

Porting UNIX Applications to AS/400 PASE

P03

ITSO AS/400 e-business University/Technical Forum

Gottfried Schimunek



© 2000 IBM Corporation

Acknowledgements



- ★ **This presentation is a result of the 'Porting UNIX Applications with AS/400 PASE' residency in April/May 2000**

- ★ **The deliverables of this residency are:**
 - the redbook - SG24-5970 (available 6/2000 as redpiece)
 - a hands-on lab and
 - this presentation

- ★ **The residents were:**
 - Janet Krueger, D.H. Andrews, Rochester MN
 - Jim Shupe, IBM Rochester Technical Support
 - Daryl Spartz, IBM Rochester, Custom Technology Center
 - Valuable information and contributions came from the
 - IBM Partnerworld Porting team, Kay Tate, Charlie Quigg and Pam Bowen
 - IBM PASE development team

AS/400 PASE Overview



★ AS/400 7 Portable Application Solutions Environment

- Is an integrated AS/400 runtime for porting UNIXJ applications
- Is not a UNIX OS on AS/400
- Is designed to expand the AS/400 solutions portfolio

★ Agenda

- Overview of what PASE is and does
- PASE from a UNIX programming perspective
- PASE from an AS/400 operations perspective
- Introduction to the PASE lab

AS/400 PASE Overview



Announcement at PartnerWorld 2000 (01/24/2000) for V4R4

- NEW IBM AS/400 TECHNOLOGY DRAMATICALLY EXPANDS UNIX PORTING OPTIONS FOR ISVS
- San Diego, CA., January 24, 2000- IBM today announced a new technology that dramatically expands porting options for Independent Software Vendors (ISV) to run UNIX applications on the AS/400.
- The new capability, called OS/400 Portable Application Solutions Environment (PASE), further enhances the integration of core application suites, such as enterprise resource planning (ERP), with supply chain and customer relationship management applications on a single, multi-function server.
- In concert with the IBM announcement, IBM Premier Business Partner J.D. Edwards announced plans to offer its Numetrix advanced planning and scheduling solution on AS/400, one of the first industry-leading ERP solution providers to use the new technology. Both announcements were made today in San Diego, at PartnerWorld 2000, IBM's major Business Partner show.
- "PASE is designed to meet requirements of highly compute-intensive applications that favor a UNIX environment," says (name), JD Edwards. PARTNER X. "With its improved porting speed for UNIX applications, PASE allows us to bring APPLICATION NAME to market faster on AS/400.
- PASE is a fully integrated component of OS/400 that uses a subset of AIX functionality. It exploits the PowerPC processor's ability to switch between AS/400 and UNIX run-time modes. Applications deployed using PASE run natively on the AS/400 and take full advantage of integration with its file systems, security and DB2/400 database. PASE is available immediately in OS/400 Version 4 Release 4.
- PASE extends AS/400's ability to provide a consolidated server platform - combining multiple application workloads, offering robust workload management services such as logical partitioning, and according to ANALYST X, "Demonstrating the flexibility to combine a diverse set of application requirements in a single server."
- The AS/400, with an install base of 650,000 machines ranging from small and medium business to the enterprise market, extends from its core strengths of ERP applications, to incorporate groupware, supply chain and e-commerce applications, all fully integrated on a single server. Applications can run independently or they can be divided into logical partitions (LPAR) to fully isolate applications with specific performance, language or time zone requirements.
- PASE applications can be integrated with AS/400 file systems and DB2/400, Java, Lotus Domino and OS/400 Integrated Language Environments (ILE) applications.
- Visit www.as400.ibm.com for more information.
- IBM and AS/400 are registered trademarks, and AS/400e is a trademark, of International Business Machines Corporation. UNIX is a registered trademark of the Open Group. All other registered trademarks, trademarks and service marks are the property of their respective owners.

Enhancements in V4R5 for more general use

AS/400 PASE Architecture



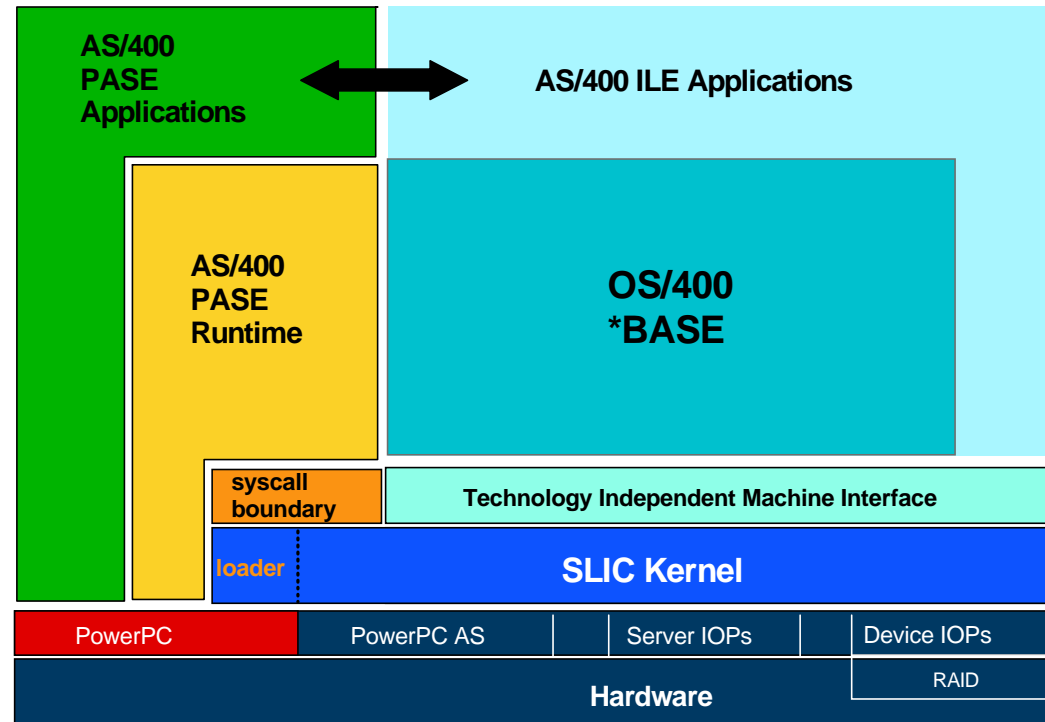
- ★ **An integrated OS/400 runtime for porting selected UNIX applications**

- Uses a subset of AIX libraries
- Not an operating system or an emulated environment

- ★ **Exploits PowerPC's ability to switch runtime modes**

- ★ **PASE applications**

- Integrated with AS/400 file systems
- Can call DB2/400, Java and ILE programs
- Exploit all aspects of AS/400 operations environment



AS/400 PASE Architecture



- AS/400 PASE uses the AS/400 processor's ability to switch runtime modes to enable running AS/400 PASE applications concurrent with AS/400 ILE applications. In fact, AS/400 PASE is intimately integrated with the ILE environment, the AS/400 file systems and DB2 Universal Database for AS/400 (see Figure 2 for a graphical depiction of the AS/400 PASE architecture). It can easily call Java and AS/400 ILE applications and thus exploit all aspects of AS/400 operations environment.
- RS/6000 and AS/400 share a common PowerPC chip. This hardware base has the ability to switch between runtime modes: addressing tags active to execute AS/400 64 bit applications; addressing tags inactive to execute 64- or 32-bit AIX applications. AS/400 PASE exploits this switching capability to execute 32-bit AIX applications on the AS/400 within AS/400 jobs. Since AS/400 PASE applications execute directly on the hardware in PowerPC mode, computationally intensive applications use the processor without any additional layers overhead. In addition to AS/400 PASE accessing the Power PC instruction set on the AS/400 directly, it can access the AS/400 PASE shared libraries (which are the ported AIX libraries), or, from the shared libraries, it can access a syscall interface to the SLIC Kernel. AS/400 PASE can also call out into the OS/400 ILE environment and access ILE applications and the AS/400 DB2 UDB.
- The System Licensed Internal Code (SLIC) kernel controls the use of the hardware and the type of address space that can be used. SLIC is a common kernel under the AS/400 PASE shared libraries and OS/400. The services that SLIC provides are common to both the Integrated Language Environment (ILE) and AS/400 PASE environments, creating a well-integrated system, regardless of which environment the application is using. The environments share the same file systems, security mechanisms, threading mechanisms, and can even share memory and connect to the same sockets (with a little care to note pointer format expectations and character set being shared).
- SLIC and OS/400 work together to provide an object-based OS on AS/400. AS/400 PASE has not caused creation of new system objects -- it uses the ones that were already built to support standard APIs for ILE. This facilitates sharing between application code and also gives AS/400 PASE an operational view that is just what AS/400 customers expect. AS/400 PASE applications look like ILE or other running applications to AS/400 operators. Programs and files for new AS/400 PASE applications are saved and restored in familiar ways, without the operator having to know that they run in a new way in operation.
- As OS/400 sits architecturally on top of the Technology Independent Machine Interface (TIMI) AS/400 PASE sits architecturally on top of the syscall interface. It looks architecturally like AIX in this respect. There is a protection layer between the kernel services for AS/400 PASE and non-privileged system or user code as there is for OS/400, but the syscall is not designed to provide the same virtual machine as the TIMI. OS/400 has the capability to move applications that maintain their intermediate form, to new hardware architectures (i.e., CISC to RISC) without recompiling. This level of compatibility is not provided by the AS/400 PASE Model.
- AS/400 PASE applications will be recompiled when the customer moves to new hardware architecture, as one would expect with an AIX application making the same transition, since they have access directly to the hardware processor.

AS/400 PASE Basics



■ Business Considerations

- AS/400 PASE is supported on AS/400e Series (9/97) and newer hardware.
- AS/400 PASE is Option 33 of OS/400 beginning in V4R4
- Requires an AIX workstation, with AIX and tools to generate PowerPC code for AS/400 PASE application development

■ Technical Considerations

- AS/400 PASE is an integrated runtime
- Provides 32-bit application support
- Does not host UNIX databases or their clients (e.g. Oracle, Sybase, etc.) at this time
- Does not support kernel extensions or device drivers for specialized I/O devices

AS/400 PASE Basics



Business Considerations

- AS/400 PASE is supported on AS/400e Series (9/97) and newer hardware.
 - Most RISC processors used in AS/400 systems are designed to support both PowerAS and PowerPC architecture and provide the supervisor (SLIC) with the ability to switch architectures on demand, but hardware implementation issues prevent us from using PowerPC mode on some existing systems without jeopardizing user data or system integrity. AS/400 PASE is only supported on the AS/400e Series (hardware introduced around 9/97). SLIC prevents AS/400 PASE programs from running on any system that is not AS/400e Series hardware.
- AS/400 PASE is Option 33 of OS/400 beginning in V4R4
 - Nominal fee per partition where AS/400 PASE is used
- Requires an AIX workstation, with AIX and tools to generate PowerPC code for AS/400 PASE application development
 - AS/400 PASE development requires an AIX 4.3.3 system to run the compiler (xlc, xIC, or some other AIX compiler) and linker (ld command). The AIX assembler for PowerPC can also be used.

Technical Considerations

- AS/400 PASE is an integrated runtime
- Provides 32-bit application support
- Does not host UNIX databases or their clients (e.g. Oracle, Sybase, etc.) at this time
- Does not support kernel extensions or device drivers for specialized I/O devices

AS/400 PASE - Brand Value



- ★ **Opens AS/400 to new classes of apps that previously weren't possible**
- ★ **Attracts new ISVs to AS/400, IBM brand (mindshare)**
- ★ **Provides UNIX programming model to developers; OS/400 application look and feel to customers**
- ★ **Drives down development costs by leveraging AIX technology**
- ★ **Reinforces AS/400 as an open platform**

AS/400 PASE - Brand Value



The AS/400 system's broad base of applications is continually enhanced by new applications coming to the platform from a variety of sources. Up to now, the AS/400 Integrated Language Environment accounted for the majority of C or C++ application ports, many of which originally ran on UNIX. Recently, however, the AS/400 system focused investment in Java and Domino (and even NT on the Integrated Netfinity Server) has opened up wide new application porting and modernization opportunities for solutions developers. With the latest addition of the AS/400 Portable Application Solutions Environment (AS/400 PASE), solutions developers have another option for porting UNIX applications rapidly to take advantage of the AS/400 systems extensive marketplace.

IBM's strategy for AS/400 PASE is to expand its solutions portfolio, focusing on specific industry and application segments. For example, new supply chain management solutions that integrate with ERP applications targeted at industrial and distribution industries.

