JavaOne

Sun's 1997 Worldwide Java Developer Conference

Enhancement for Mission Critical Applications

Todd Yancey

Fujitsu Software Corporation

Presentation Outline

- Fujitsu Overview
- Mission Critical Systems
- Java^{TM} Acceleration
- Java Business Class Libraries
- Java Wrapper for Legacy Application Integration
- 3270 Host Terminal Emulator

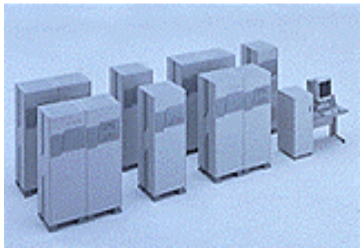
Fujitsu Limited

2nd largest computer company

- Annual revenues -- $37 Billion
- Employees -- 164,364
- 400 technical companies
  - Amdahl
  - Fujitsu Software
  - HaL
  - ICL
  - Ross Technology

Fujitsu Products

Super Computers


Mainframes

Desktop PCs

MO Drives

Laptops

Tablets

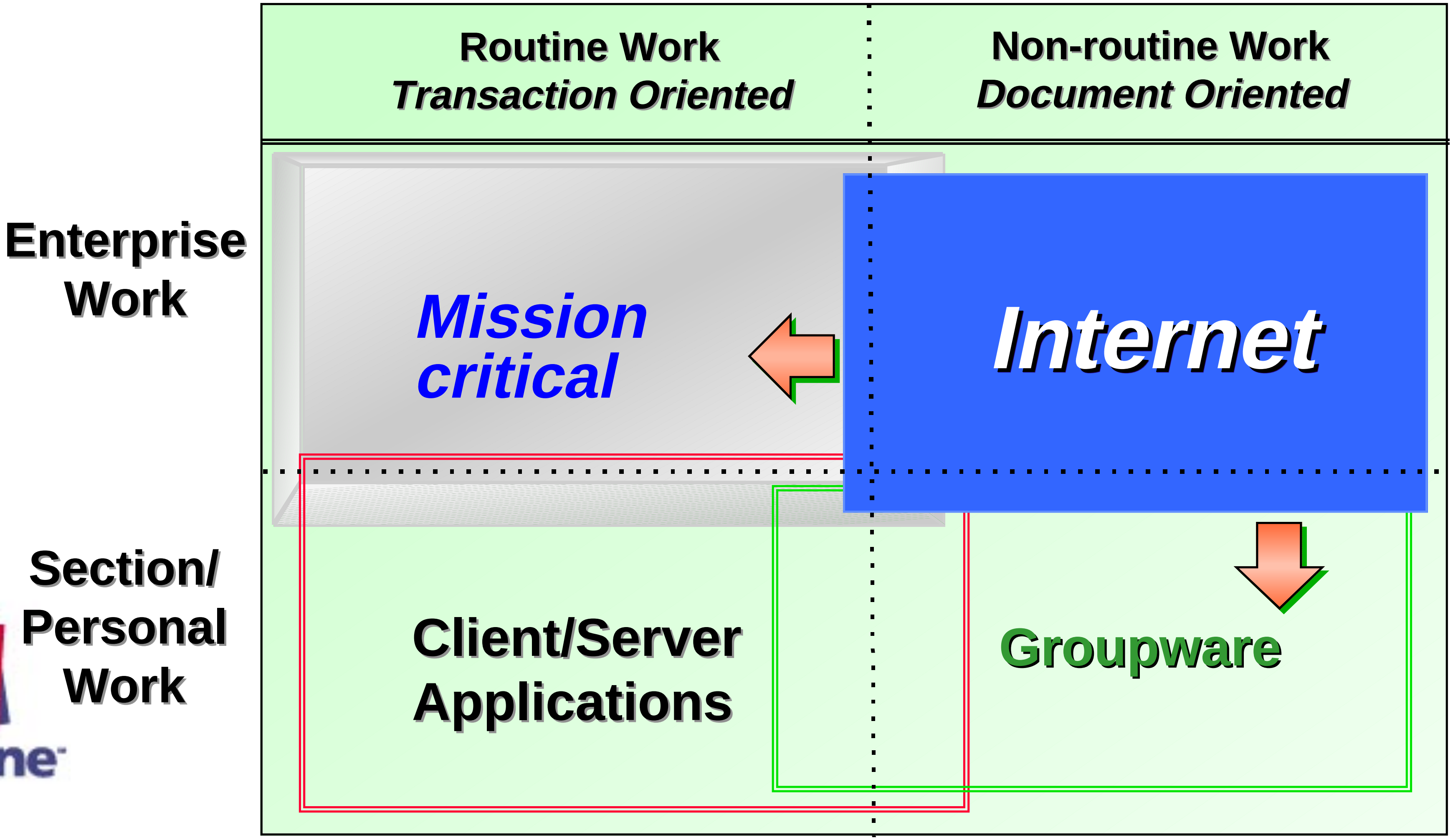
Point of Sale

Fujitsu

- SPARC processors
  - HaL SPARC64™
  - hyperSPARC™
  - TurboSPARC™
  - microSPARC-II™
- HaL named Workstation of the Year
- $3 billion per year in R&D

