



Transaction Processing Facility

TPF Update

TPF Users Group, Acapulco

May 2004

Stuart Waldron



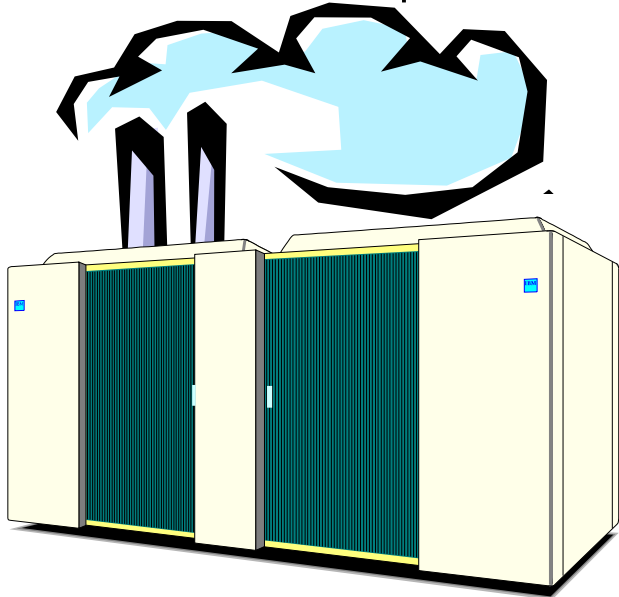
How do we keep up with the speed of change?

- We can continue to adopt the chaotic hoard theory of programming or
- Working towards a more complete, ubiquitous, developer environment that leverages the skills you already have
 - ▶ Both college hires and those with 20+ years
- Allow for the massive use of ported code
 - ▶ No one can afford to write it all from scratch
 - ▶ This includes, largely, open source software
- Make a greater use of middleware
 - ▶ directories, web servers, security servers, DB servers, queues, and ???



So How are we doing?

- z/OS development model
 - JCL, PDS, etc.
 - z/VM execs
- RYO Tooling
- Assembler as the language of only choice
- All code either IBM provided or RYO



- Linux development model
 - makefiles, libraries
 - UNIX - browser based tooling
- COTS tooling
- C/C++ , PERL, more to come
- Hundreds of thousands of lines of ported code already in TPF

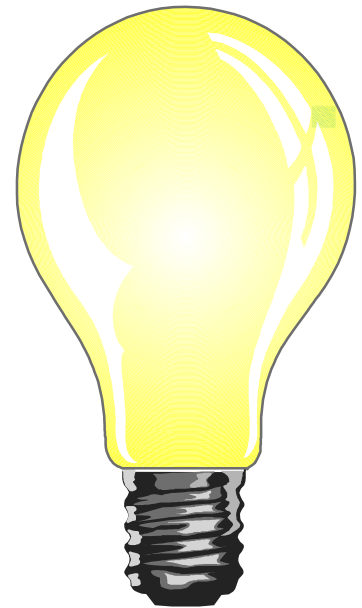


LAMP

- Linux
 - ▶ TPF moving to a Linux programming model
- Apache
 - ▶ TPF has been running Apache for years
- MySQL
 - ▶ Customer requirements for a native SQL engine?

