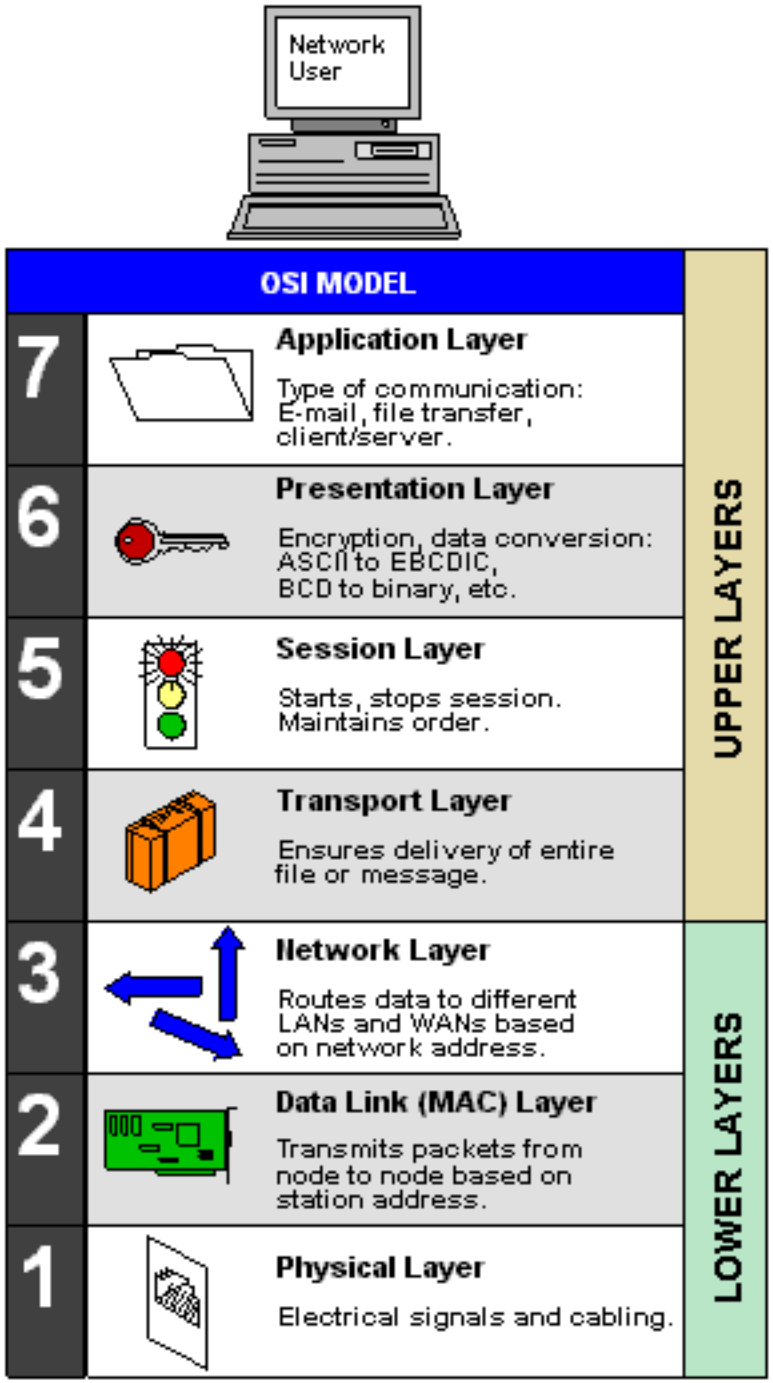
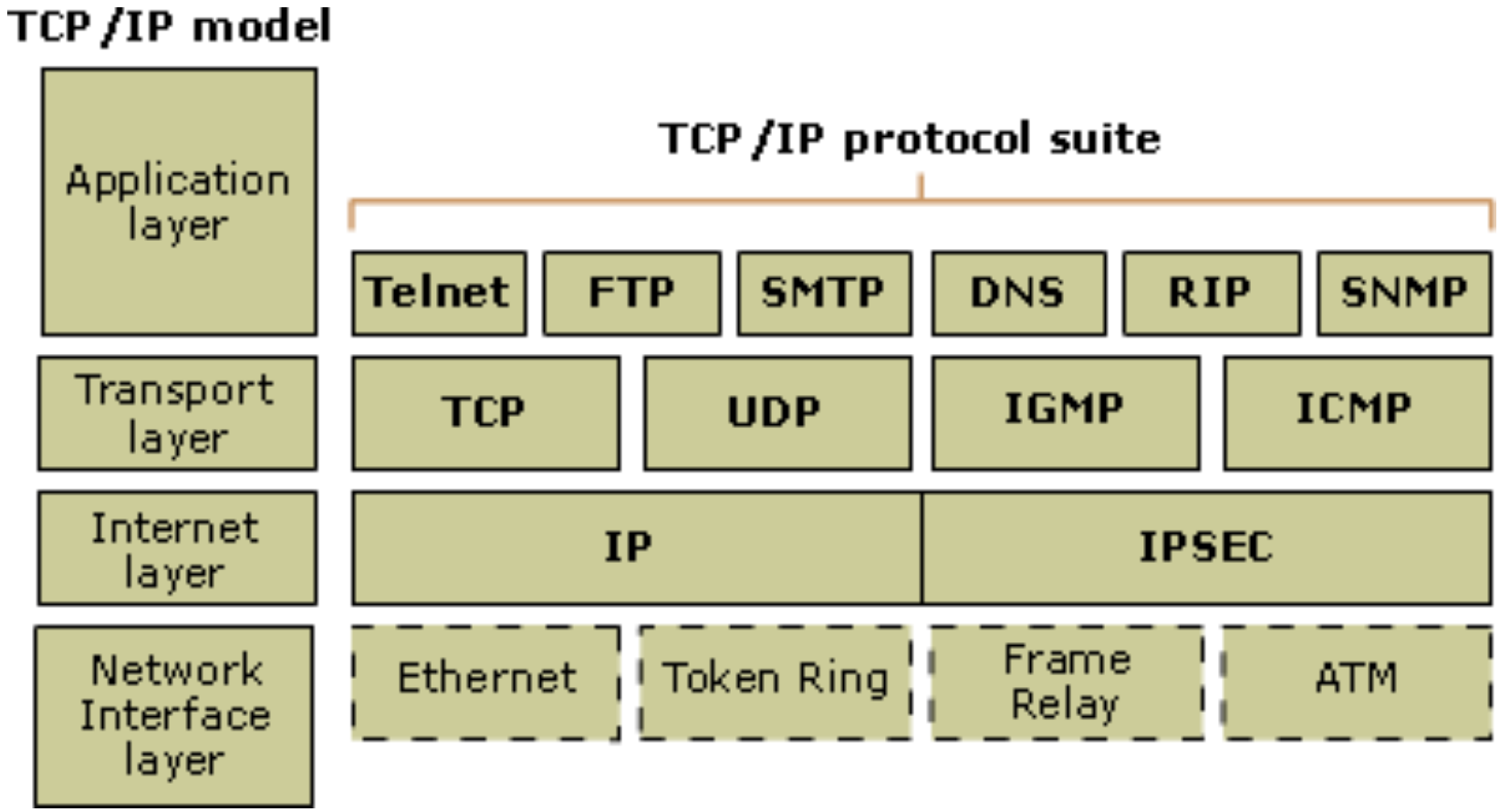
Architectural Patterns: Examples of use