Network Computing

Java is the key technology



Mission Critical Systems

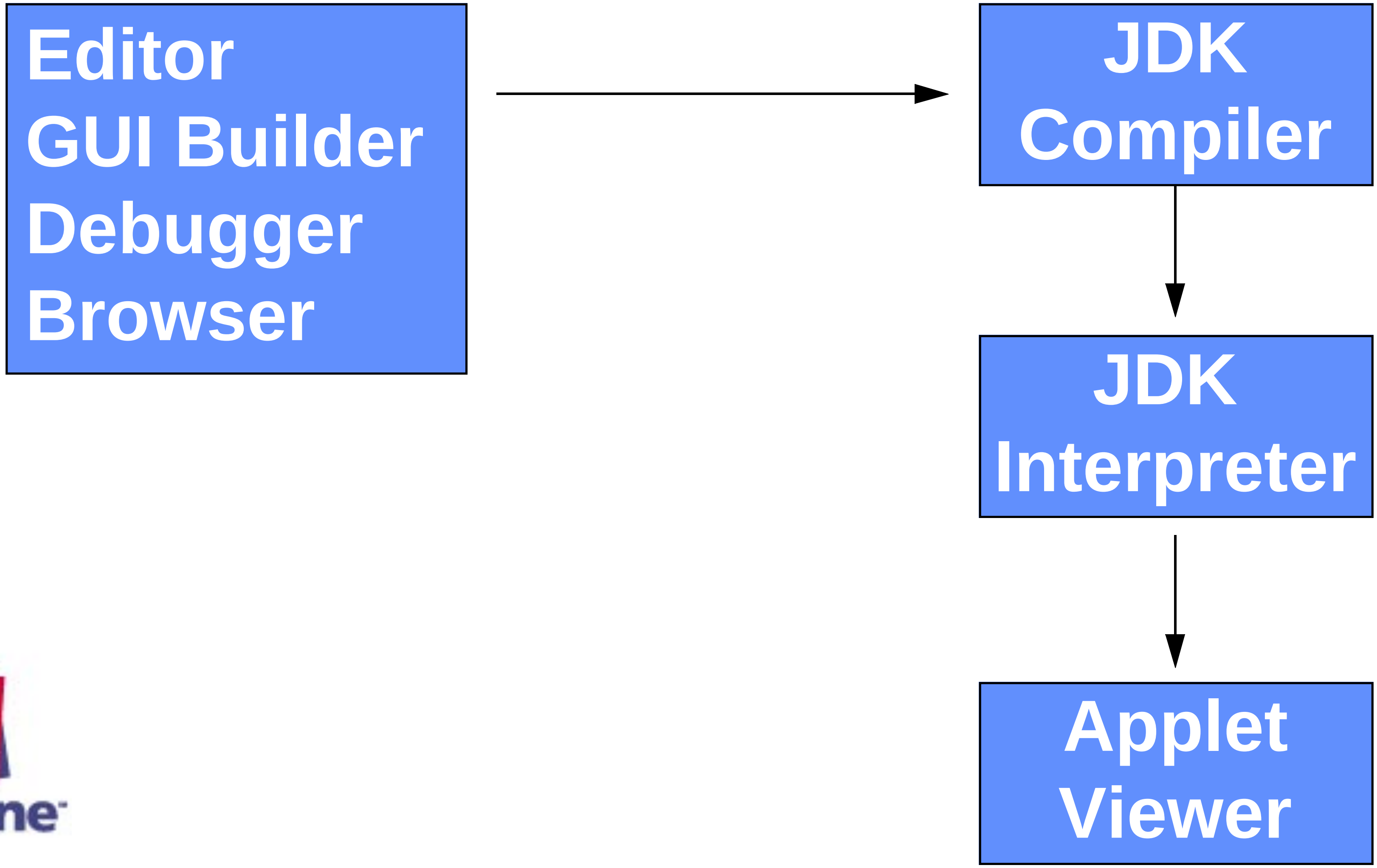
Requirements

- High speed execution
- Sufficient class library
- Reuse of legacy applications
  - Redeployment of legacy programs
  - Connection of legacy applications
- Internationalization

Basic Assumptions

- Java is an alternative to C++
- "Servlets" will evolve to the point where execution performance on the server is critical