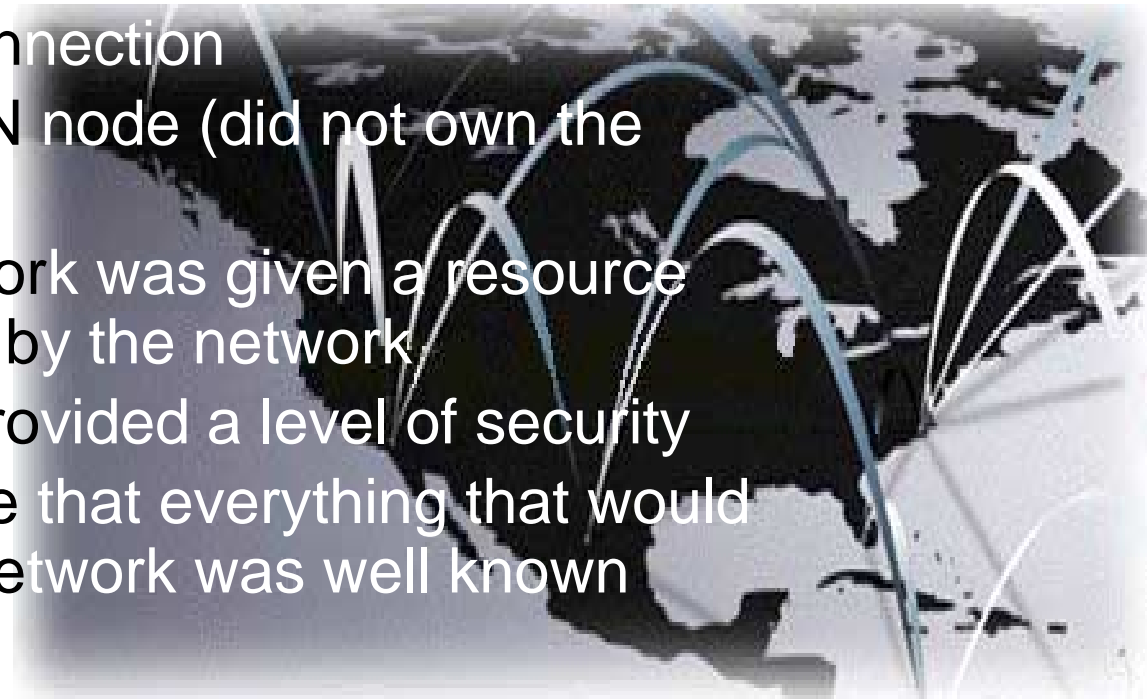
PMP

- ▶ As a backend sever, are there any requirements for a graphical user interface?



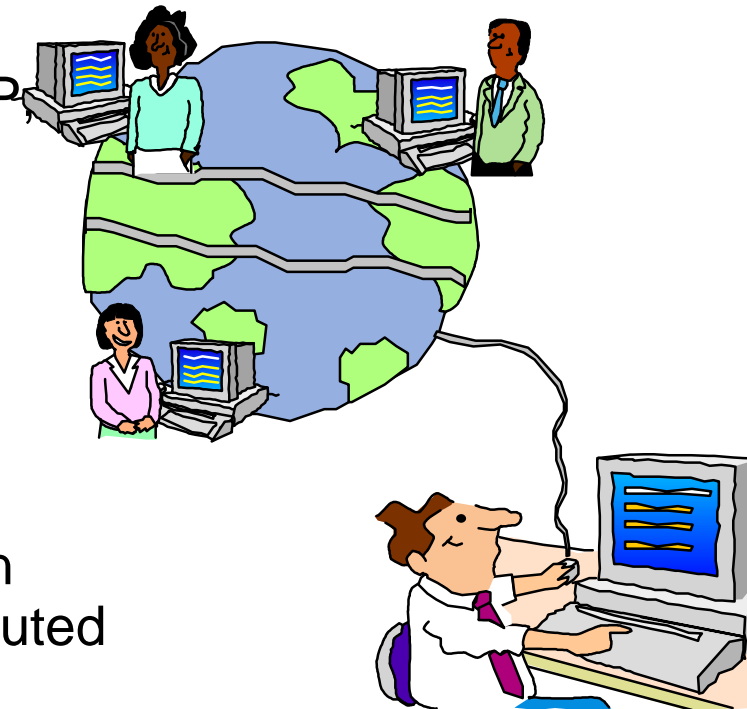
But we still have to talk to the Outside world, we started with...