Not all porting solutions on the AS/400 system equally apply to all applications. Certain applications will fit better in ILE while others will fit better in AS/400 PASE.

AS/400 PASE Key Points



- ★ Familiar AIX interfaces "on the inside" to the application
- ★ Familiar AS/400 interfaces "on the outside" to the customer
- ★ AS/400 PASE is a runtime, not an emulator -- it uses the AS/400 hardware directly and efficiently
- ★ AS/400 PASE is integrated -- it does not introduce new system security or reliability concerns
- ★ The AS/400 system architecture facilitates breakthrough technologies like AS/400 PASE

AS/400 PASE Key Points



When you are planning to port an application from AIX to the AS/400 you have two choices: you can port to the AS/400 ILE environment or you can port to the AS/400 PASE environment. The strongest aspects of AS/400 PASE are:

- One of the primary strengths of AS/400 PASE is the large number available OS services. If the UNIX/AIX APIs you use are already supported there is very little application porting to do! You can determine how your application stacks up against our supported APIs by using our “filtering” process. The front end tool can be found at <http://www.as400.ibm.com/developer/porting/apitool.html>. It is a shell script that produces a list of APIs which you send to the IBM porting team. The IBM team runs the back end analysis and returns a report to you indicating whether the application fits well on the system, and where modifications may be needed, if any. More information on this process is in Chapter 5, “Application API analysis” on page 37.
- AS/400 PASE provides a good environment for running computationally intensive applications on the AS/400 system by providing highly optimized math libraries.
- AS/400 PASE allows you to use existing AIX based build processes, which is especially useful when you have an existing, complicated process that is not readily movable to a new platform.
- AS/400 PASE is tailored so that existing AIX applications will find porting to the AS/400 PASE environment simple and friendly.
- AS/400 PASE supplies support for fork and exec which does not currently exist on the AS/400 system (except through spawn which is somewhat different).
- AS/400 PASE is currently the best tool for satisfying dependencies on an ASCII character set and for satisfying dependencies on X-Windows support.
- Applications that rely on a large number of pointers and pointer casting may have an easier port to a 32-bit AIX addressing model.
- AS/400 PASE fully supports several dialects of C, C++ and FORTRAN.
- Shell programming is well supported.

Who Is AS/400 PASE For?



★ Solution providers (initial focus)

- AS/400 providers who want to attract smaller complementary solutions
- Solution providers who are new to AS/400
- Providers with short time-to-market windows

★ Customers

- Customers who want new applications to run their businesses more effectively
- Customers who want to continue to run their AS/400s just as they have before even while using PASE
- The strength of the AS/400 architecture give us unique opportunities to deliver state of the art technologies to them

Who Is AS/400 PASE For?



There are 5 different audiences for PASE:

- executives and planners who are looking at using AS/400 PASE for portions of their solutions,
- UNIX application providers who are considering a port of their application to the AS/400 platform,
- businesses who already leverage both AS/400 and AIX solutions and who would like to consolidate their solutions into one platform,
- AS/400 shops looking to purchase or use an AS/400 PASE application,
- AS/400 shops looking to purchase a UNIX solution who want to encourage the vendor of that solution to consider AS/400 delivery and support.

AS/400 Applications Using PASE



★ AS/400 users and operators see integrated AS/400 solutions

- Do not require AIX or UNIX skills
- Use standard OS/400 operations interfaces, backup and security



★ ISVs

- Compile on AIX 4.3.3 workstation with AD tools
- See AIX programming interfaces on AS/400
- Require AS/400 product integration and support skills

AS/400 Applications Using PASE



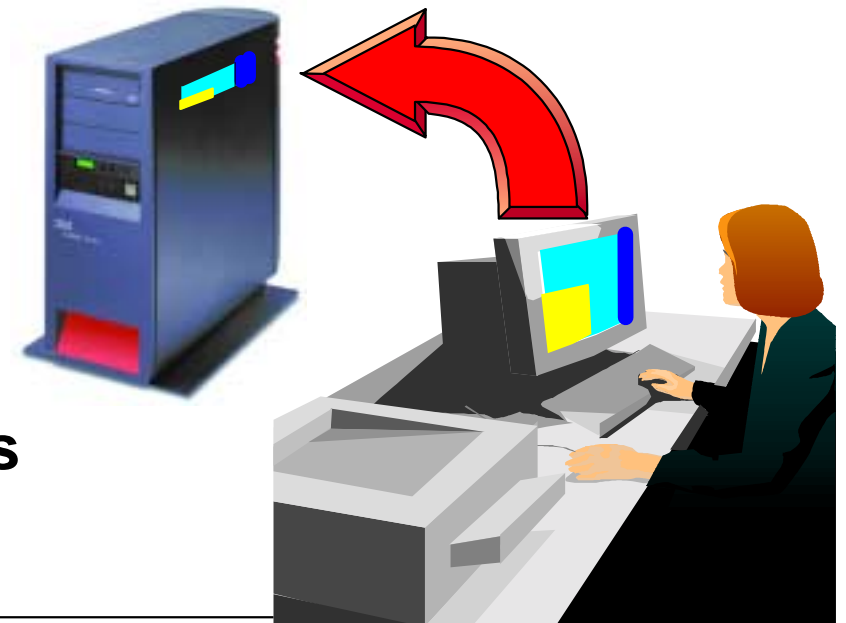
- AS/400 customers running applications ported using PASE do not have to deal with UNIX system operations. PASE applications run in a normal AS/400 job with standard work management (subsystems), AS/400 integrated file system with standard save/restore operations, and standard AS/400 security. No special operations are required.
- ISVs porting and supporting their applications using PASE on the AS/400 require a combination of AIX and OS/400 programming and support skills. PASE applications are built and compiled on an AIX workstation (Version 4.3.3). Primary application function continues to use their existing UNIX skills where they are adding their greatest value. AS/400 programming skills are required for converting database calls, providing integration with AS/400 security, and operational interfaces (installation, start-up, shutd-down, etc.).
- ISVs also require AS/400 skills in their support teams as customers will call using AS/400 instead of UNIX terminology.

Views of AS/400 PASE



- ★ **System owners see an AS/400**
- ★ **Operators/Admins see OS/400**
 - ▶ Do not require AIX or UNIX skills
 - ▶ Use standard OS/400 operations interfaces, work management, backup, and security
- ★ **Users see more complete solutions and increased productivity**

- ★ **Application developers use AIX programming interfaces**
- ★ **Some AS/400 product integration and support skills are required**



Views of AS/400 PASE



PASE can be viewed from two completely different perspectives. At run time, it is nothing more than another mode for running applications. At development time, There are two different sides of PASE; what System owners see at runtime, and what application developers see when they are building applications. The system owners do not require AIX or UNIX skills, they just have a wider portfolio of solutions to choose from. Developers not only require AIX skills, they require an AIX workstation to do development; development support is not provided on the AS/400. Both sets of users need to concern themselves with the following objects:

- AS/400 job
 - PPM code runs within a job, called from OS/400
 - Can call other OS/400 system facilities and program models from PPM code
 - Normal Work Management facilities
 - Some problem determination and performance monitoring capabilities available from OS/400 interfaces

- AS/400 Integrated File System
 - Data stored in IFS
 - Code stored in IFS
 - Normal Storage Management facilities
 - Save/restore capabilities inherent in IFS
 - Normal OS/400 install procedures into IFS
 - Normal PTF installation procedures into IFS

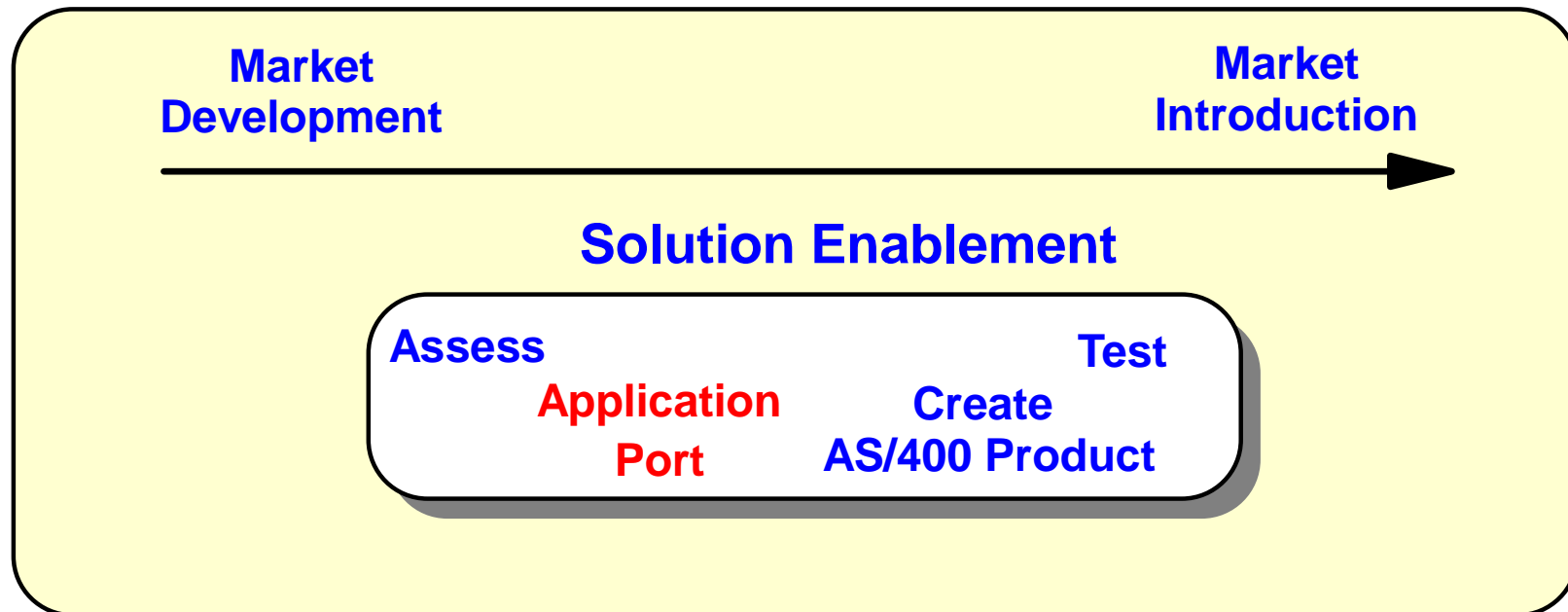
- Security
 - Fully integrated with the rest of the system (user profiles, authentication,...)

- Database Access
 - Database is callable, requires EBCDIC accommodation like ODBC (or like JVM program calls to DB2/400)

Bringing Solutions to Market



- ★ AS/400 PASE can significantly improve porting time



Bringing Solutions to Market



- Bringing an application to the AS/400 market requires an ISV to perform market development, solution enablement (including porting the application) and market introduction. Of these, PASE improves several aspects of the solutions enablement phase. Market development and introduction are common to all techniques of moving ISV applications to the AS/400 market.
- When using PASE for enablement, some application assessments can be faster since character set and pointer size assumptions do not change from what they are for AIX code. Also, PASE runtime includes more flexibility in its C language support compared to the full ANSI compliant support in ILE C. A significant time savings may be achieved in the porting phase by not having to recreate the application's build environment. The testing cycle for a PASE application may also be simpler.
- All applications, whether ported in ILE or PASE require a similar amount of customization to create an AS/400 product that meets AS/400 customers' expectations of full integration with database, save/restore, security, ease of installation, licensing and robust support services.

Positioning: Porting Options



★ The right tool for the job:

- Java focuses on cross-platform portability
- Domino focuses on cross-platform groupware enablement
- AS/400 ILE focuses on tuning to and exploiting OS/400 facilities
- AS/400 PASE focuses on speed of porting for selected UNIX applications

★ Common strengths:

- All can integrate with AS/400 operations and security
- All can integrate with AS/400 database and file systems

★ AS/400 competitive advantage:

- All run as native applications
- All can run on a single server or be separated in logical partitions
- *Offers complementary NT applications via Integrated Netfinity Server*

Positioning: Porting Options



- PASE adds another porting option for ISVs, extending the AS/400's ability to offer applications from a variety of sources, including Java, UNIX, NT and Domino. Each of the porting options has its own strengths.
- Java focuses on cross platform portability and is especially dominant in e-business and Websphere application ports.
- Domino focuses on groupware application portability and offers a full range of core business application integration options, such as Domino Enterprise Connection Services (DECS) for real-time database access and Lotus Enterprise Integrator for larger database transfers.
- AS/400 Integrated Language Environment is the mainstay of the majority of AS/400's core application base, with its highly optimized services for DB2/400 access and other AS/400-tuned facilities.
- Now PASE adds a further porting option for highly compute intensive UNIX applications that, with a good fit in function, can improve the time-to-market for UNIX application ports.