Java Development



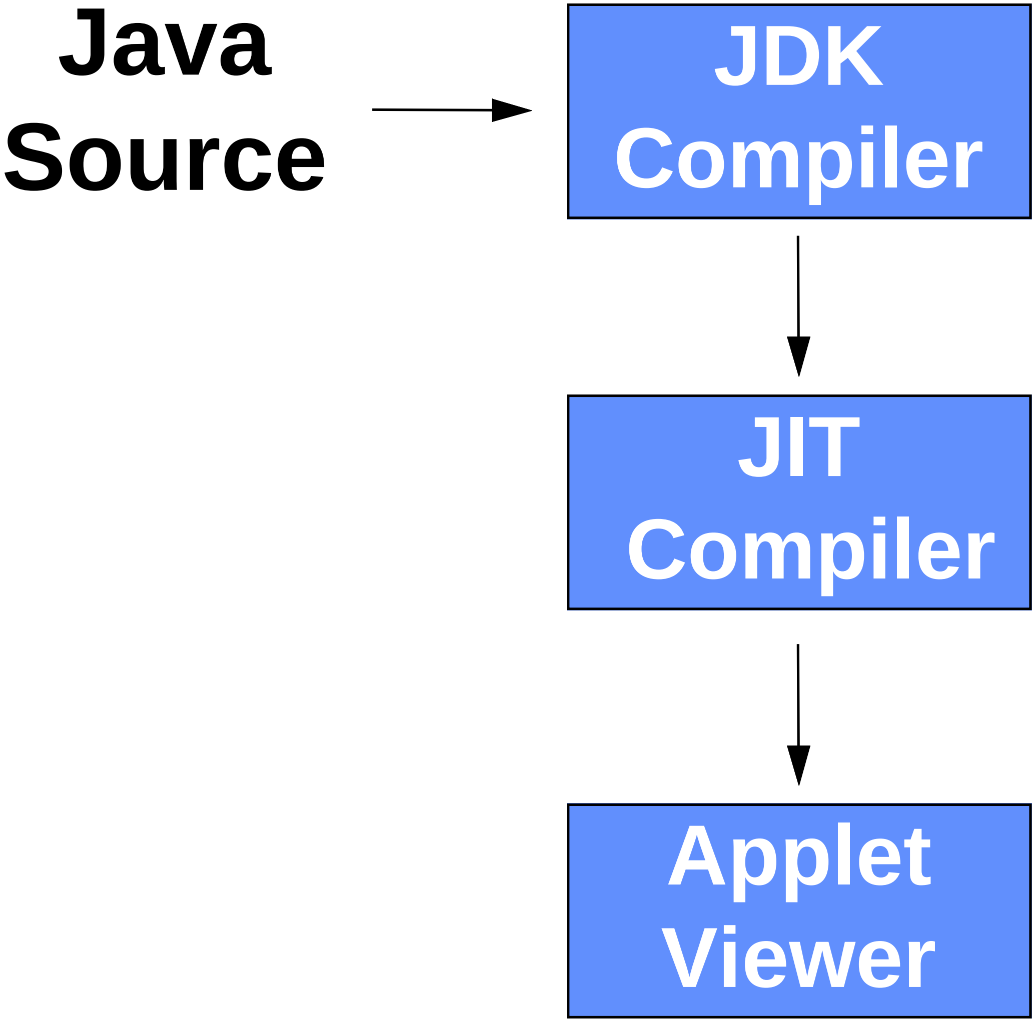
Editor
GUI Builder
Debugger
Browser

JDK Compiler

JDK Interpreter

Applet Viewer

JIT Compiler



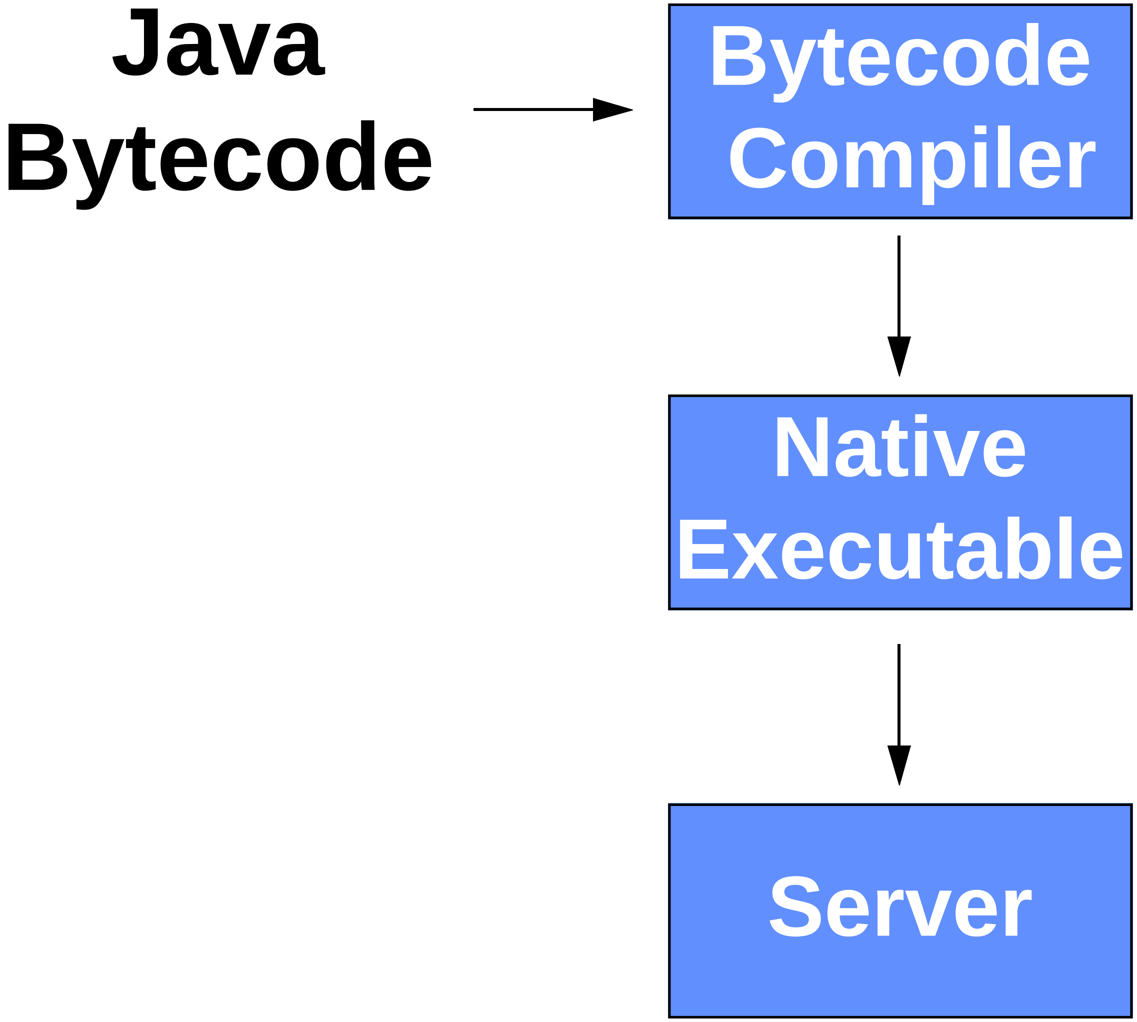
Java Source

JDK Compiler

JIT Compiler

Applet Viewer

Bytecode Compiler



Java Bytecode

Bytecode Compiler

Native Executable

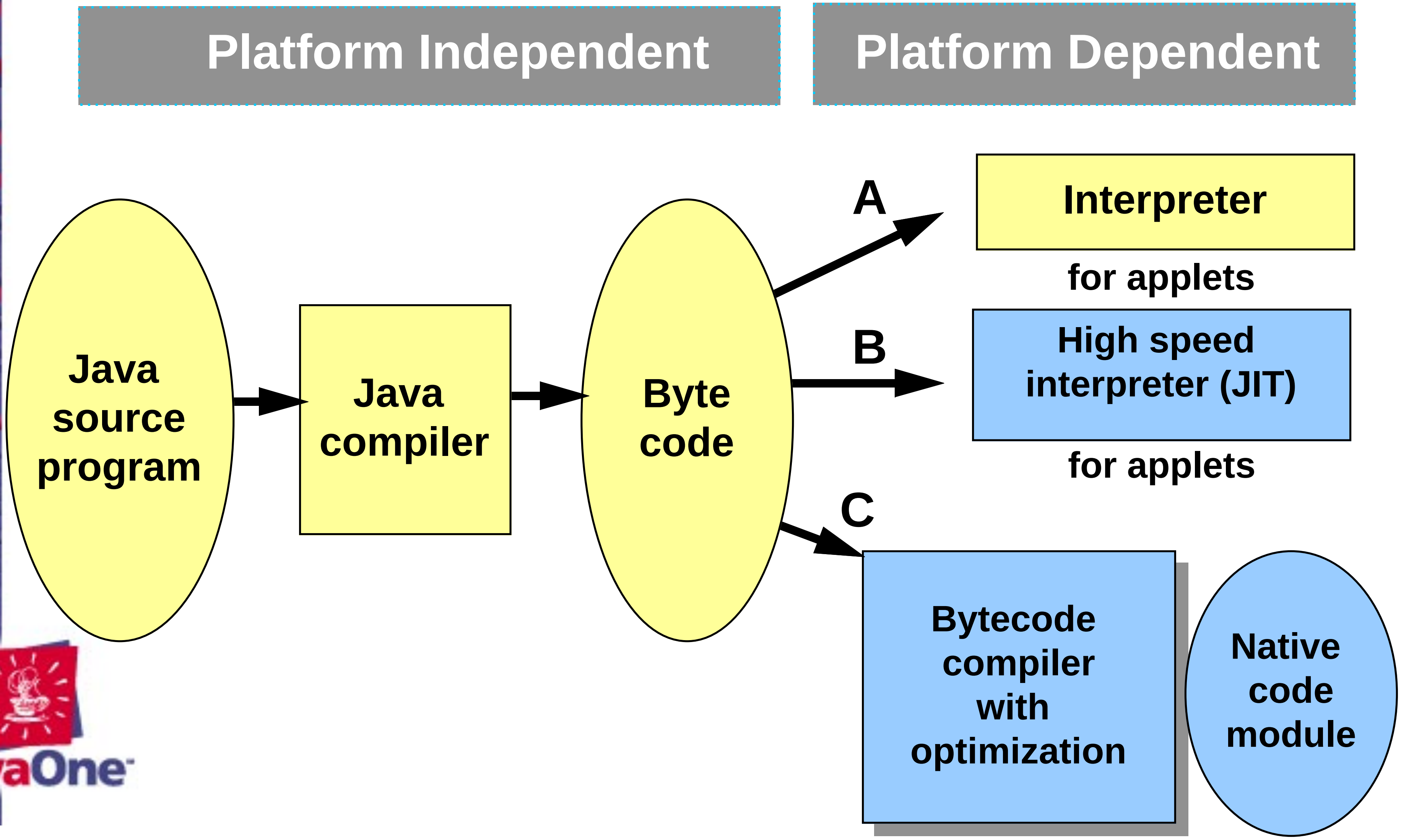
Server

Java Acceleration

SPARC-Solaris and Intel-Windows

- Just In Time compiler
- Static bytecode compiler

Java Application Program Execution



Accelerate Java Program Execution

Execution method	Application portability	Execution attribute	Execution performance	Applicability	Schedule
Interpreter	good	one time	1	immediately executable (on Web browser)	97/2Q
JIT			5-10		97/3Q
Bytecode compiler		iterative	5-50	Client/Server application programs	97/2Q