- ALC
 - ▶ 6 bit connectionless protocol (p1024b)
- SNA
 - ▶ Connection oriented protocol
 - ▶ PU5 , TPF own the connection
 - ▶ PU2.1, TPF was a LEN node (did not own the connection)
 - ▶ Everything in the network was given a resource "name" and controlled by the network
 - ▶ Level of control also provided a level of security
 - ▶ Built under the premise that everything that would be connecting to the network was well known



The Internet World

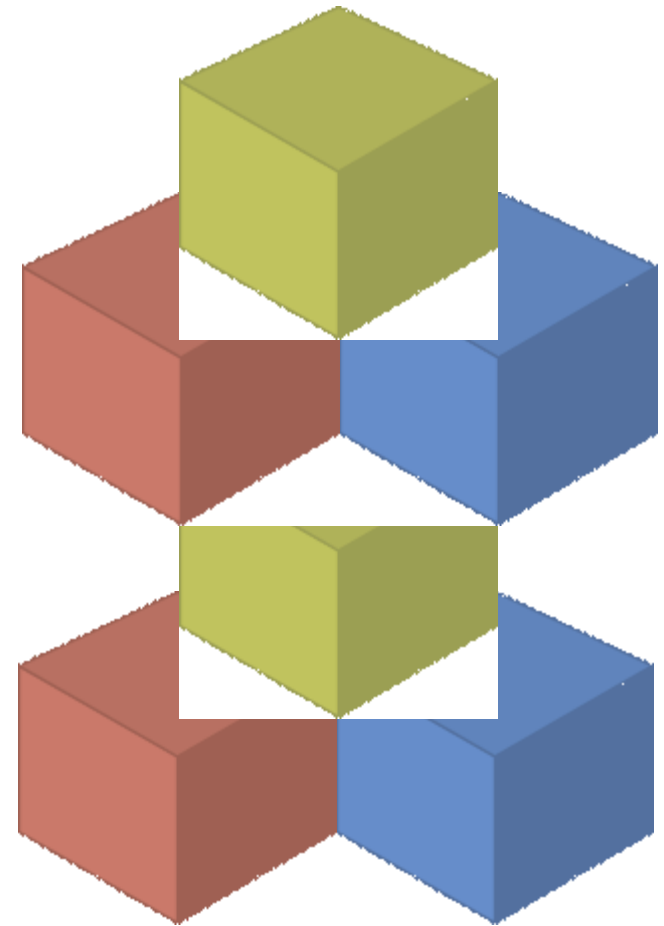
- Largely built on connection oriented TCP/IP
 - ▶ However there is very little knowledge of the end to end connection above the transport layer
 - ▶ We made many improvements in security and reliability (many local issues remain)
- Many protocols, patterns of use
 - ▶ MQ, HTTP, SMTP, POP, IMAP, SNMP, FTP, MATIP, IIOP supported by TPF
- Connection Architectures
 - ▶ two tier (very reliable, easier to operate)
 - ▶ three tier (invented to support less capable systems)
 - ▶ n tiers (evil)
- Moving to a Service Oriented Architecture in an attempt to better handle the demands of distributed computing



What is a Service Oriented Architecture?