Positioning: Consolidated Server



★ Single server, multiple applications

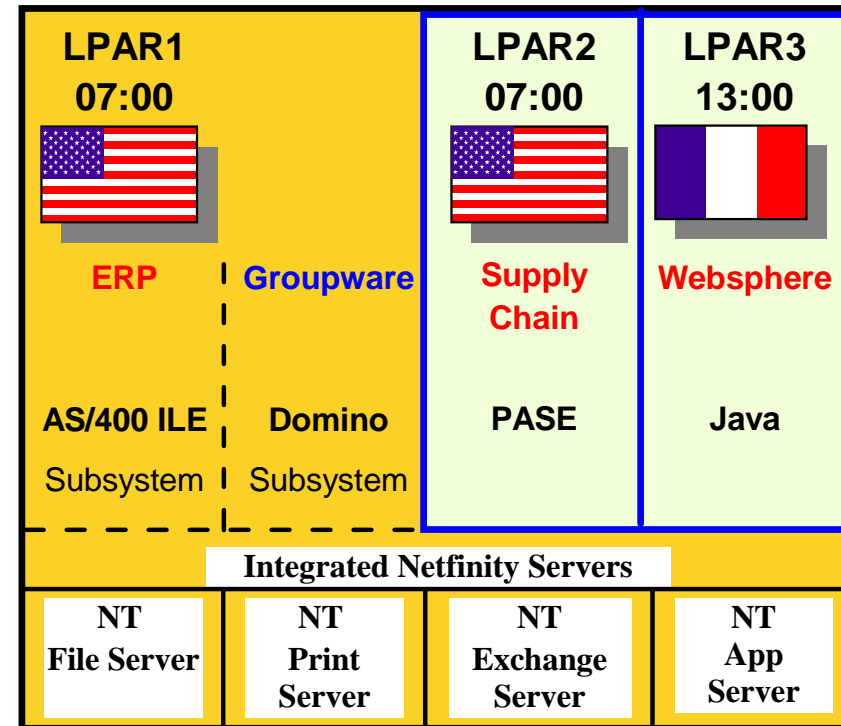
- Java
- PASE
- NT
- AS/400



★ Workload management

- Logical partitions
- Subsystems
- Integrated Netfinity Server

★ Incorporates RS/6000, S/390 and Netfinity technology



Positioning: Consolidated Server



- As companies rationalize their IS infrastructure, they require servers to have the flexibility to manage multiple applications, from a variety of platforms and operating systems. A consolidated server stands as an alternative to the single application per server model that is prevalent in today's UNIX and WinTel markets. In addition, a consolidated server must also offer robust workload management services and outstanding availability characteristics.
- Today, the AS/400 stands out as a leading example of a consolidated server: combining multiple application workloads, offering robust workload management services such as logical partitioning, and -- according to Gartner Group -- demonstrating the industry's leading single system availability track record.
- In this example, you can see how the AS/400 extends from its core strengths of ERP applications, to incorporate groupware, supply chain and e-commerce applications, all fully integrated on a single server. Applications can run independently in separate subsystems in a single instance of OS/400, or they can be divided into logical partitions (LPAR) to fully isolate applications with specific performance, scheduling or time zone requirements.
- The AS/400's ability to assimilate applications from other platforms has increased dramatically as a result of AS/400's investments in Domino and Java. But, grabbing attention recently have been two services that provide AS/400 runtime capability for NT and UNIX applications.
- The first is the Integrated Netfinity Server: a PC under the covers of the AS/400, logo'ed by Microsoft to run standard Windows NT. Today, over 20% of AS/400's shipped include an Integrated Netfinity Server to run complementary NT applications.
- Now, PASE (Portable Application Solutions Environment) provides a UNIX application runtime on the AS/400, enabling the rapid porting and deployment of applications, resulting in better integration of supply chain and ERP suites in a single, multi-function server. PASE exploits the PowerPC processor's ability to switch between AS and UNIX runtime modes, while retaining full integration with AS/400's file systems, security and database.
- IBM is uniquely positioned to provide a consolidated, flexible server. AS/400's success relies on access to IBM broad base of enterprise server technologies: logical partitions from the S/390, Intel server design leadership from Netfinity, UNIX runtime capability from RS/6000 and AIX.

Solutions Assessment



★ Business assessment

- Strong market opportunity and fits AS/400 industry solutions portfolio
- Initial targets are ISVs with existing UNIX applications and IBM relationship
- IBM PartnerWorld for Developers assessment shows good technical fit

★ Technical assessment factors

- Specific UNIX function dependencies, such as *fork*
- Infrastructure factors, build & test complexity
- Solution developer skills and application base
- Level of interaction with DB2 Universal Database for AS/400 or existing AS/400 ILE or Java applications
- Level of compute intensive or floating point operations

Solutions Assessment



- PartnerWorld for Developers, AS/400 (formerly Partners in Development) uses an automated process for assessing the ease of portability of ISV applications. A shell script is run against the application to assess its use of APIs -- systems services and database calls. The results are compared with AS/400 API support across a variety of programming models, including AS/400 ILE and PASE. Although it is hard to generalize and each application must be considered individually, many UNIX applications are suitable to be ported to the AS/400 via either ILE or PASE. The report resulting from the technical analysis is used to determine where the application has a good fit technically, and becomes part of a wider business assessment by IBM and the ISV.
- Some UNIX applications that favor PASE runtime are those that rely heavily on functions like fork without exec. This function is not supported directly in ILE and would normally be converted to use spawn. Other factors include complexity of the ISV's existing UNIX application build and test environment and available programming skills. A PASE implementation could in effect reuse the build environment on AIX if that environment was very complicated, and could provide an easier port in this aspect.
- For additional considerations, see the PartnerWorld for Developers porting website www.as400.ibm.com/developer/factory in the porting section.

Does AS/400 PASE Fit My Solution?



- Existing UNIX source code base, preferably AIX (will port to AIX if a good choice)
- PWD AS/400 porting team analysis shows good fit with supported APIs for a target release
- Dependence on a function or language environment that is not available on, or is incompatible with the traditional architecture of OS/400 (e.g., FORTRAN or Smalltalk direct code gen)
- Application requires CPU-intensive performance or smaller footprint (by minimizing pointer space)
- Substantially increased cost to port to ILE, AS/400 skill unavailable, or constrained time to market
- Code base is such that required changes fall into category of traditional UNIX-to-UNIX port modifications for the ISV

Does AS/400 PASE Fit My Solution?



Existing UNIX source code base, preferably AIX (will port to AIX if a good choice)

PWD AS/400 porting team analysis shows good fit with supported APIs for a target release

Dependence on a function or language environment that is not available on, or is incompatible with the traditional architecture of OS/400 (e.g., FORTRAN or Smalltalk direct code gen)

Application requires CPU-intensive performance or smaller footprint (by minimizing pointer space)

Substantially increased cost to port to ILE, AS/400 skill unavailable, or constrained time to market

Code base is such that required changes fall into category of traditional UNIX-to-UNIX port modifications for the ISV

Targeted Applications



- ★ Existing UNIX source code base, preferably AIX (will port to AIX if successful)
- ★ Analysis with Porting Team sifter front-end and PASE analysis shows good fit with supported APIs
- ★ Dependence on a function or language environment that is not available on, or is incompatible with the architecture of OS/400 (e.g., FORTRAN or Smalltalk direct code gen)
- ★ Concerns in cost of porting to ILE or AS/400 skill unavailable or time to market
- ★ Code base is such that required changes fall into category of traditional UNIX-to-UNIX port modifications for the vendor
- ★ Where performance adequately matches target customer expectations (current AS/400 business partners comparing to older OS/400 version or new partners comparing to AIX version)

Targeted Applications



Existing UNIX source code base, preferably AIX (will port to AIX if successful)

Analysis with Porting Team sifter front-end and PASE analysis shows good fit with supported APIs

Dependence on a function or language environment that is not available on, or is incompatible with the architecture of OS/400 (e.g., FORTRAN or Smalltalk direct code gen)

Concerns in cost of porting to ILE or AS/400 skill unavailable or time to market

Code base is such that required changes fall into category of traditional UNIX-to-UNIX port modifications for the vendor

Where performance adequately matches target customer expectations (current AS/400 business partners comparing to older OS/400 version or new partners comparing to AIX version)

Applications Not Targeted



- ★ **Requires real-time execution profile (real-time not an AS/400 target market)**
- ★ **Reliance on kernel extensions or device drivers for specialized I/O devices (SLIC kernel is not extendible)**
- ★ **Extensive use of database with very fine-grained access, where ASCII to EBCDIC conversions or static SQL use substantially affect performance, or where there is a database function mismatch with an expected UNIX database**

Applications Not Targeted

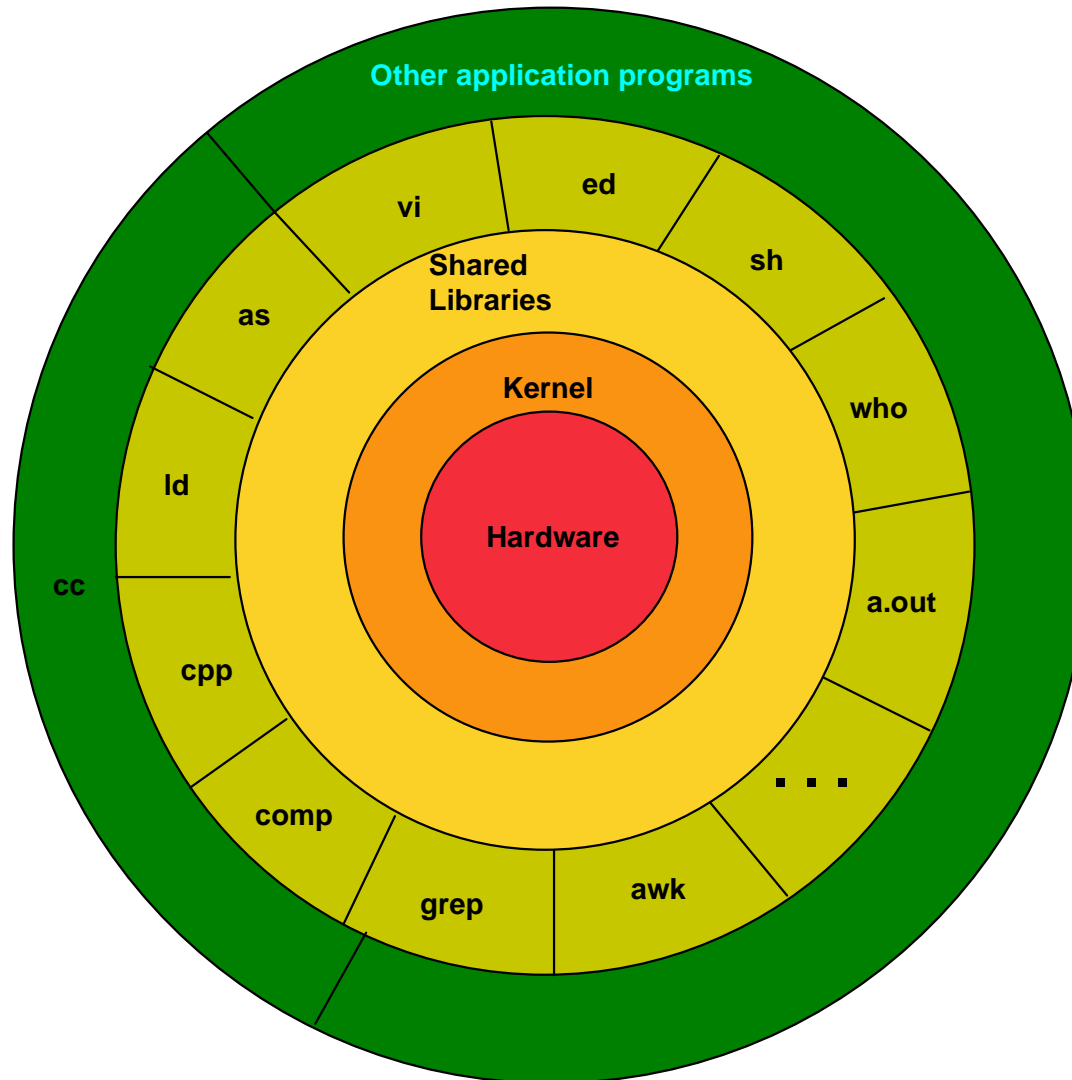


Requires real-time execution profile (real-time not an AS/400 target market)