Just In Time Compiler

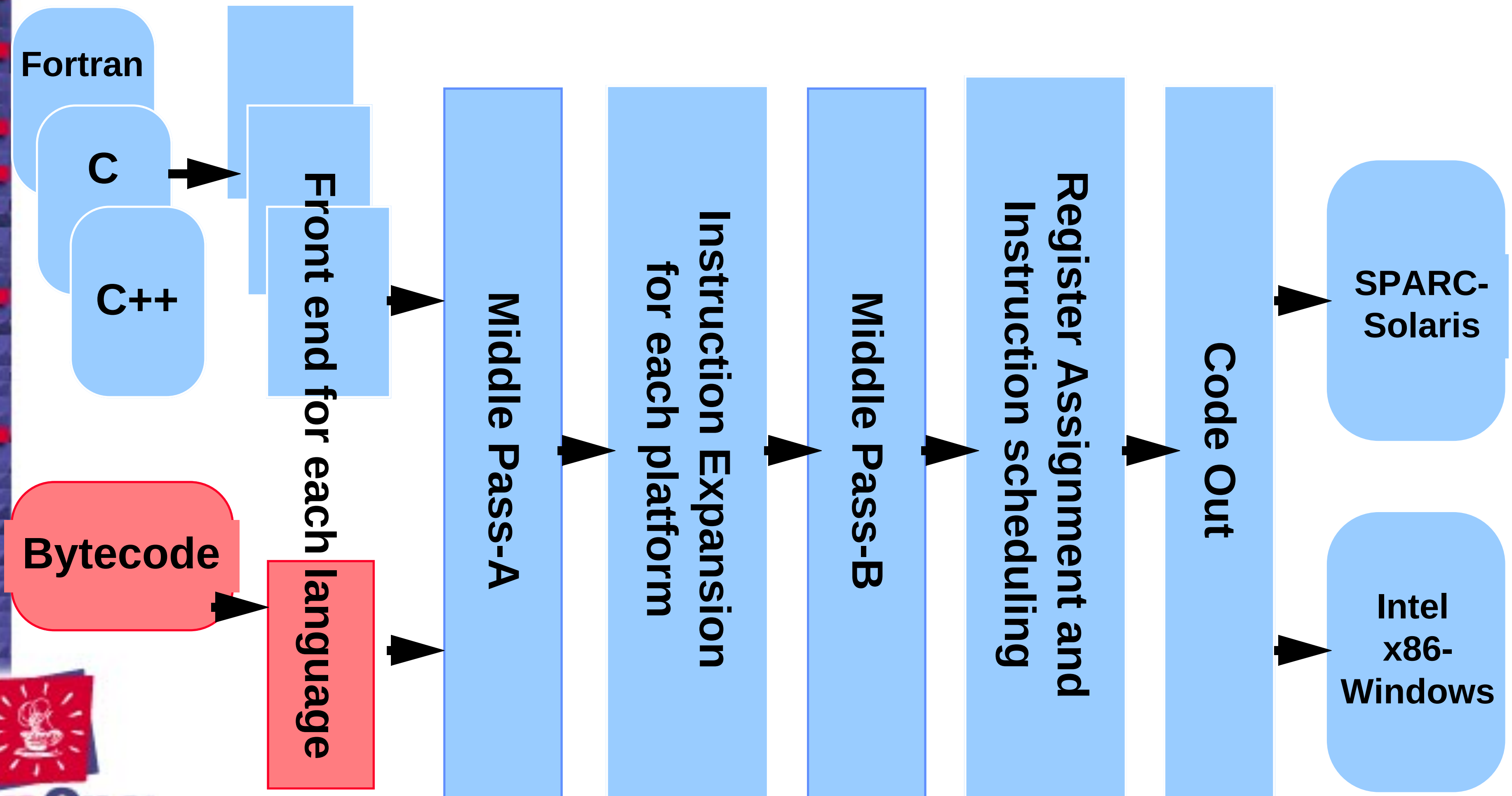
Application portability

- Bytecode compiled during program execution
- Bytecode is used to achieve high speed execution
- Replace Java Interpreter by JIT

Bytecode Compiler

- Bytecode compiler reads bytecode and outputs Optimized Native Code
- Because it is invoked before the execution, it has enough time to optimize code like other languages (C, C++, and Fortran)

Bytecode Compiler Structure


Fortran
C
C++
Bytecode
Front end for each language
Middle Pass-A
Instruction Expansion for each platform
Middle Pass-B
Register Assignment and Instruction scheduling
Code Out
SPARC-Solaris
Intel x86-Windows

Same structure for C or Fortran compilers

Optimization Technologies

- Basic
- High level
- Advanced
- Java specific

Basic Optimization

Same technology for C++ and Fortran

- Optimize each instruction
- Dual optimization phases
  - Front end
  - Middle pass
  - Instruction expansion
  - Register assignment and scheduler
  - Code out

High Level Optimizations

- Originally used for supercomputers
- Traditional optimization
- Program structure optimization
  - Loop unrolling
  - Loop fusion
  - Inline expansion

Advanced Optimizations

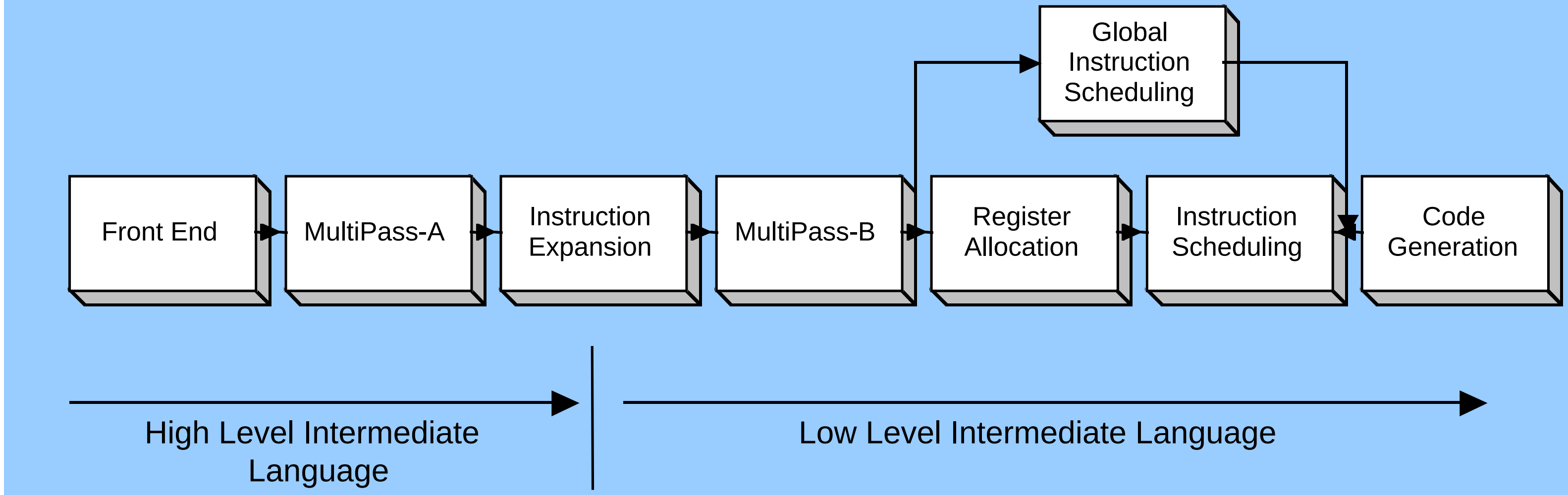
- Architecture specific optimizations
- Local instruction scheduling
- Global instruction scheduling