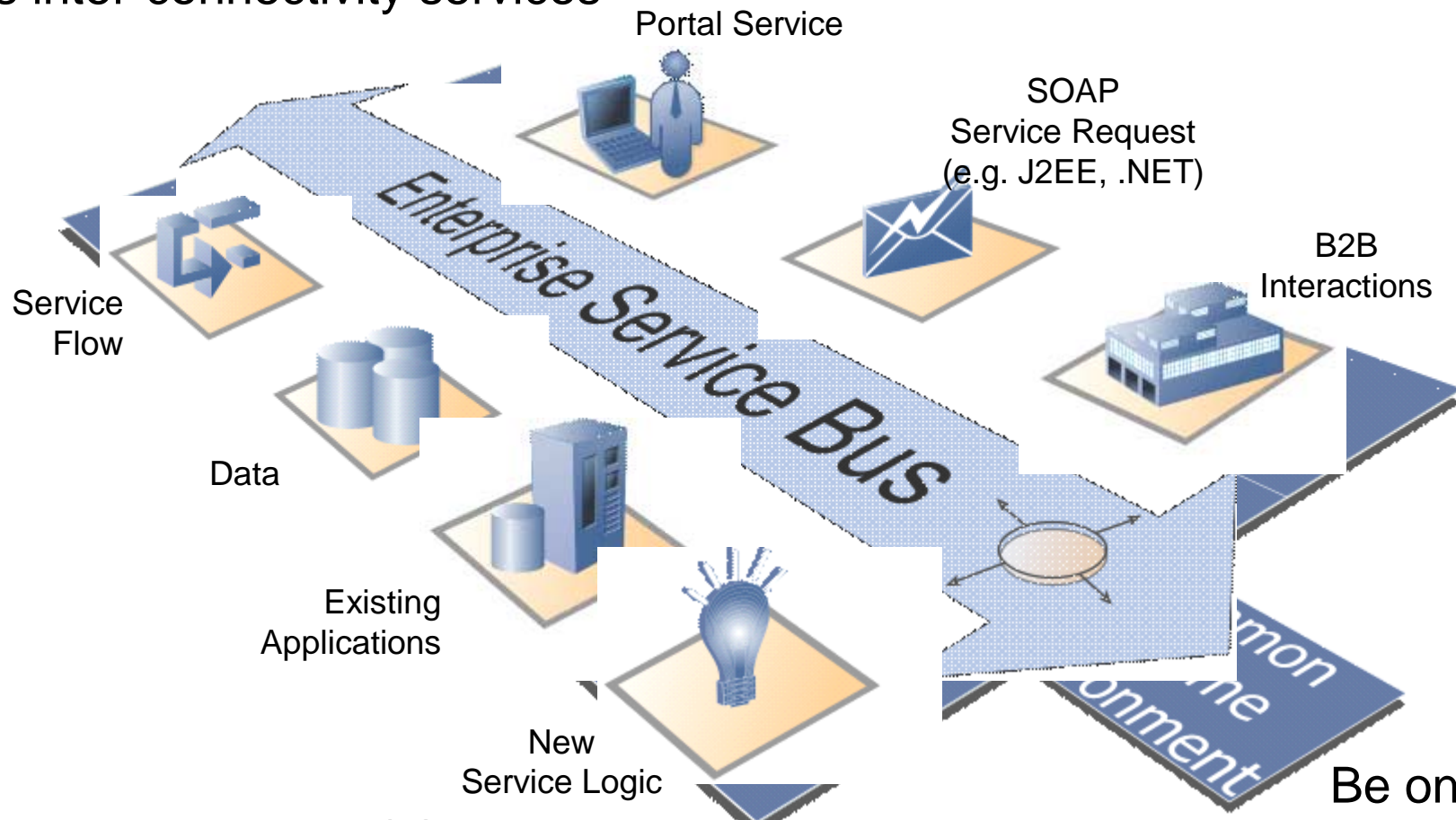
- An **approach** for **building** distributed systems that deliver application functionality as **services** to either end-user applications or other services

- It defines :
 - ▶ An architecture that leverages **open standards** to represent software **assets as services**.
 - ▶ Provides a **standard way of representing and interacting** with software assets
 - ▶ Individual software assets become **building blocks** that **can be reused** in developing other applications
 - ▶ **Shifts focus to application assembly** rather than implementation details
 - ▶ Used externally to **integrate with applications outside of the enterprise**