Source: Internet

The OSI reference model



TCP/IP Model


TCP /IP model
Application layer
Transport layer
Internet layer
Network Interface layer
TCP /IP protocol suite
Telnet
FTP
SMTP
DNS
RIP
SNMP
TCP
UDP
IGMP
ICMP
IP
IPSEC
Ethernet
Token Ring
Frame Relay
ATM

A 3-tier application

Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

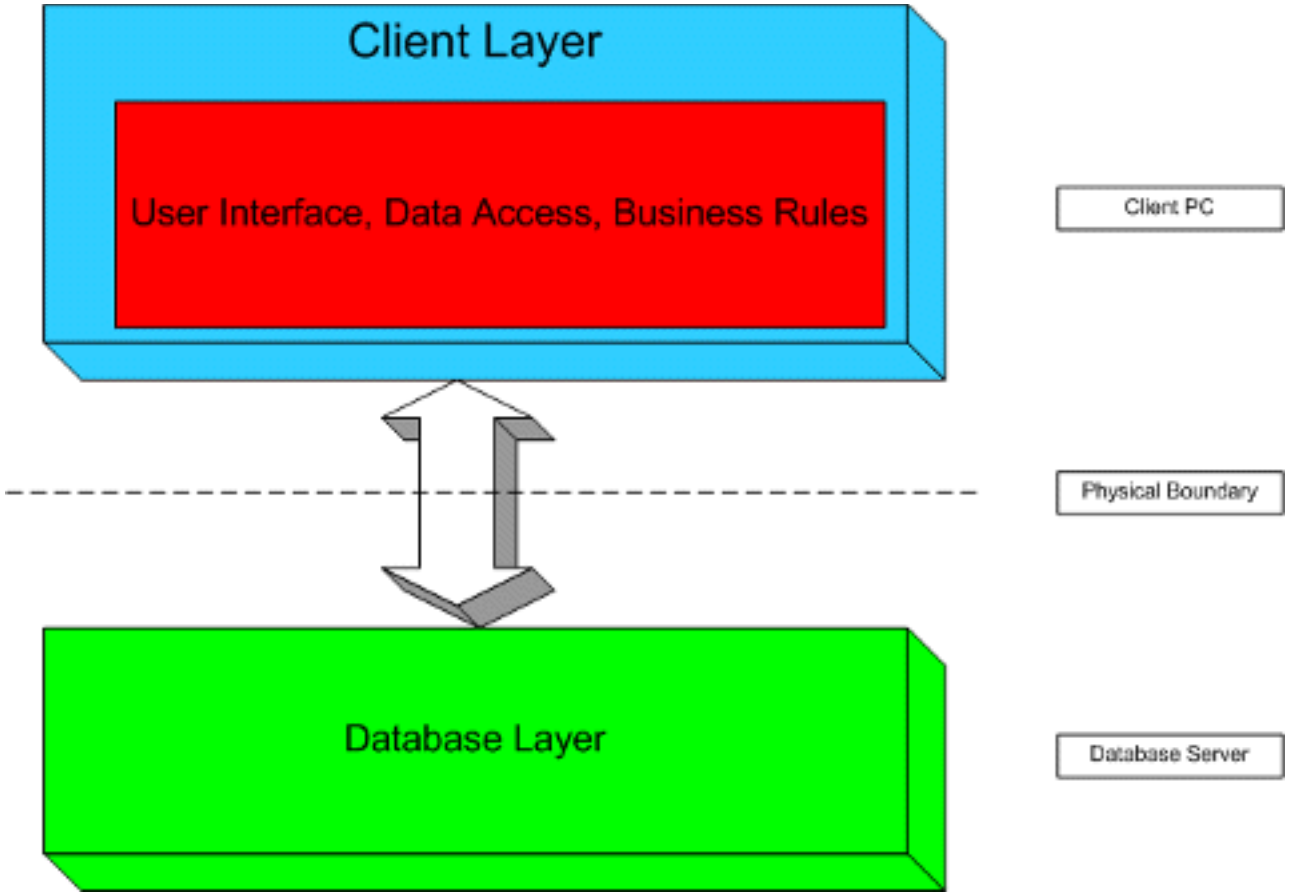
Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