Java Specific Optimization

- Class initializer optimization
  - Eliminating judgement if already initialized
- Error check elimination
  - Eliminating array boundary check

Compiler Structure

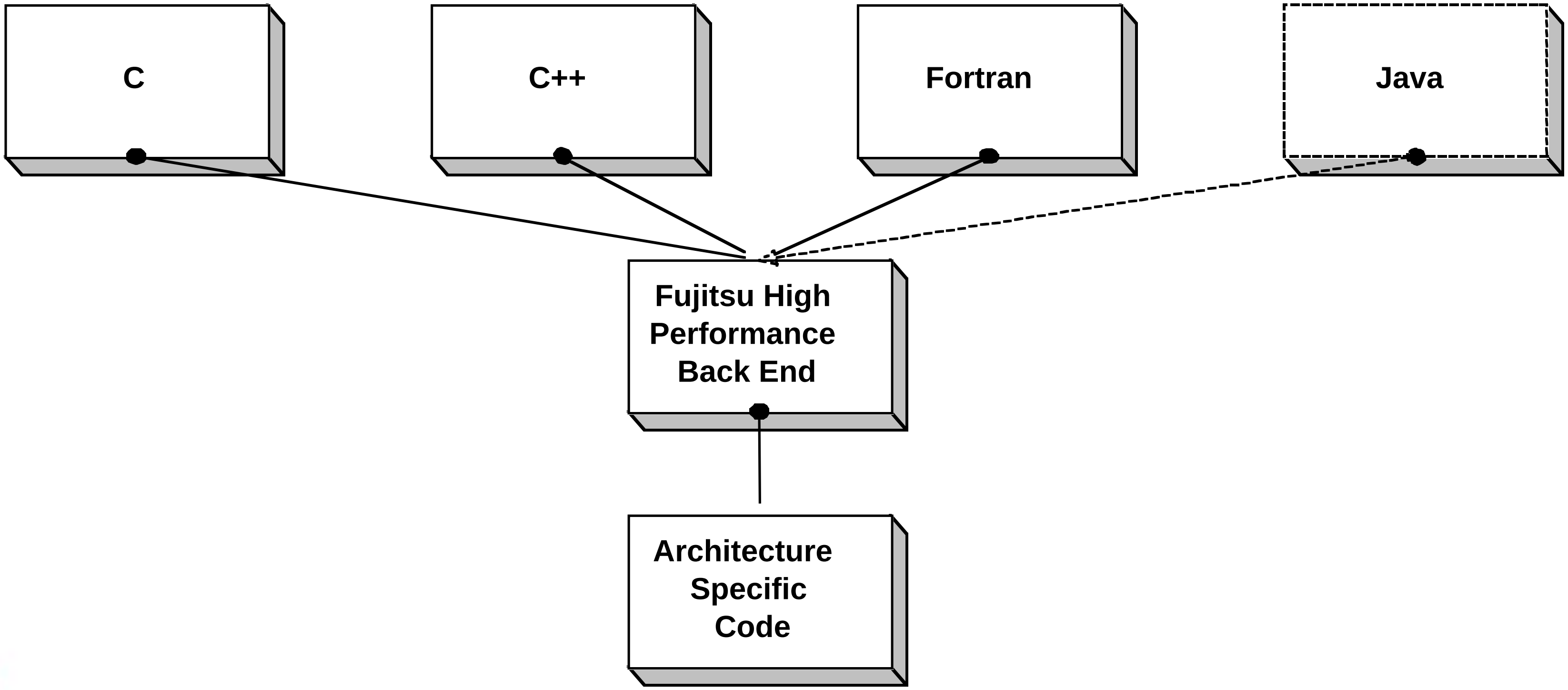
Fujitsu Compiler Phases



MultiPass-A: Traditional optimizations (common expression elimination, etc.), Program reconstruction

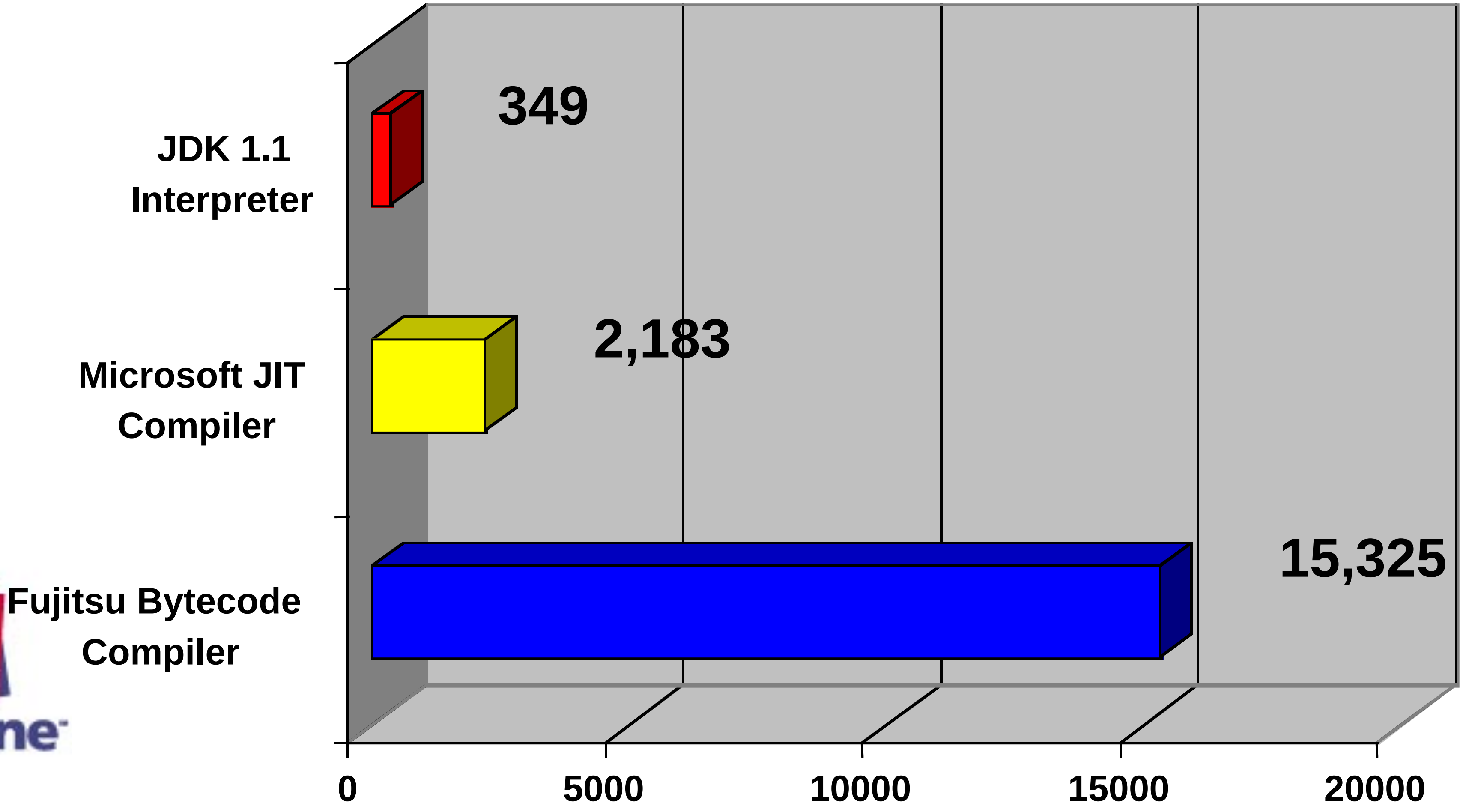
MultiPass-B: Advanced optimizations specific to hardware architecture.