The Enterprise Service Bus is a logical architectural construct

Provides inter-connectivity services

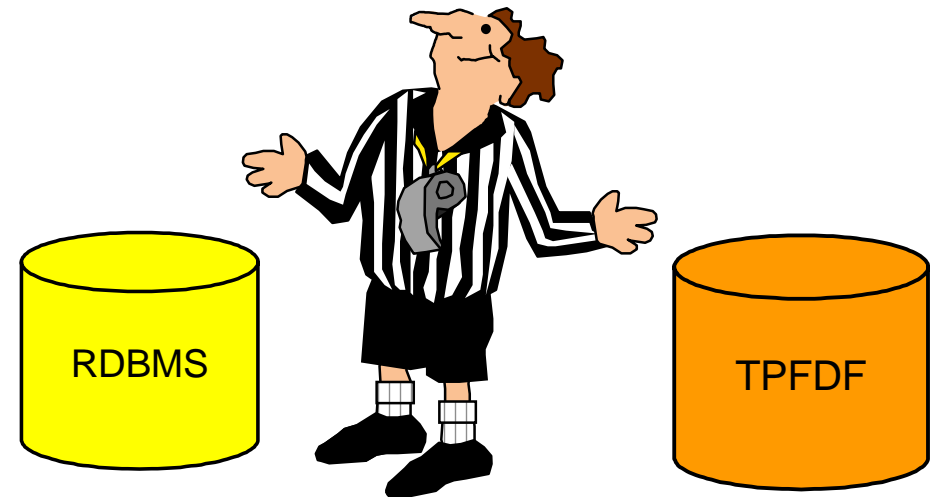


An implementation of an SOA architecture

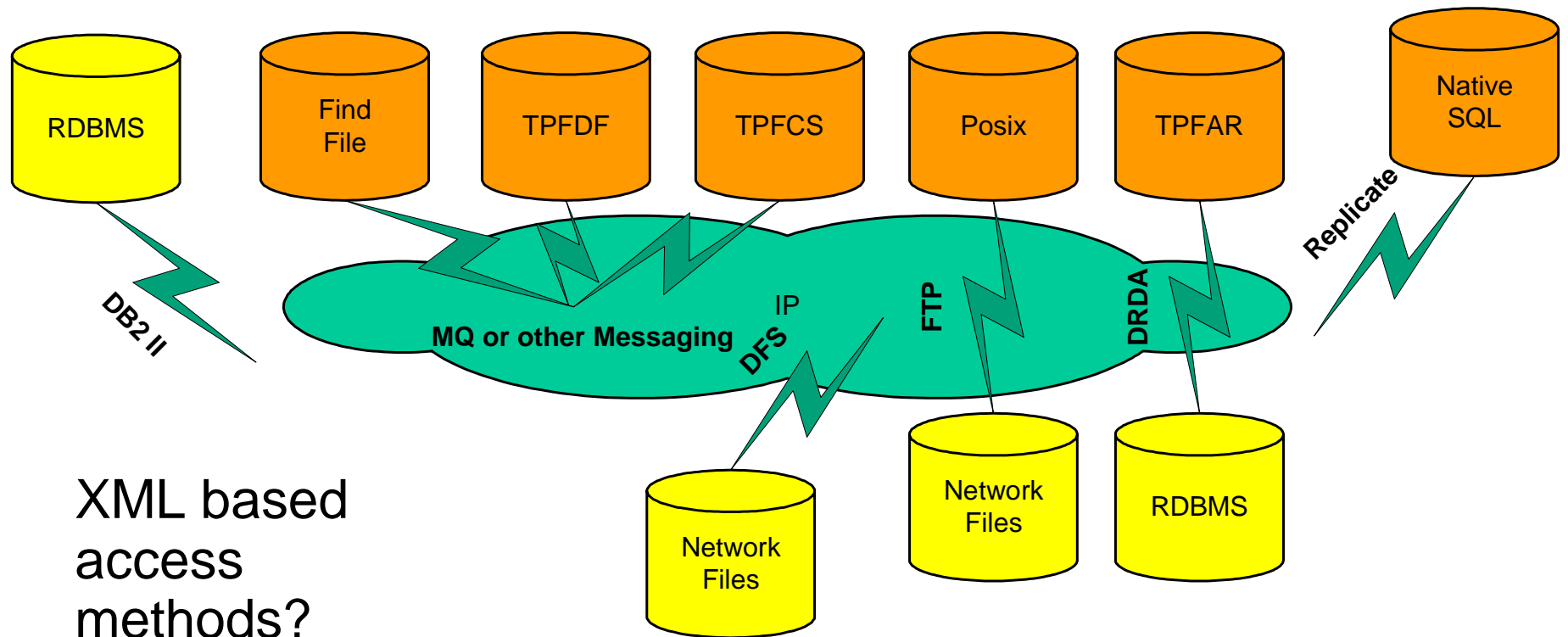
Be on the Bus!
Not in front of it!