Reliance on kernel extensions or device drivers for specialized I/O devices (SLIC kernel is not extendible)

Extensive use of database with very fine-grained access, where ASCII to EBCDIC conversions or static SQL use substantially affect performance, or where there is a database function mismatch with an expected UNIX database

UNIX Architecture

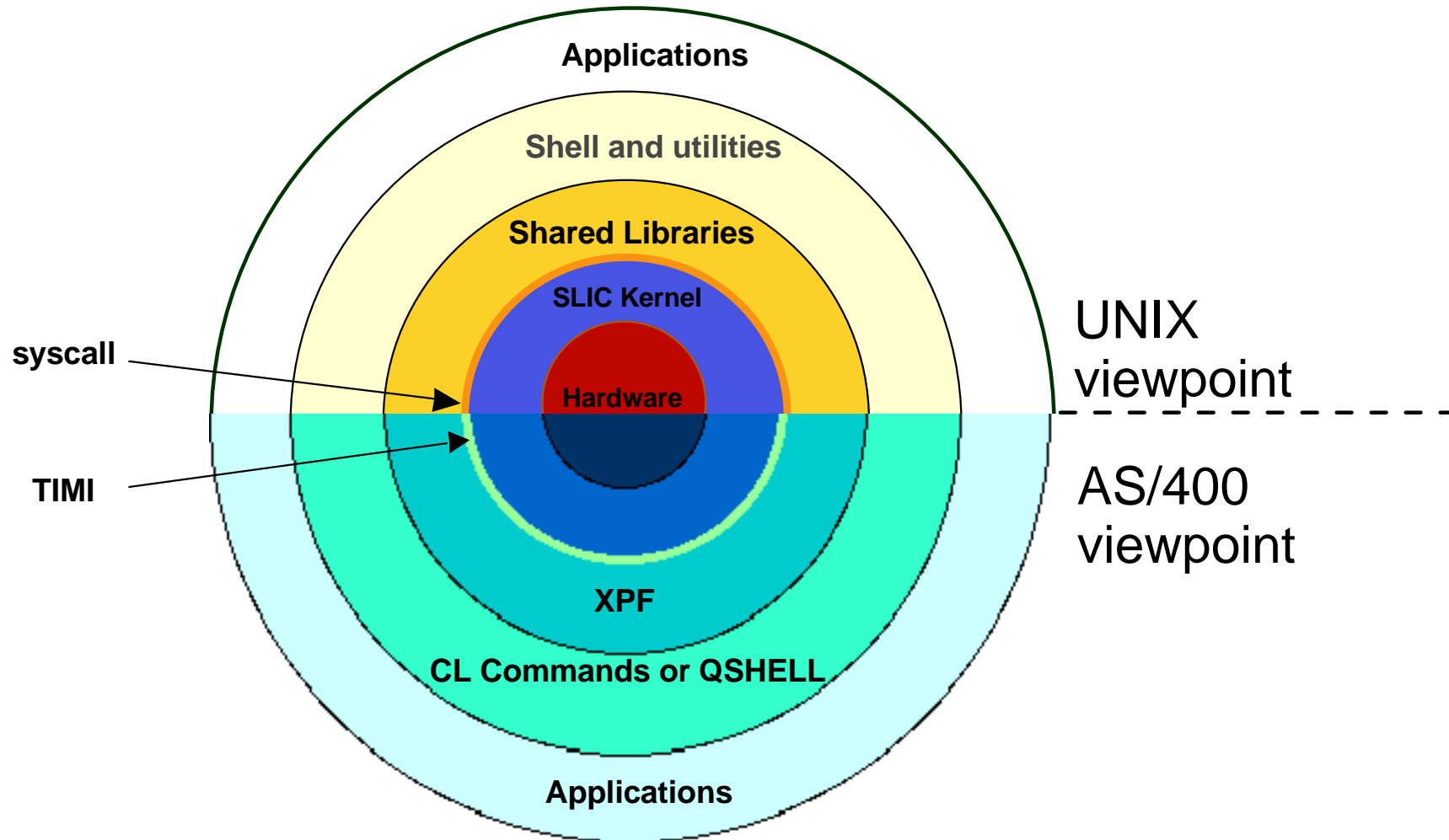


UNIX Architecture



Unix architecture can generally be portrayed as a series of concentric circles. The innermost circle is the hardware being supported. Outside that is the operating system kernel. Next are the libraries directly supported by the UNIX operating system. Next are the shells and utilities provided by the operating system. After that come libraries and other middleware provided by the application developers. And on the outermost circle are the Unix applications.

PASE in UNIX Terms



PASE in UNIX Terms

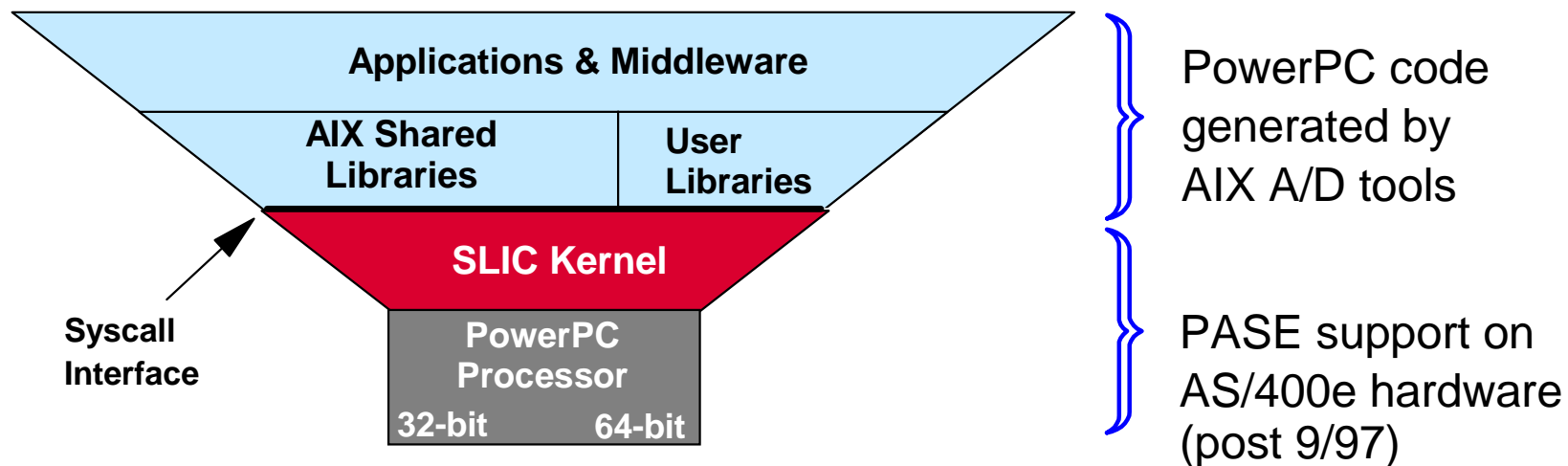


PASE is an environment on the AS/400 that is designed to behave like AIX to the applications. But instead of mapping most of the services directly to the hardware, they are mapped into OS/400 services. At runtime, the system operator can view PASE from the lower half of the circle; because all of the services are mapped into OS/400 services, the applications can be managed just like any other AS/400 applications.

AS/400 PASE Architecture



- ★ PASE enables UNIX PowerPC code (generated by AIX A/D tools) to be run by the SLIC kernel above the AS/400 syscall interface (not an emulator)
- ★ PASE exploits the processor's ability to dynamically switch between AS/400 execution modes (64-bit PowerPC AS, 64-bit PowerPC, 32-bit PowerPC). PASE is 32-bit only at this time.

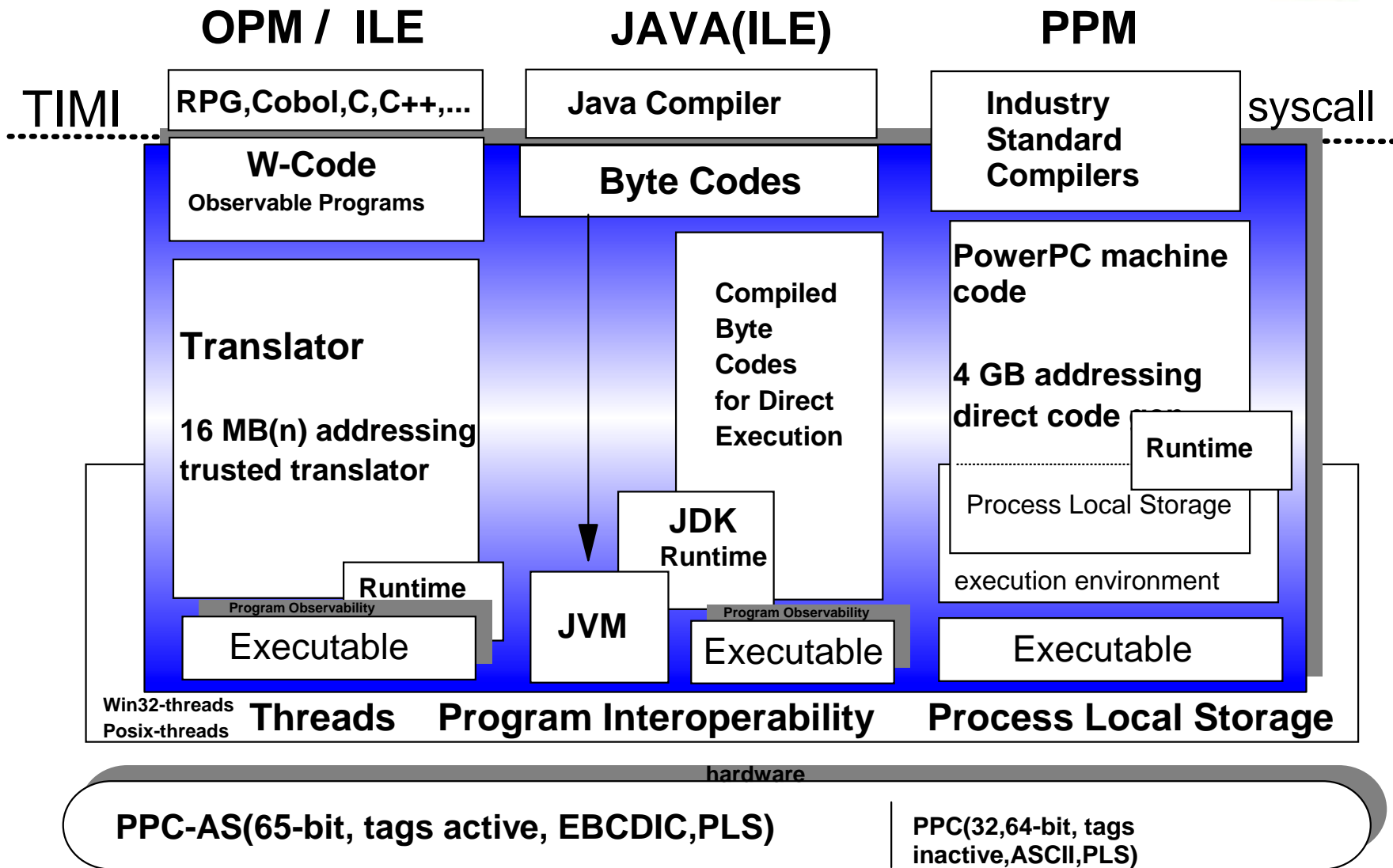


AS/400 PASE Architecture



Another way to think about this is as if the PASE environment is a funnel, mapping all of the service requests into the system for the application. PASE relies on the hardware to provide full support for the 32-bit instructions used in AIX application binaries.

