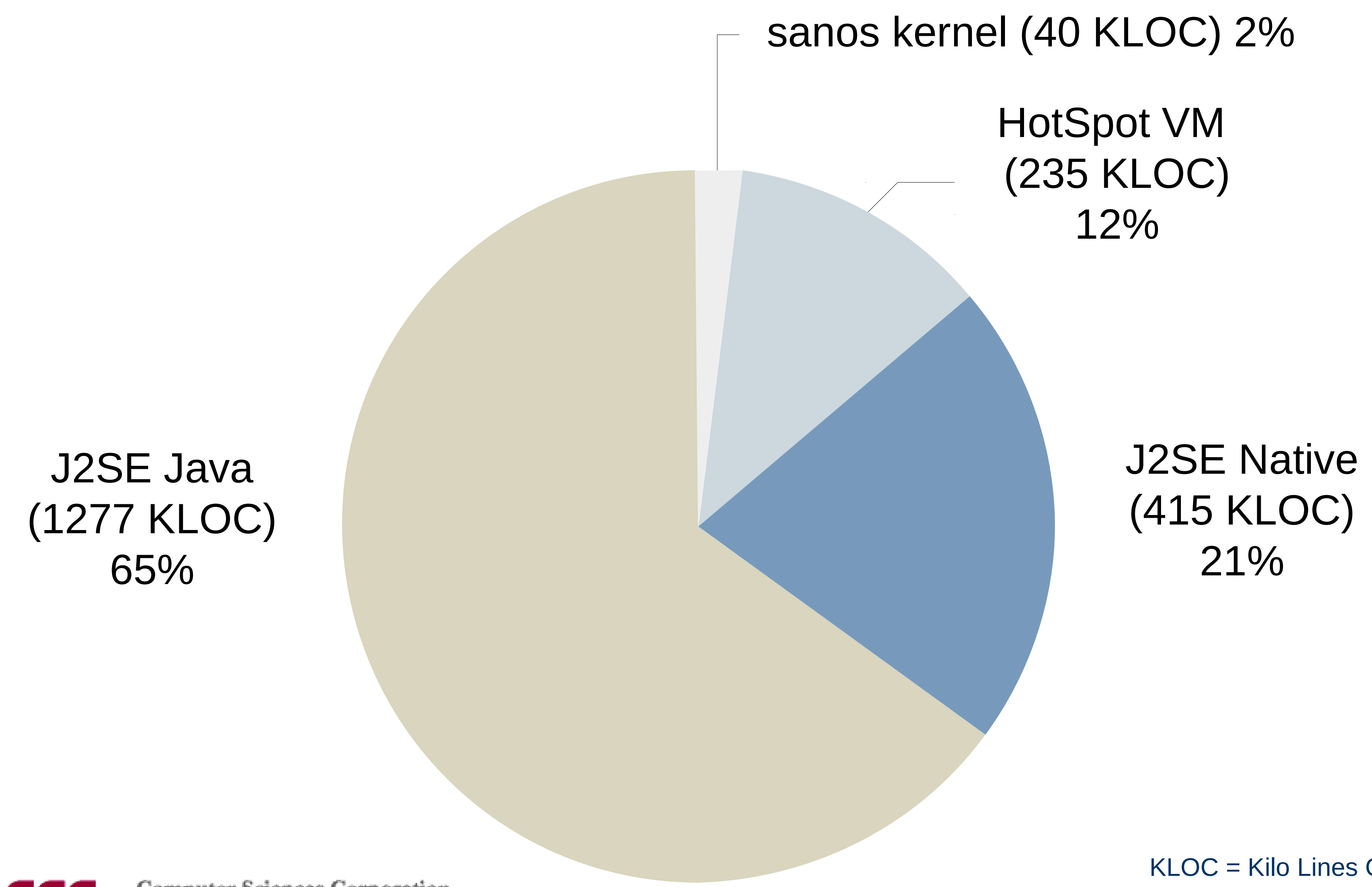
dn_comp
dn_expand
res_mkquery
res_query
res_querydomain
res_search
res_send

netdb

gethostbyaddr
gethostbyname
gethostname
getprotobyname
getprotobynumber
getservbyname
getservbyport
inet_addr
inet_ntoa

jbox.dk

Where is the code?



KLOC = Kilo Lines Of Code

Conclusion

Is Java an operating system ?

No, but if you add 2% to the code that is already there it can become an operating system!

Did we write our own operating system ?

No, we only made the kernel, SUN did the remaining 98%!

sanos has been released as open source (BSD license) and is available for download at www.jbox.dk