TPF as a DB Server

- Relational Database has inherent limitations on scale and availability
 - Relies of Data Replication or Segregation to scale
 - Replication
 - ▶ Very common, reaches practical limits with master DB is wholly occupied with replication
 - Segregation
 - ▶ Has the drawback of extending DB awareness into the network causing the distribution of business logic as well
 - Both often used together, very complex to manage
- TPF uses a pointer driven data model though less flexible, very powerful in it's ability to scale and provide high reliability
 - There is little need in TPF to replicate or distribute data
 - ▶ Contend there is no such thing as flawless replication



DataBase Discussion



Strategy

- Find a means to exploit the power of TPF without the requirement for such a highly specialized skill set
- Find a means to integrate TPF via service based architecture to the many distributed technologies that will be necessary
- Find a means to secure and manage the whole in a way that comes as close as possible to the advantages of a monolithic system



1. What is Open Source Software (OSS)?

- Freely available software in source code format
- Redistribution cannot be prohibited (royalty free)
- Often developed in a collaborative community effort
- OSS community very sensitive about meaning & use of term "open source"
 - ▶ ex: OSI and Open Source, FSF and Free Software
- Most important OSS **license** criteria
 - ▶ source code must be available
 - ▶ right to use and copy software
 - ▶ right to modify/prepare derivative works
 - ▶ right to distribute (in original or modified form)
 - ▶ not necessarily: code free of charge, charges possible distribution
 - ▶ originator or distributor assumes no liability
 - charges possible for guarantees, indemnities