AS/400 PASE and Other Program Models

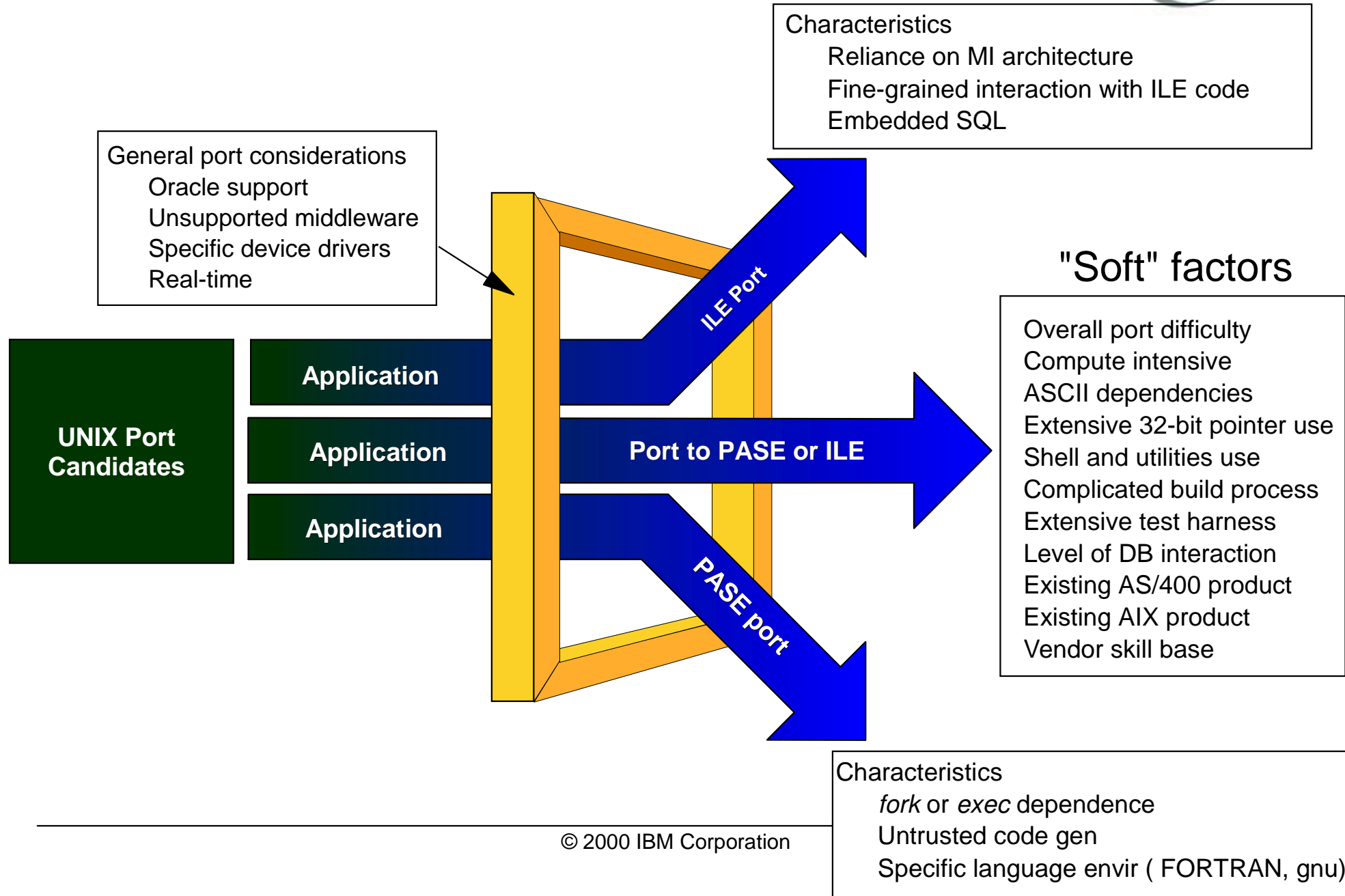


AS/400 PASE and Other Program Models



This page shows all of the program models now available for applications running on OS/400 V4R5. Traditional applications can use either OPM (Old Program Model) or ILE (Integrated Language Environment). Both of those models run an instruction set that is mapped into TIMI (the Machine Interface layer that shields the application from the hardware). Java applications run under the context of a JVM (Java Virtual Machine) that behaves as if it was running Java byte codes in the applications. On the AS/400, however, Java objects are compiled into executable objects that also use TIMI services. PASE is different; it runs AIX binaries, and those objects use the Syscall interface to access operating system services.

UNIX Application Porting Factors



© 2000 IBM Corporation

UNIX Application Porting Factors



When you decide to port an application to the AS/400 from UNIX, there are basically two options now; you can port to ILE or to PASE. Some application designs will not move easily to ILE; if there is a high dependence on fork or exec, or if the code was designed to leverage the AIX hardware or uses untrusted code gen, a port to PASE will be much more practical than a port to ILE. Applications that do a lot of database manipulation, on the other hand, are probably best off being ported to ILE, because all of the database calls have to be intercepted and routed to OS/400 services. Additionally, PASE will probably trigger many more data conversions than database access in an ILE program. There are many other considerations. That is why the first step is submitting your code for an API analysis, and finding out what the IBM porting team would recommend.

AS/400 PASE Code Development



- ★ Binaries copied to the AS/400 and placed in a stream file in the Integrated File System (IFS)



- ★ PowerPC code development done on an RS/6000 workstation using AIX 4.3 A/D tools (editors, compilers, linkers, debuggers)

- ★ PASE code (binaries) is called by an application with the Qp2RunPase API (only AS/400e hardware)

AS/400 PASE Code Development



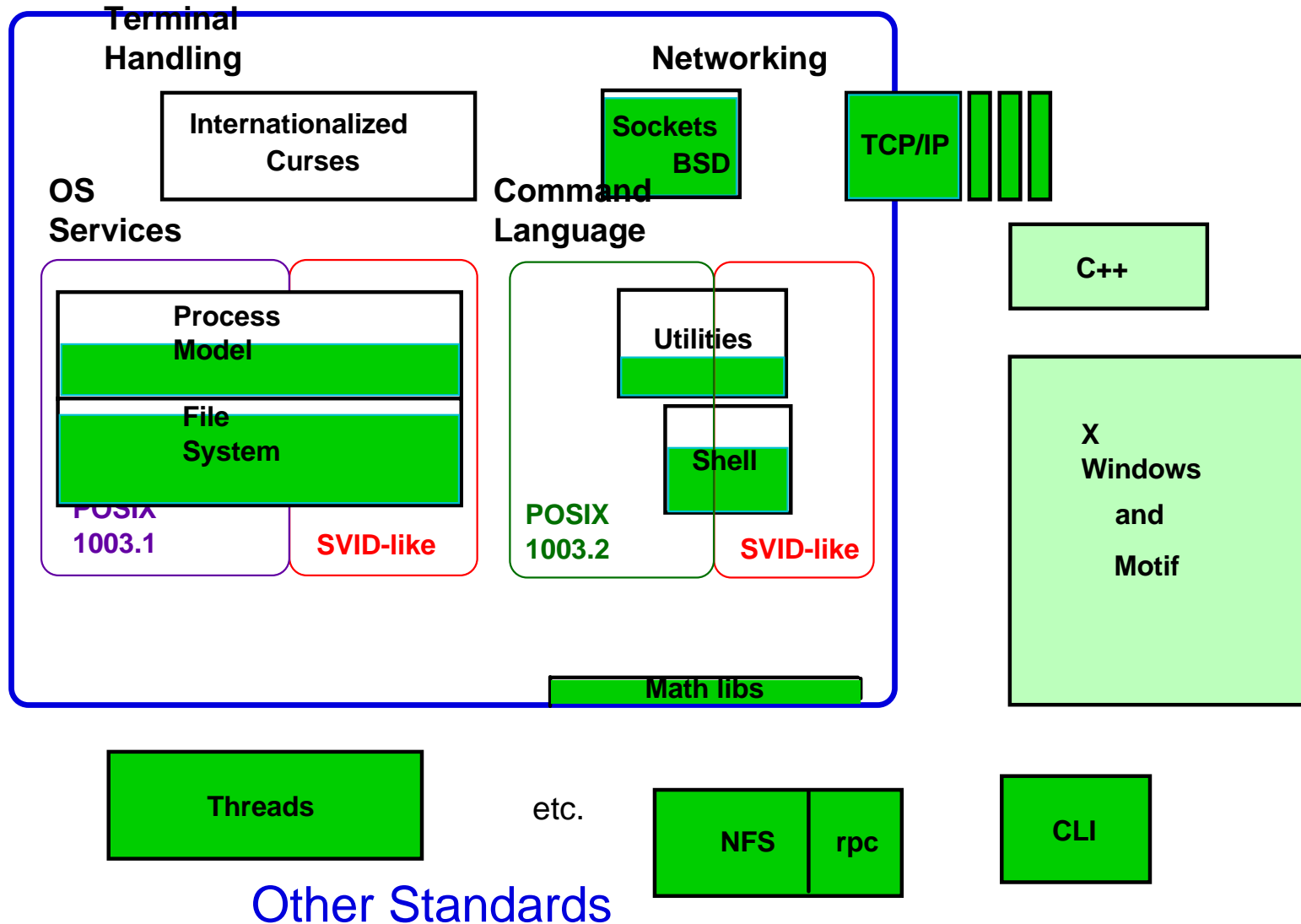
PASE code development ends up being a 3 step process, involving both an AIX workstation and the AS/400:

1. PowerPC code development done on an RS/6000 workstation using AIX 4.3 A/D tools (editors, compilers, linkers, debuggers)
2. Binaries copied to the AS/400 and placed in a stream file in the Integrated File System (IFS)
3. PASE code (binaries) is called by an application with the Qp2RunPase API (only AS/400e hardware)

UNIX Interface Support in ILE V4R4



Single UNIX Specification



UNIX Interface Support in ILE V4R4

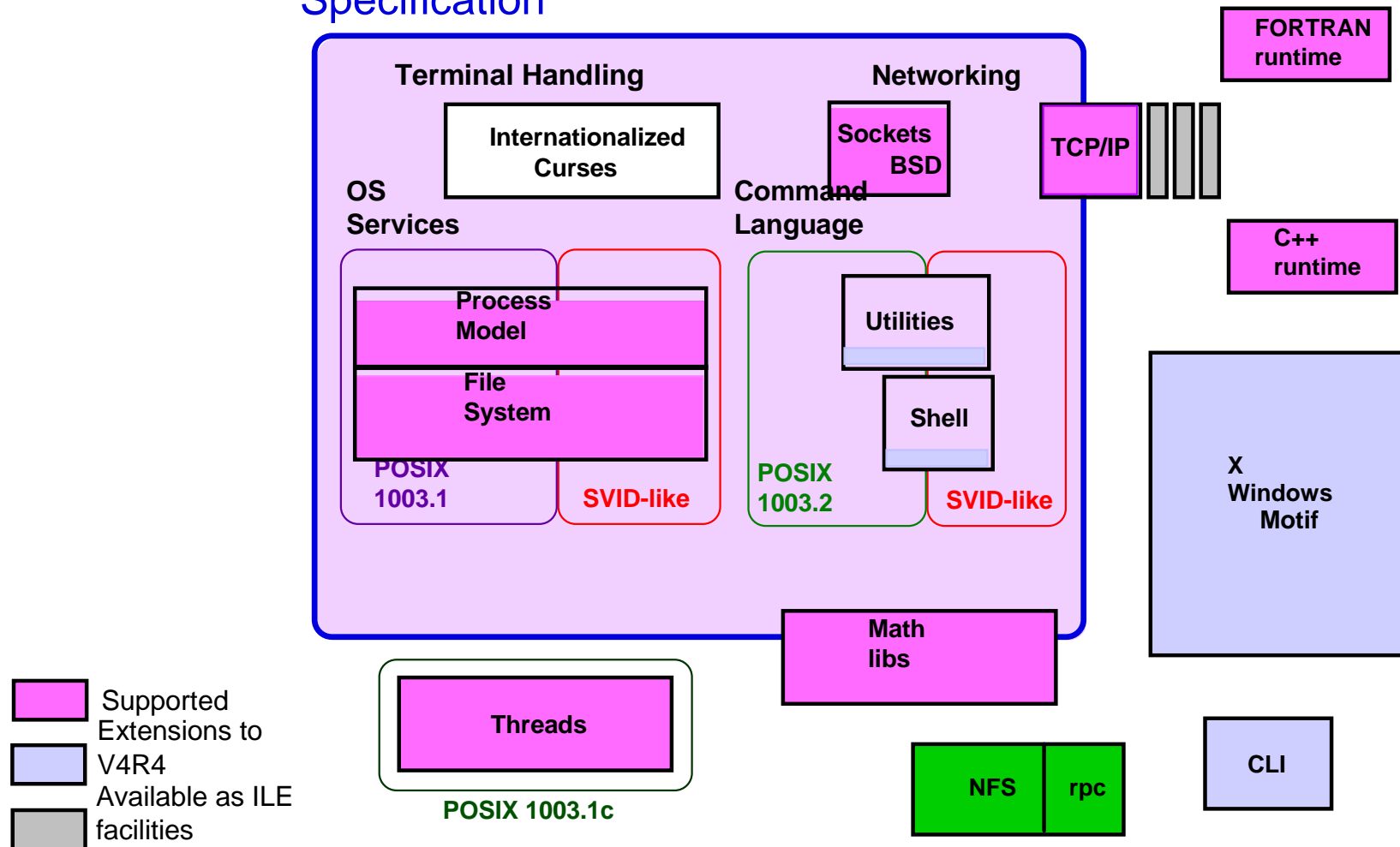


- This chart is intended to give an overall feel for good coverage for many important UNIX functions within OS/400 itself. They are fully integrated with other system function and are not in "another environment."
- The darker green fill within the previously defined boxes show approximate percentage of the coverage of the function by the OS/400 ILE code. Over the past five or so years, OS/400 has been enhanced dramatically to support UNIX-style functions. Great care has been taken to provide the correct semantics from the standards within the OS/400 environment and to make the functions interoperate with other pre-existing functions where meaningful.
- The lighter green fill indicates the availability of that function via a PRPQ or business partner offering. The background color is absent, indicating that these functions run in an EBCDIC environment with OS/400 16-byte pointers. Security and other system services files are implemented differently in OS/400 from the /etc files typically found in UNIX systems; so that APIs like getpwuid (to get the user information associated with a uid) may work, but APIs that would have read the /etc/passwd file like getpwent are not supported.