Common Back End



CaffeineMark 2.5 Performance

Pentium 150MHz, 64MB, Windows NT 4.0



Java Business Class Libraries

- GUI libraries
- Distributed processing libraries
- Database libraries
- Business class libraries

GUI Libraries

Based on JDK 1.1 AWT

- Extension of GUI components
  - Text field with input validity check
  - Format editing
  - List and choice for data classes
- New GUI components
  - Listview, table, form, etc.
- GUI templates for DB processing

Distributed Processing Libraries

3-tier synchronous distributed processing

- Client only need be aware of stationary Master agent objects
- Service redefine methods in mobile Servant agent or stationary Master agent objects
- Each service is executed by invoking a Servant through the Master

Multi-Server Support

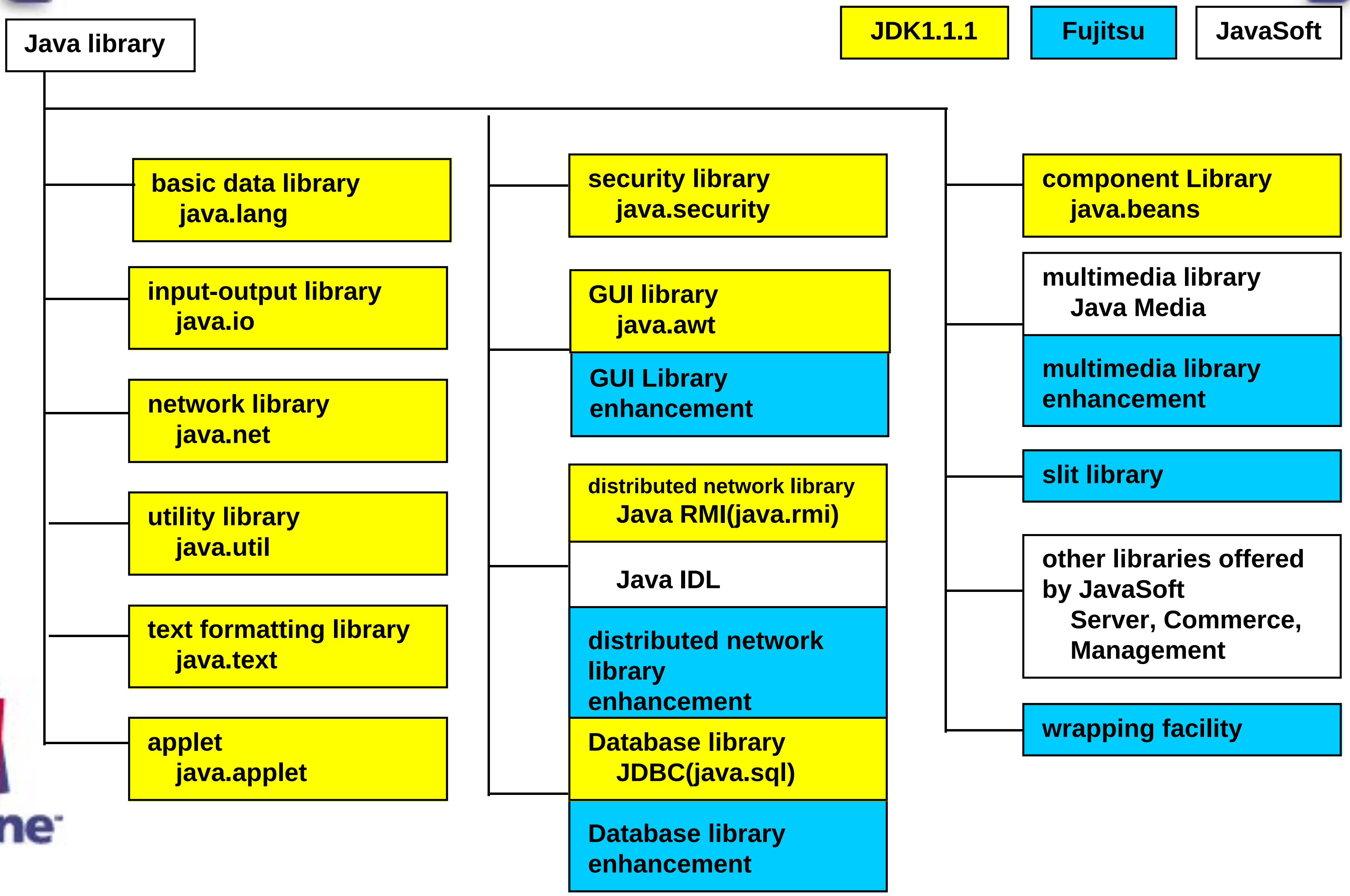
- Master/Servant invocation can be cascaded so that Servants can start new Masters
- Using itinerary objects, traveling Servants that traverse many servers can be implemented

Database Libraries

Without complex SQL or ODQL

- Data extraction using parameters
  - Table name for data extraction
  - Column name for data extraction
  - Condition for data extraction
  - Condition for sort
  - Condition for group
- Direct updating of extracted data

Business Class Libraries



JDK1.1.1

Fujitsu

JavaSoft

Java library

basic data library
java.lang

input-output library
java.io

network library
java.net

utility library
java.util

text formatting library
java.text

applet
java.applet

security library
java.security

GUI library
java.awt

GUI Library enhancement

distributed network library
Java RMI(java.rmi)

Java IDL

distributed network library enhancement

Database library
JDBC(java.sql)