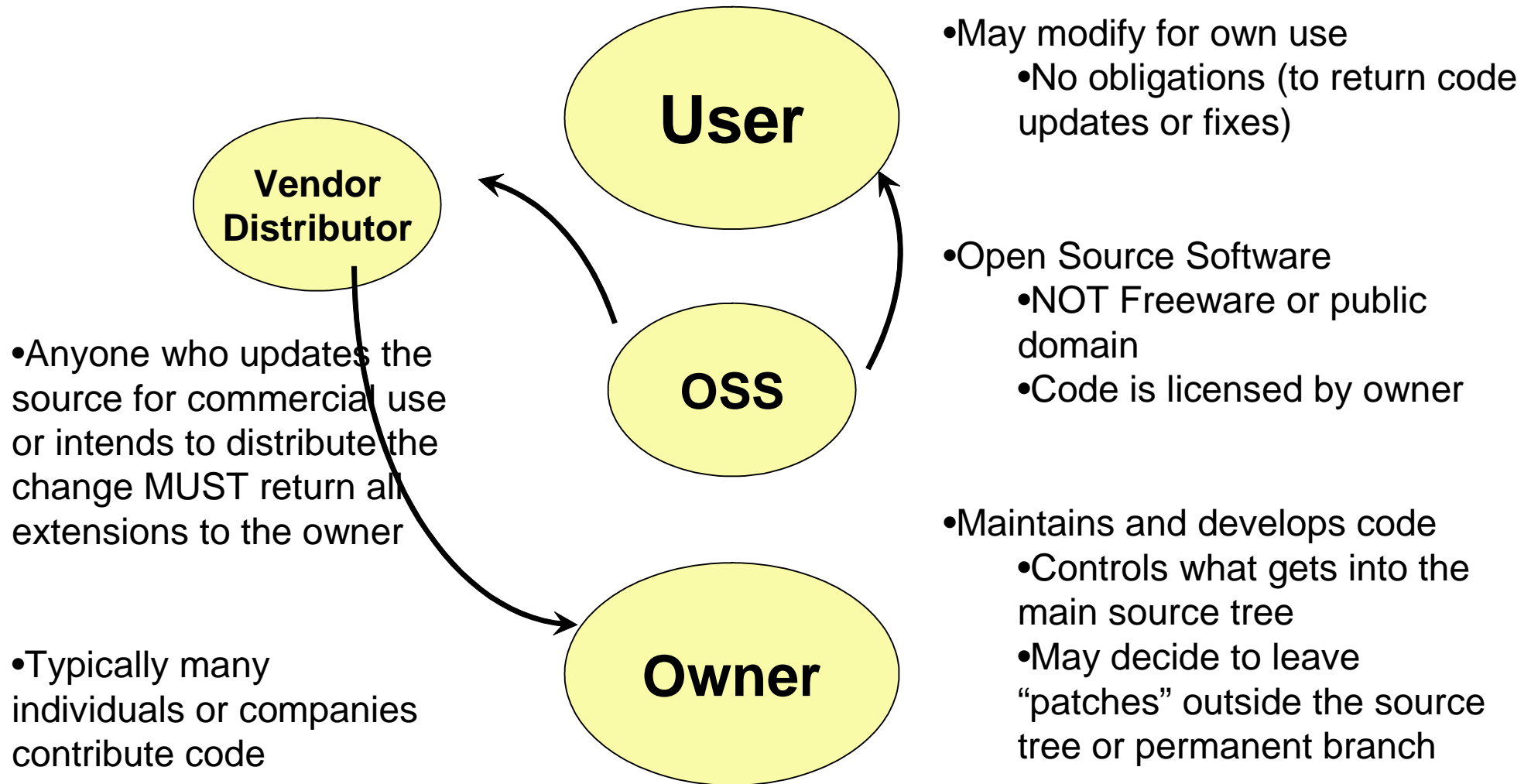
www.opensource.org



1. What is Open Source Software (OSS)? cont.

- Anyone can copy, modify and redistribute the source code without paying royalties or fees
- Many OSS licenses in use today
 - ▶ Each open source license reflects specific objectives of its author(s)
 - ▶ **Most of the obligations in open source licenses relate to the distribution of OSS**
- **Not "Public Domain"** - Frequent misconception
- Copyright law gives author rights
 - ▶ **author determines appropriate uses of a work**
 - e.g. you can't reproduce or modify a work without author's permission
 - ▶ author may grant permission via a license
 - ▶ License grants permission, **but may also impose obligations**

OSS Relationships



1.2 The OSI Open Source Definition

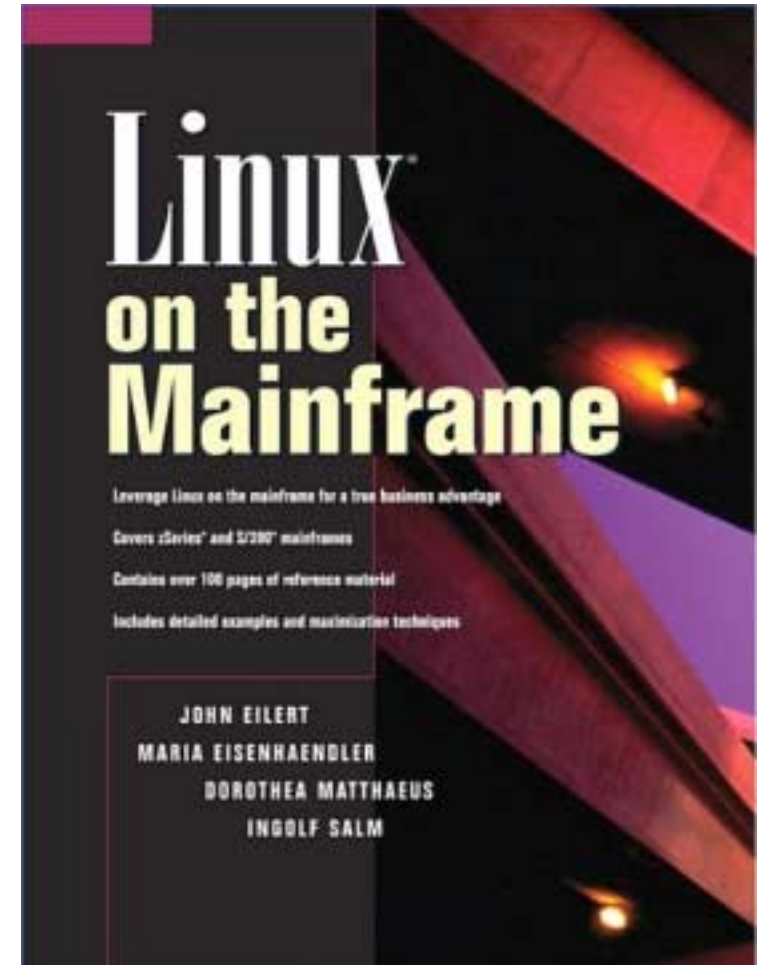
- Open Source Initiative (OSI) maintains a formal definition of OSS recognized by OSS community. OSI has a web site that:
 - ▶ Posts the "Open Source Definition" or "OSD" (<http://www.opensource.org>)
 - ▶ Catalogs OSI-approved licenses that follow the OSD
 - ▶ Most of the popular OSS licenses are listed on this site
- OSD's current criteria (paraphrased):
 - ▶ Free Redistribution: license cannot require royalties and cannot prevent anyone from selling or giving away the code as a component of an aggregate distribution
 - ▶ Source Code: source code must be licensed with the program, or must be made available for no more than a reasonable reproduction cost
 - ▶ Derivative Works: license must allow modifications and derivative works and must allow them to be distributed under the same terms as the original software license
 - ▶ Integrity of the Author's Source Code: license can restrict source code from being distributed in modified form only if it allows the distribution of "patch files"
 - ▶ No Discrimination: license must not discriminate against any person or group
 - ▶ No Discrimination Against Fields of Endeavor: license may not prohibit using the program in any field of endeavor
 - ▶ Distribution of License: rights attached to program cannot require that the program be part of any particular software distribution