Standard UNIX Interface Support in PASE



Single UNIX Specification



Standard UNIX Interface Support in PASE



This chart is intended to give an overall feel for how much additional coverage is provided by the PASE environment.

The only functions NOT supported at all by PASE are the Internationalized Curses functions. An application that is heavily dependent on these will require a fair amount of redesign.

Note that database functions are not generally viewed as a basic part of a UNIX operating system, so the level of effort required to convert any database calls to AS/400 CLI aren't shown in this picture.

AS/400 PASE Structure

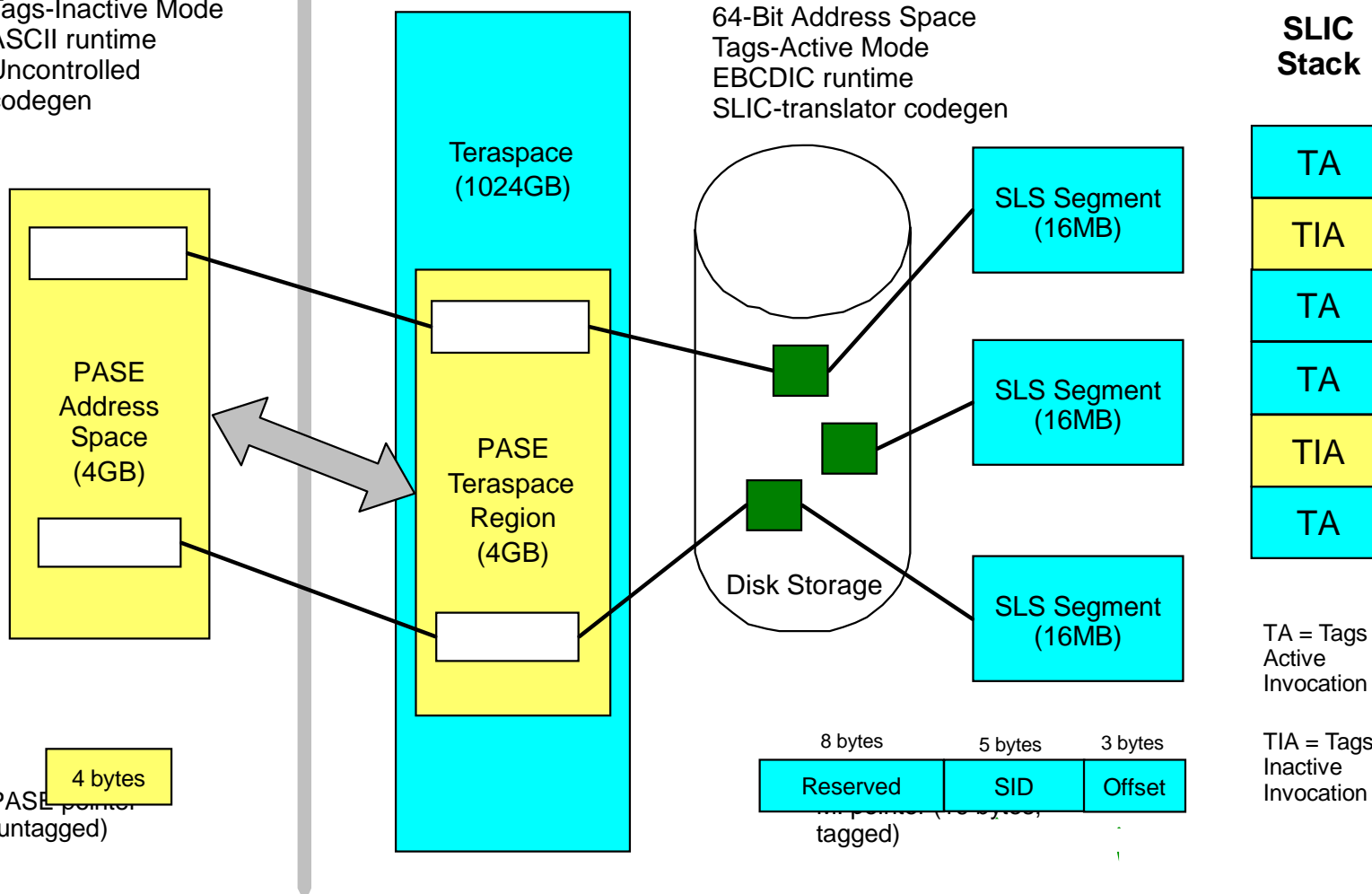


32-Bit Address Space
PASE
 Tags-Inactive Mode
 ASCII runtime
 Uncontrolled codegen

ILE / OPM / SLIC

64-Bit Address Space
 Tags-Active Mode
 EBCDIC runtime
 SLIC-translator codegen

SLIC Stack



AS/400 PASE Structure



- This chart shows the relationship between AS/400 PASE and the SLS environment. The memory for selected SLS segments is shared by mapping it into the private address space. The SLIC-managed call stack contains frames for tags-active invocations. There is a separate call stack for tags-inactive invocations. AS/400 PASE programs deal with 4-byte (untagged) pointers, in contrast to the 16-byte tagged MI pointers used in the SLS environment.

AS/400 PASE API Support



★ Language runtime environments supported:

- C
- C++
- Fortran

★ Common APIs/functions supported by AS/400 PASE

- Process and threads models including fork(), exec and pthreads families of APIs
- Same stream file system APIs as ILE (designed to meet UNIX95 or Single UNIX Spec stds)
- Sockets, TCP/IP also use existing OS/400 services
- X Window System libraries (X11R5 in v4r4) and Motif

★ API Analysis of application can be requested at:

- ibm.com/as400/developer/porting/apitool.htm

AS/400 PASE API Support



Language runtime environments supported:

- C
- C++
- Fortran

Common APIs/functions supported by AS/400 PASE

- Process and threads models including fork(), exec and pthreads families of APIs
- Same stream file system APIs as ILE (designed to meet UNIX95 or Single UNIX Spec stds)
- Sockets, TCP/IP also use existing OS/400 services
- X Window System libraries (X11R5 in v4r4) and Motif

API Analysis of application can be requested at:

- ibm.com/as400/developer/porting/apitool.htm

Launching PASE



- ★ Invocation from a 5250 terminal screen, Qp2Shell
- ★ Invocation from CL, Qp2Shell
- ★ Invocation from the AS/400 PASE terminal, Qp2Term
- ★ Invocation from CL, Qp2Term
- ★ Invocation from an ILE application, Qp2RunPase
- ★ Invocation from an ILE application, Qp2CallPase

Launching PASE



Invocation from a 5250 terminal screen, Qp2Shell

- An application can be started in the AS/400 PASE environment by using the CL command QP2SHELL and passing as the parameter the command to launch.

Invocation from CL, Qp2Shell

- Program QP2SHELL runs a Portable Application Solutions Environment (PASE) program in the job where it is invoked. The first argument for QP2SHELL is the path name for a PASE program. Additional arguments are optional. Copies of all argument strings are passed to the PASE program (following the C language convention that the first argument is the name of the program itself). QP2SHELL initializes PASE environment variables with a modified copy of the entire ILE environment. A PASE environment variable is initialized for every ILE environment variable, but the initial value of any PASE variable (except those with a name beginning "PASE_") is overridden by the value of an environment variable with a name that concatenates the prefix PASE_ with the original variable name.

Invocation from the AS/400 PASE terminal, Qp2Term

- The default shell prompt will be written to the screen once the terminal session is ready. A full set of utilities can be invoked with standard output and standard error written and scrolled in the terminal screen. Applications can also be started.

Invocation from CL, Qp2Term

- Program QP2TERM runs an "interactive terminal session" that submits a batch job to run a Portable Application Solutions (PASE) program in a batch job, and uses the workstation display in the interactive job to present output and accept input for files stdin, stdout, and stderr in the batch job. Any argument strings for QP2TERM are passed to program QP2SHELL in the batch job. The arguments specify a path name and argument data for a PASE program that runs in the batch job.

Invocation from an ILE application, Qp2RunPase

- The Qp2RunPase() function runs a Private Address Space Environment (PASE) program in the job where the API is invoked. It loads a PASE program (including any necessary shared libraries) and then transfers control to the program. Qp2RunPase works like the AIX execve function, including the ability to run shell scripts and the rules for resolving shared libraries (sometimes using the LIBPATH environment variable). Qp2RunPase does not overlay or destroy the state of any ILE code currently active in the process, so control returns to the caller of Qp2RunPase when the PASE program runs the exit function.

Invocation from an ILE application, Qp2CallPase

- The Qp2CallPase() function calls a procedure in a Private Address Space Environment (PASE) program in the job where the API is invoked. Qp2CallPase is useful to allow an ILE procedure that was invoked by a PASE program to "call back" into the PASE program.

Environment Variables



- ★ If you use the "launchers", ILE environment variables are copied into AS/400 PASE
- ★ If you use the QP2RunPase() API they are not implicitly copied
- ★ Special PASE_XXX environment variables are set as XXX in PASE
 - these override the same named variable copied from ILE
 - this allows two environments without collisions in ILE
- ★ Changes to environment variables while running in AS/400 PASE are NOT reflected back in ILE

Environment Variables



If you use the "launchers", ILE environment variables are copied into AS/400 PASE

If you use the QP2RunPase() API they are not implicitly copied

Special PASE_XXX environment variables are set as XXX in PASE

- these override the same named variable copied from ILE
- this allows two environments without collisions in ILE

Changes to environment variables while running in AS/400 PASE are NOT reflected back in ILE

Database and File Access



- ★ **Applications have access to integrated file system (IFS) files through C, C++ and Fortran APIs**

- ★ **Applications have access to DB2 UDB for AS/400 through the same CLI interfaces used in the ILE environment**
 - **AS/400 PASE CLI interfaces thinly wrapper the OS/400 native CLI interfaces.**
 - **To use AS/400 specific CLI interfaces, recompile using the OS/400 header file.**
 - **ASCII/EBCDIC conversions are performed.**

Database and File Access



Applications have access to integrated file system (IFS) files through C, C++ and Fortran APIs

Applications have access to DB2 UDB for AS/400 through the same CLI interfaces used in the ILE environment

- AS/400 PASE CLI interfaces thinly wrapper the OS/400 native CLI interfaces. There are minimal differences from UDB for RS/6000. Some AIX applications may need to port to the OS/400 CLI interfaces. For details see "DB2 UDB for AS/400 SQL Call Level Interface (ODBC)" at: (<http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/BOOKS/QB3AVG02/CCONTENTS>)
- To use AS/400 specific CLI interfaces, recompile using the OS/400 header file in library QSYSINC, file H, member SQLCLI (e.g. using an NFS mount). Any RS/6000 CLI header file(s) need to be removed from the application's source to avoid conflicts at compile time.
- ASCII/EBCDIC conversions are performed to AS/400 PASE CCSID, providing the application with an ASCII view of the data. The CCSID can be overridden using `SQLOverrideCCSID400()`.