Database library enhancement

component Library
java.beans

multimedia library
Java Media

multimedia library enhancement

slit library

other libraries offered by JavaSoft
Server, Commerce, Management

wrapping facility

Java Wrapper for Legacy Applications

Supported components

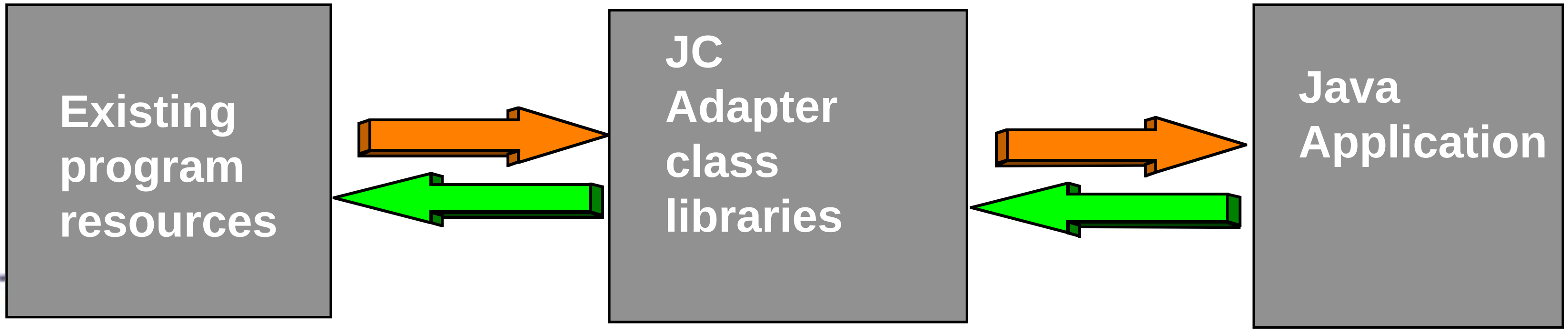
- COBOL library
- C library

Java Wrapper

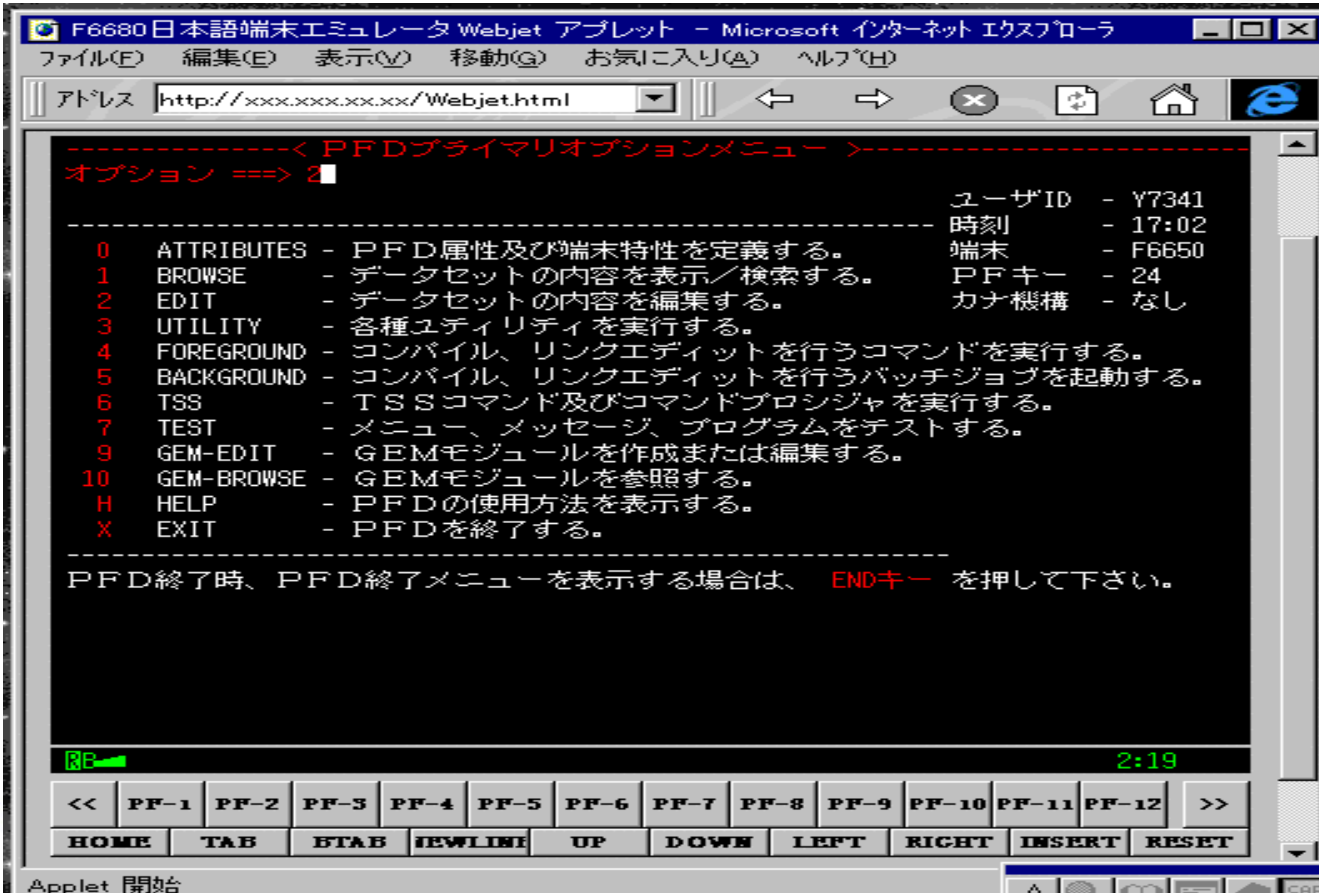
Wrapper generates a Java class to interface with existing program resource

Integrate growing number of components to Java

Java Wrapper consists of class libraries for wrapping and a GUI Wizard for class creation