1.2 The OSI Open Source Definition - cont.

- OSD's current criteria (paraphrased) cont:
 - ▶ License Must Not Be Specific to a Product: rights attached to program cannot require that the program be part of any particular software distribution
 - ▶ License Must Not Restrict Other Software: license may not place restrictions on any other software that is distributed along with the licensed OSS
 - ▶ License Must Be Technology Neutral: license may not require a particular technology or interface style
- **There are over 100 different licenses in use today that claim to be OSS**
 - ▶ most share features common to the OSD criteria
 - ▶ over 40 licenses have been OSI-approved
 - ▶ **GPL is not** the only Open Source license
 - ▶ Some licenses that provide access to source code in fact contain significant restrictions that differ from open source licensing and, specifically, do not conform to the Open Source Definition criteria
- Each open source license reflects specific objectives of its author(s)
- As with all license agreements, legal advise should be sought from your counsel. Nothing in this presentation is offered as or intended to be legal advice.

Suggested Reading

- Linux on the Mainframe
 - Chapters
 - Linux on the Mainframe - an Introduction
 - Planning for Linux
 - Is Linux on the Mainframe for Me?
 - Making the Most of Linux on the Mainframe
 - Running Applications
 - ▶ Publisher: Prentice Hall
 - ▶ ISBN: 013101415
- IBM Redbooks
 - ▶ ibm.com/redbooks



IBM Hospitality Night

- TPF Programmer's Challenge
- TPF Toolkit for WebSphere
- SOAP and XML processing
- TPF Information Center
- TPF Web
- Finding data on the Web (APARs)
- TPF Operations Server
- and more !

Wednesday Morning Education

- TPF Programmer's Challenge - IBM's Solution

IBM Presentations

- TPF Toolkit for WebSphere
- TPF ITRRs
- TPF Copy Services
- TPF Operations Server
- OSA-ICC
- SOAP for TPF
- Web Services for TPF
- Compiler Update

IBM Presentations ,, ,, Up Next!!!!

- TPF Toolkit for WebSphere V2.0
 - ▶ Peter Nicholls
- TPF Programmer's Challenge
 - ▶ Colette Manoni

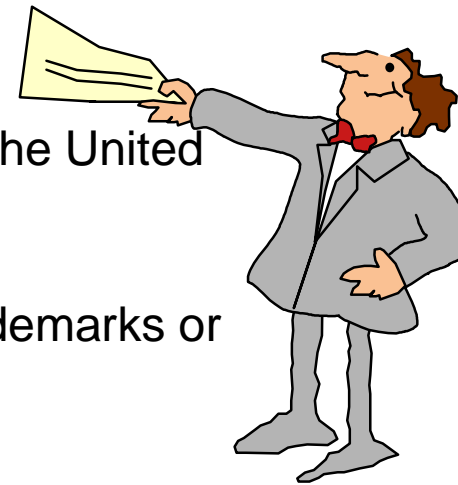
A word from our sponsor

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

IBM, Websphere, MQSeries, zOS, z/VM and zSeries are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.



- Product Information:
 - ▶ <http://www.ibm.com/tpf>