PASE Headers and Exports



★ **If your application needs AS/400 specific functions, use:**

- `as400_libc.exp` and `as400_libc_r.exp` :
 - AS/400 extensions to AIX `libc.a` and `libc_r.a`
 - e.g. exports functions needed to call ILE procedures
- `as400_protos.h` : contains prototypes for the AS/400 unique functions (e.g. for calling ILE)
- `as400_types.h` : contains types such as `ILEpointer` and ILE argument lists
- `qp2user.h` : contains protoypes for `QP2RunPase()` and `QP2CallPase()` APIs
- `libdb400.exp` : contains DB2 UDB for AS/400 CLI support

PASE Headers and Exports



If your application needs AS/400 specific functions, use:

- `as400_libc.exp` and `as400_libc_r.exp` :
- AS/400 extensions to AIX `libc.a` and `libc_r.a`
- e.g. exports functions needed to call ILE procedures
- `as400_protos.h` : contains prototypes for the AS/400 unique functions (e.g. for calling ILE)
- `as400_types.h` : contains types such as ILEpointer and ILE argument lists
- `qp2user.h` : contains protoypes for QP2RunPase() and QP2CallPase() APIs
- `libdb400.exp` : contains DB2 UDB for AS/400 CLI support

These are available by sending email to RCHGO400@us.ibm.com

Building on AIX



- ★ **Compile on 4.33**
- ★ **If your program needs to use AS/400-unique functions in AS/400 PASE (such as `_ILECALL` to call an ILE procedure, or using type `ILEpointer` to store a tagged MI pointer), you need additional header files**
- ★ **If you use `ILEpointer` type use both the following compile options to properly align ILE pointers**
 - `-qlngdbl128`
 - `-qalign=natural`
- ★ **Use "site namefmt 1" to transfer your files into the integrated file system (IFS)**

Building on AIX



Compile on 4.33

If your program needs to use AS/400-unique functions in AS/400 PASE (such as `_ILECALL` to call an ILE procedure, or using type `ILEpointer` to store a tagged MI pointer), you need additional header files

If you use `ILEpointer` type use both the following compile options to properly align ILE pointers

- `-qlngdbl128`
- `-qalign=natural`

Use "site namefmt 1" to transfer your files into the integrated file system (IFS)

Access to other OS/400 Program Models



★ AS/400 PASE programs can call ILE programs

- automatically using system services (e.g. fopen() API)
- **systemCL() API** - run a CL command, including CALL to a *PGM object
- **_ILECALL() API** - takes a procedure pointer to any ILE procedure

★ ILE programs can:

- **call back to the AS/400 PASE program** using the QP2CallPase() API
- simply return to the calling AS/400 PASE program

Access to other OS/400 Program Models



AS/400 PASE programs can call ILE programs

- automatically using system services (e.g. fopen() API)
- systemCL() API - run a CL command, including CALL to a *PGM object
- _ILECALL() API - takes a procedure pointer to any ILE procedure

ILE programs can:

- call back to the AS/400 PASE program using the QP2CallPase() API
- simply return to the calling AS/400 PASE program

Accessing ILE, continued



★ Calling ILE programs from AS/400 PASE requires some forethought:

- Providing a correct procedure pointer type
 - recommended way is `_ILELOAD()` and `_ILESYM()` APIs
- Correct compilation and declaration for boundary aligned arguments
 - `size_ILEarglist()` and `build_ILEarglist()` can help
 - storage malloc'ed in AS/400 PASE is aligned
 - static and automatic storage is NOT aligned, so you must code to align it
 - aggregates (structures and unions) require planning and management
- Providing correct pointer types for arguments
 - use `ILEpointer` type in `as400_types.h`
- ASCII/EBCDIC considerations when passing or sharing data
 - conversion is NOT performed for you (except for `stdin/stdout/stderr`)
- ALL ILE programs or service programs called from AS/400 PASE must be compiled `TERASPACE(*YES)`

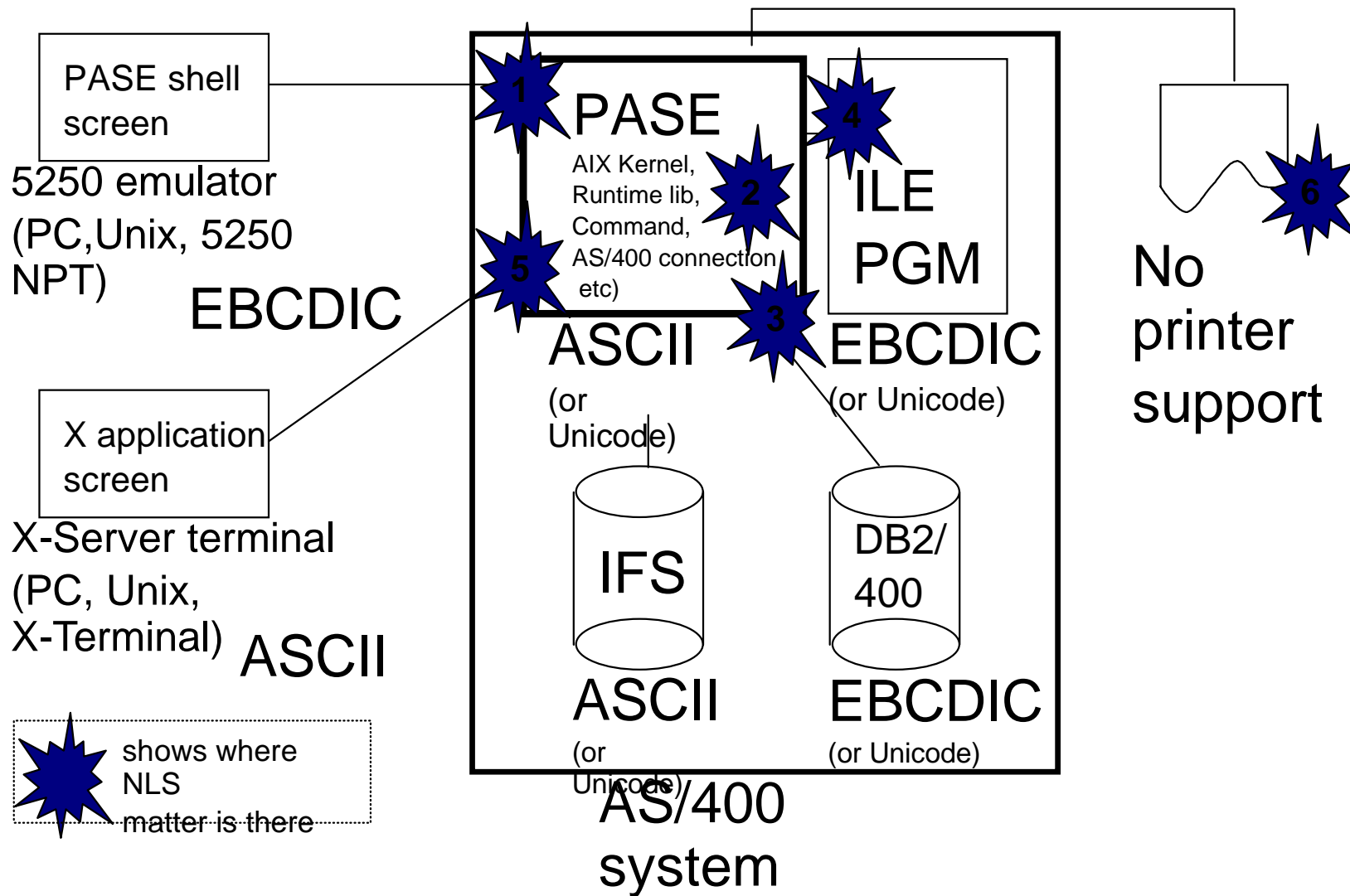
Accessing ILE, continued



Calling ILE programs from AS/400 PASE requires some forethought:

- Providing a correct procedure pointer type
 - recommended way is `_ILELOAD()` and `_ILESYM()` APIs
- Correct compilation and declaration for boundary aligned arguments
 - `size_ILEarglist()` and `build_ILEarglist()` can help
 - storage malloc'ed in AS/400 PASE is aligned
 - static and automatic storage is NOT aligned, so you must code to align it
 - aggregates (structures and unions) require planning and management
- Providing correct pointer types for arguments
 - use `ILEpointer` type in `as400_types.h`
- ASCII/EBCDIC considerations when passing or sharing data
 - conversion is NOT performed for you (except for `stdin/stdout/stderr`)
- ALL ILE programs or service programs called from AS/400 PASE must be compiled `TERASPACE(*YES)`

ASCII/EBCDIC Conversions



ASCII/EBCDIC Conversions



It is important to note when ASCII is being used in the system, and when EBCDIC is being used; whenever data is transferred from one part of the system to another, data conversions are required. In some cases, these conversions are done automatically, as long as the data in question is properly tagged. In other cases, the application must handle the conversions. Following are key areas to understand within the PASE environment:

- AS/400 PASE runs in CCSID 819 unless otherwise specified on an environment variable on the QP2RunPase() API
- AS/400 PASE does EBCDIC->ASCII conversion on all argument strings passed to the AS/400 PASE main()
- AS/400 PASE opens all files in binary, so no text translation is done.
 - Use iconv() routines (in AS/400 PASE or in ILE) if you need data conversion
- The stdin/stdout/stderr are treated specially and are translated between ASCII and EBCDIC to present AS/400 PASE programs with expected ASCII data.
- Other items that AS/400 PASE does automatic ASCII / EBCDIC conversions on include: file meta-data (path names), security names (user/group names) and several UNIX standard APIs which return text strings (e.g. gethostname).
- PASE does not automatically do conversions of data on arguments passed to ILE procedures (_ILECALL use) or on callback from ILE to PASE (Qp2CallPase).

AS/400 PASE Security



- ★ **AS/400 PASE uses its own memory space**, enforced at the hardware level
- ★ **Both the Java and PASE program models directly put executables on the AS/400.** However, they are secured the same way as the executables created on the system itself. If you run a program, the owner can only do what his privileges allow, or, by what he can adopt if adoption of authority is allowed. **Your system continues to be secured via user profile privileges and correct use of adoption.**
- ★ **As always, it is important to control discretionary access control overrides** (the ability to give someone else the ability to give access) **to prevent problems** (such as the introduction of viruses).

AS/400 PASE Security



From a security point of view, PASE programs are subject to the same security restrictions as any other programs on AS/400. To run a PASE program on AS/400, you must have authority to the AIX binary in the integrated file system. You must also have the proper level of authority to each of the resources accessed by that program, or the program will receive an error when you attempt to access those resources.

AS/400 PASE uses its own memory space, enforced at the hardware level

Both the Java and PASE program models directly put executables on the AS/400. However, they are secured the same way as the executables created on the system itself. If you run a program, the owner can only do what his privileges allow, or, by what he can adopt if adoption of authority is allowed. Your system continues to be secured via user profile privileges and correct use of adoption. As always, it is important to control discretionary access control overrides (the ability to give someone else the ability to give access) to prevent problems (such as the introduction of viruses).

AS/400 PASE Performance



- ★ **As with all performance sizings, predicting the performance of an application running in AS/400 PASE depends on a number of variables and will vary by application.**
- ★ **Application-specific analysis and tuning should always be an integral part of porting a product to AS/400 PASE.**
- ★ **Particular attention should be paid to:**
 - Application Structure
 - Use of system services
 - Database dependencies

AS/400 PASE Performance



As with all performance sizings, predicting the performance of an application running in AS/400 Portable Application Solutions Environment (AS/400 PASE) depends on a number of variables and will vary by application.

Therefore, application-specific analysis and tuning should always be an integral part of porting a product to AS/400 PASE.

Of particular interest are:

- **Application Structure:** The computational/branching portion of the application is expected to perform similarly to how it would on an identically configured RS/6000 server. (Note that this also describes the execution of much of the AS/400 PASE runtime code, which can be considered 'computational/branching.')
- **Use of system services:** AS/400 PASE uses OS/400 IFS, Security, TCP/IP, Sockets, Process and Thread support, etc. The performance of the portions of the application that access these services is expected to be similar to applications that access the services using ILE interfaces. There are some specific areas where an AIX application may perform significantly differently in PASE than it does on equivalent RS/6000 hardware, because the system services that are invoked use differing amounts of system resource. A specific example would be an application that uses the fork interface extensively for hundreds of very small jobs; when these forked jobs are mapped into AS/400 program processes, they will use substantially more resource than they did under AIX on the RS/6000.
- **Database dependencies:** UNIX applications will need to be retargeted to DB2 UDB for AS/400. Direct interfaces to the database CLI are available within AS/400 PASE. Performance will depend on the types of interfaces used, data access patterns, database structure, etc. EBCDIC to ASCII conversion takes place when accessing DB2 UDB for AS/400, which may add a small overhead to the calls.