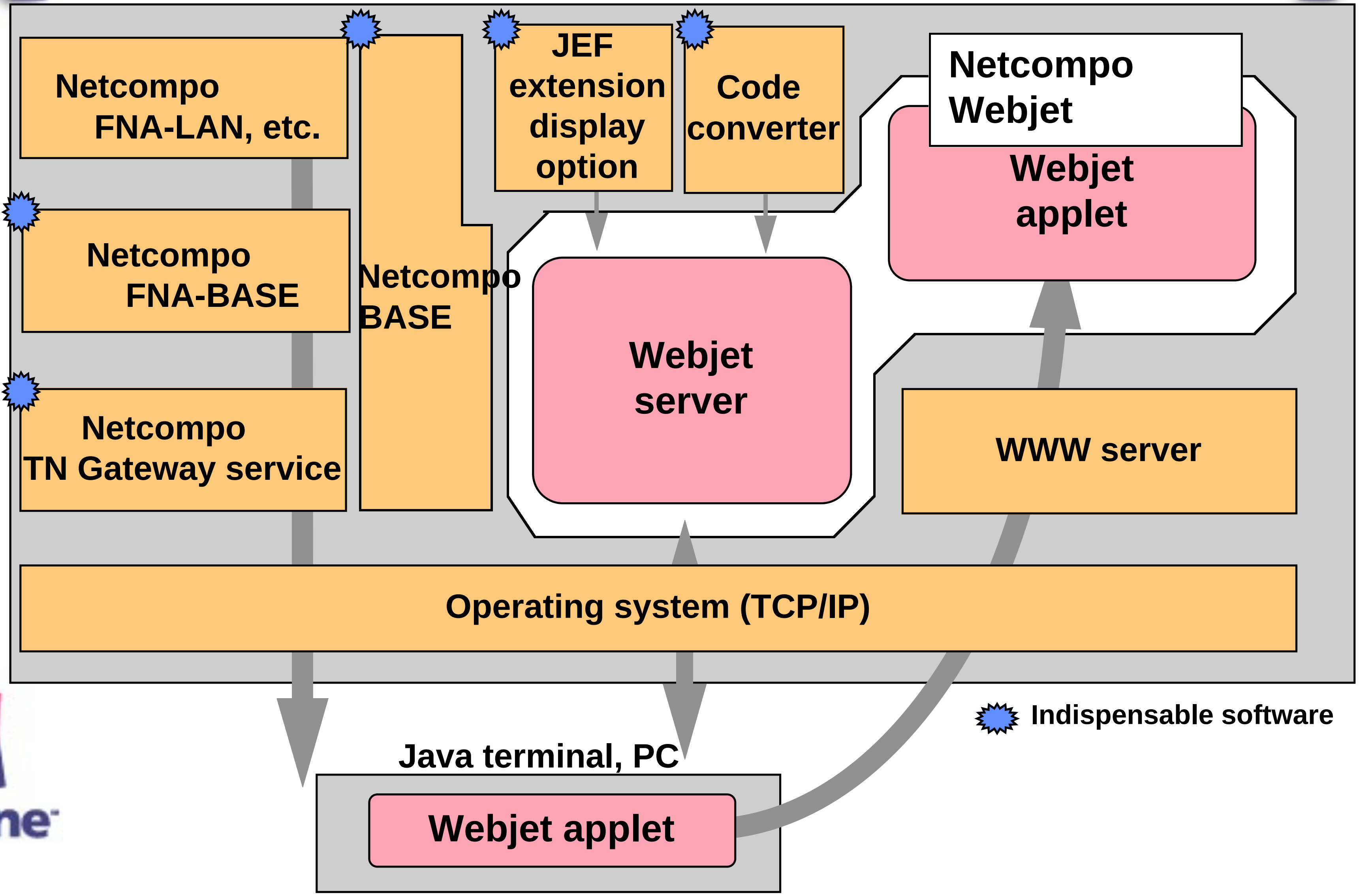


IBM 3270 Host Emulator with Fujitsu 6680/6970



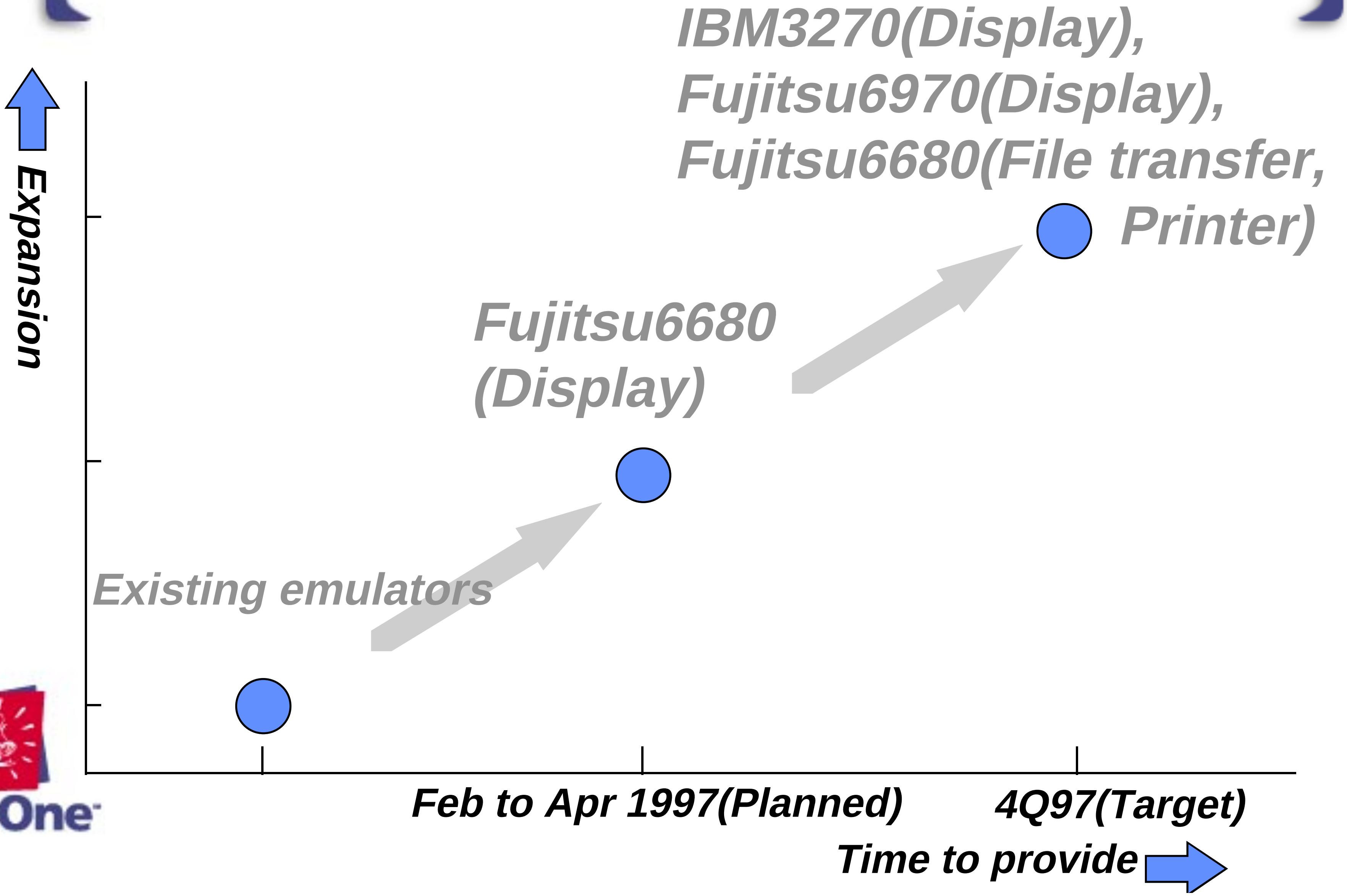
```
---------------< ＰＦＤプライマリオプションメニュー >-------------------------
オプション ===> 2
                                                           ユーザID   - Y7341
----------------------------------------------------------- 時刻       - 17:02
  0   ATTRIBUTES - ＰＦＤ属性及び端末特性を定義する。      端末       - F6650
  1   BROWSE     - データセットの内容を表示／検索する。    ＰＦキー   - 24
  2   EDIT       - データセットの内容を編集する。          カナ機構   - なし
  3   UTILITY    - 各種ユティリティを実行する。
  4   FOREGROUND - コンパイル、リンクエディットを行うコマンドを実行する。
  5   BACKGROUND - コンパイル、リンクエディットを行うバッチジョブを起動する。
  6   TSS        - ＴＳＳコマンド及びコマンドプロシジャを実行する。
  7   TEST       - メニュー、メッセージ、プログラムをテストする。
  9   GEM-EDIT   - ＧＥＭモジュールを作成または編集する。
 10   GEM-BROWSE - ＧＥＭモジュールを参照する。
  H   HELP       - ＰＦＤの使用方法を表示する。
  X   EXIT       - ＰＦＤを終了する。
-----------------------------------------------------------
ＰＦＤ終了時、ＰＦＤ終了メニューを表示する場合は、 ENDキー を押して下さい。

RB                                                                     2:19
```

3270 Host Emulator



✹ Indispensable software

Emulator Roadmap



Expansion

Existing emulators

Fujitsu6680
(Display)

IBM3270(Display),
Fujitsu6970(Display),
Fujitsu6680(File transfer,
Printer)

Feb to Apr 1997(Planned)

4Q97(Target)

Time to provide

Q&A

Todd Yancey
Fujitsu Software Corporation
Developer Tools Group
Phone: (408) 456-7785
FAX: (408) 456-7050
Email: tay@adtools.com