AS/400 PASE Operational View



AS/400 Work Management

- PASE code runs within a job, called from OS/400
- Can call other OS/400 system facilities and program models from PASE code
- Normal Work Management facilities
- Some problem determination and performance monitoring capabilities available from OS/400 interfaces

AS/400 Integrated File System

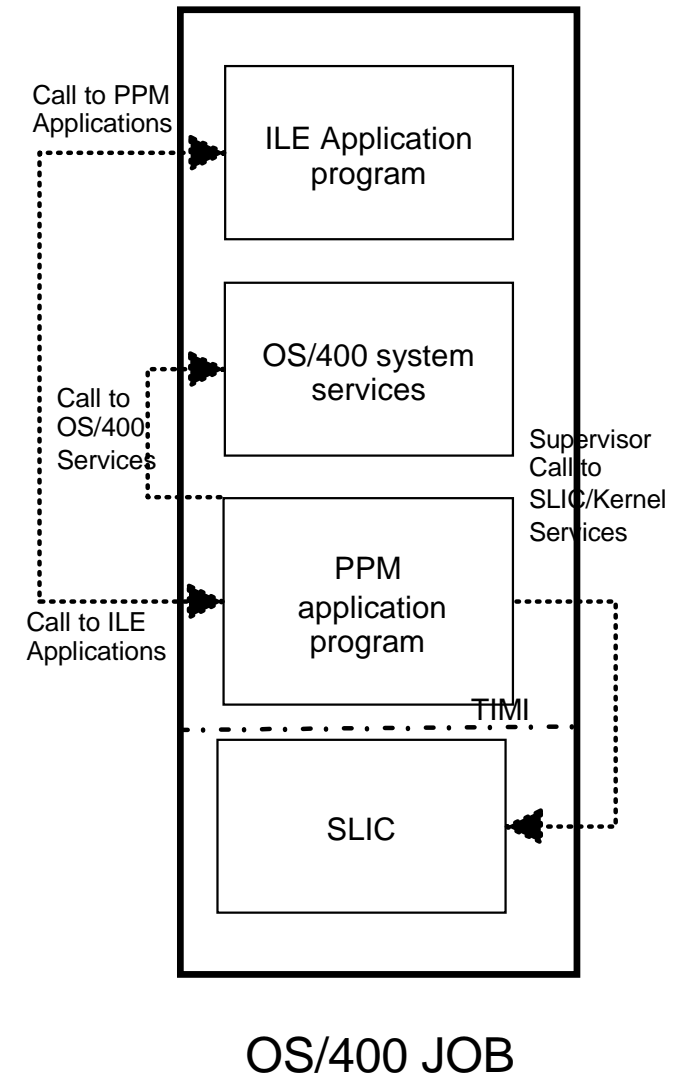
- Data stored in stream files
- Code stored in stream files
- Normal Storage Management facilities
- Save/restore capabilities inherent in IFS
- Normal OS/400 install procedures into IFS
- Normal PTF installation procedures into IFS

Security

- Fully integrated with the rest of the system (user profiles, authentication,...)

Database Access

- Database is callable using CLI, requires EBCDIC accommodation like ODBC or Java Virtual Machine



AS/400 PASE Operational View



AS/400 Work Management

- PASE code runs within a job, called from OS/400
- Can call other OS/400 system facilities and program models from PASE code
- Normal Work Management facilities
- Some problem determination and performance monitoring capabilities available from OS/400 interfaces

AS/400 Integrated File System

- Data stored in stream files
- Code stored in stream files
- Normal Storage Management facilities
- Save/restore capabilities inherent in IFS
- Normal OS/400 install procedures into IFS
- Normal PTF installation procedures into IFS

Security

- Fully integrated with the rest of the system (user profiles, authentication,...)

Database Access

- Database is callable using CLI, requires EBCDIC accommodation like ODBC or Java Virtual Machine

Example Ports



GNU Zip

- Utility needed for most open source

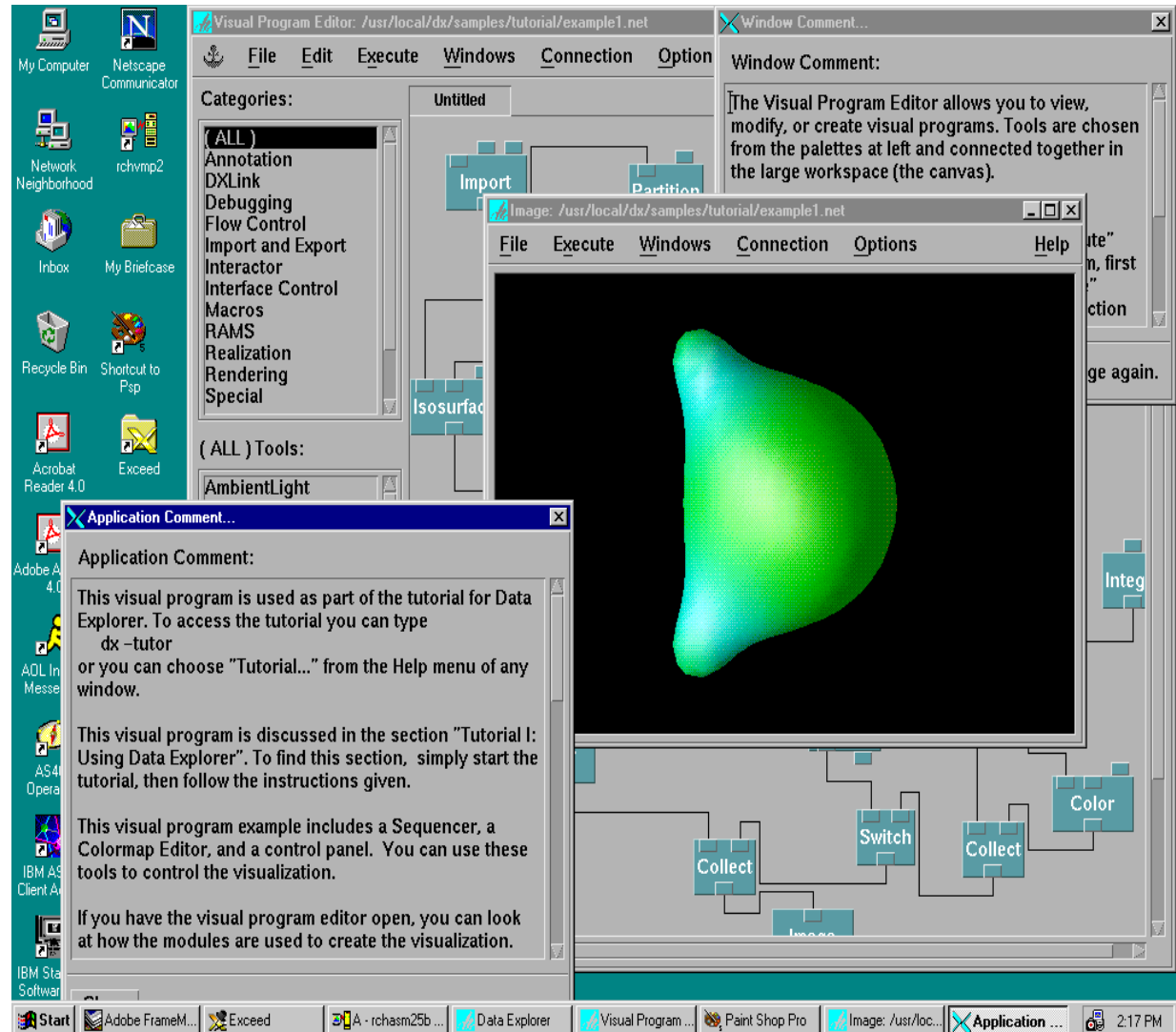
OpenDX

- Open Source Project based on IBM's Visualization Data Explorer, known as OpenDX

GNU Perl

- Popular and powerful scripting tool

ILE Call example



Example Ports



The AS/400 PASE redbook team enlisted the following criteria in choosing an application to port:

- Source and/or binaries freely available for download from the Internet
- Showcases the support of the X Window library support in AS/400 PASE
- Identify some of the more simple issues that may be encountered

Searching the World Wide Web, the team found the Open Source Project based on IBM's Visualization Data Explorer, known as OpenDX located at <http://www.opendx.org>. OpenDX is a software package for the visualization of scientific, engineering and analytical data. As it turns out, OpenDX requires no porting as the binaries are available and work without source changes, except to one shell script.

One of the early problems discovered is that many open source sites distribute archives that were generated using the GNU zip utility. Obtaining and porting the GNU zip is the first task in the porting examples.

The third porting example is GNU Perl. Perl is a popular and powerful scripting tool that is used as a base in many web and e-mail sites. The popular midrange mailing list, Midrange-L at www.midrange.com, is implemented using Perl.

Last, in many of your ports you may find the need to integrate ILE code and PASE code together. An ILE Call example is provided in the lab to show you how to link the two program models together.

AS/400 PASE Summary



- ★ Provides **de facto standard process model, addressing and ASCII** environment on AS/400
- ★ For both system code and applications AS/400 PASE may:
 - Reduce direct porting costs and skill requirements
 - Maintain currency with upgrades
 - Improve time to market
 - Provide staging platform for an application while it is being more fully integrated into an AS/400 solution
- ★ **Leverages stable AIX interfaces in a stable AS/400 environment**
- ★ Delivers improved performance or application footprint in some cases
- ★ Facilitates using A/D tools familiar to many people in the industry
- ★ Delivers safe generation of untrusted code and **potential access to third party compilers or runtimes.**
- ★ AS/400 solution creation activities are key for applications using AS/400 PASE to meet AS/400 customer expectations

AS/400 PASE Summary



Provides de facto standard process model, addressing and ASCII environment on AS/400

For both system code and applications AS/400 PASE may:

- Reduce direct porting costs and skill requirements
- Maintain currency with upgrades
- Improve time to market
- Provide staging platform for an application while it is being more fully integrated into an AS/400 solution

Leverages stable AIX interfaces in a stable AS/400 environment

Delivers improved performance or application footprint in some cases

Facilitates using A/D tools familiar to many people in the industry

Delivers safe generation of untrusted code and potential access to third party compilers or runtimes.

AS/400 solution creation activities are key for applications using AS/400 PASE to meet AS/400 customer expectations

Resources



- ★ **Web page: as400.ibm.com/developer/factory/pase**
 - includes latest PTF list
- ★ **Redbook: SG24-5970 (available 6/2000 as redpiece)**
- ★ **AS/400 Techline**
 - General technical support and questions about AS/400 PASE
- ★ **Rochester Custom Technology Center (CTC)**
 - Fee-based consulting and programming services
- ★ **RCHGO400@us.ibm.com**
 - API usage analysis
 - "Launcher" save files
 - AS/400 header and export file

Resources



Beyond the new redbook (SG24-5970), most of the available resources are on the web. It will be critical for you to provide all of your developers with web access, as well as with external e-mail. The main web page, with links to all other pages that are pertinent, is on the PartnerWorld for AS/400 Developers, AS/400 web site at:

- <http://www.as400.ibm.com/developer/factory/pase/>

All of our work with PASE required a base set of PTFs; before attempting to run any PASE applications, check the above site to ensure you have the latest PTFs installed.

PartnerWorld for AS/400 Developers, AS/400 is also providing an API analysis tool as well as answers to base questions; the most important e-mail address to put in your address book is RCHGO400@us.ibm.com.

AS/400 Techline is providing general technical support and questions about AS/400 PASE.

The Rochester Custom Technology Center (CTC) is providing fee-based consulting and programming services. Contact information for the CTC can be found on the following web site:

- <http://www.as400.ibm.com/service/>

Trademark



Copyright International Business Machines Corporation 2000

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States or other countries or both:

ADSTAR	DataGuide	NetFinity
Advanced Function Printing	DataPropagator	NetView
AFP	DB2	OfficeVision
AIX	IBM	OS/2
AnyNet	IBM Network Station	OS/400
Application Development	Information Warehouse	PowerPC
APPN	Integrated Language Environment	PowerPC AS
AS/400	Intelligent Printer Data Stream	Print Services Facility
AT	IPDS	PSF
BrioQuery	JustMail	SanFrancisco
BRMS	Net.Commerce	SmoothStart
Client Series	Net.Data	SystemView

cc:Mail, Lotus, Lotus Notes, Lotus Domino, Domino.Action, and Domino.Merchant are trademarks or registered trademarks of Lotus Development Corporation.

Microsoft, Windows, NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Java and all Java-related trademarks or logos are trademarks or registered trademarks of Sun Microsystems, Inc in the United States and other countries.

IBM's VisualAge products and services are not associated with or sponsored by Visual Edge Software, Ltd.

Pentium is a trademark of Intel Corporation.

Other company, product, and service names may be trademarks of their respected providers.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources. Sources for non-IBM list prices and performance numbers are taken from publicly available information. Including vendor announcements, vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdraw without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific statement of direction.