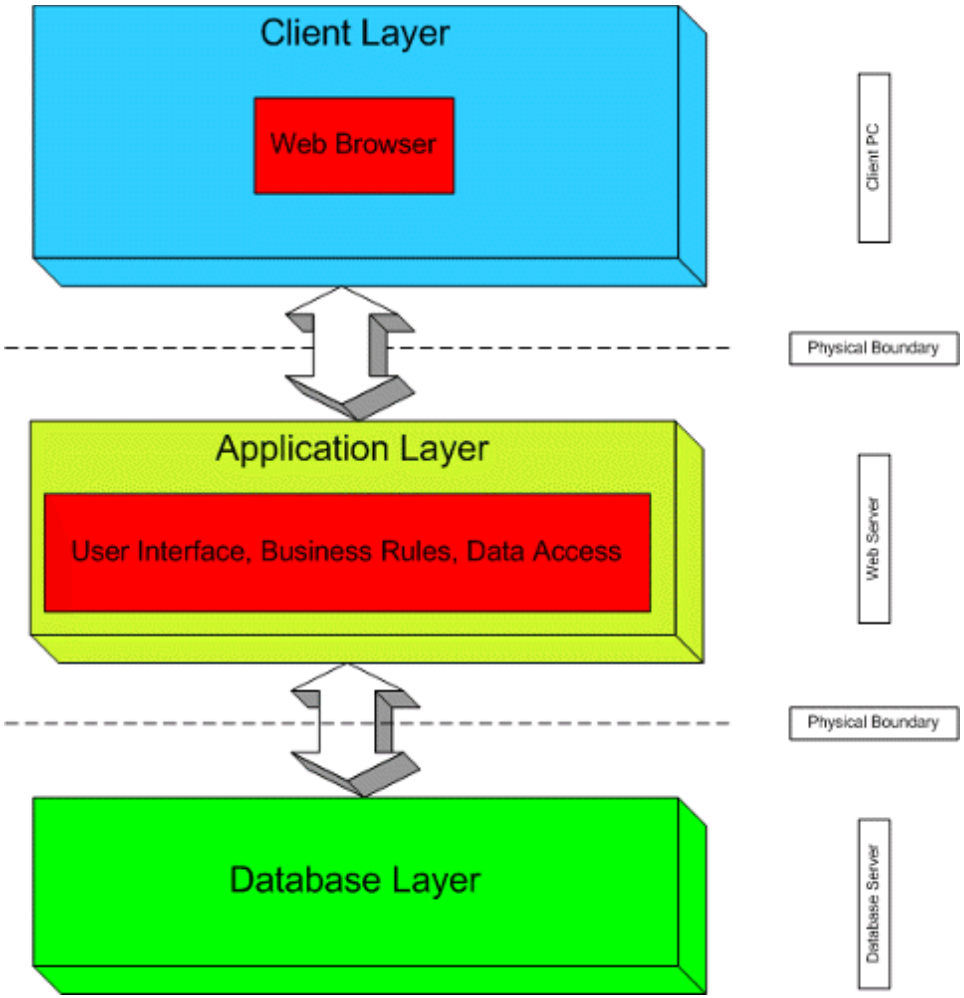
Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

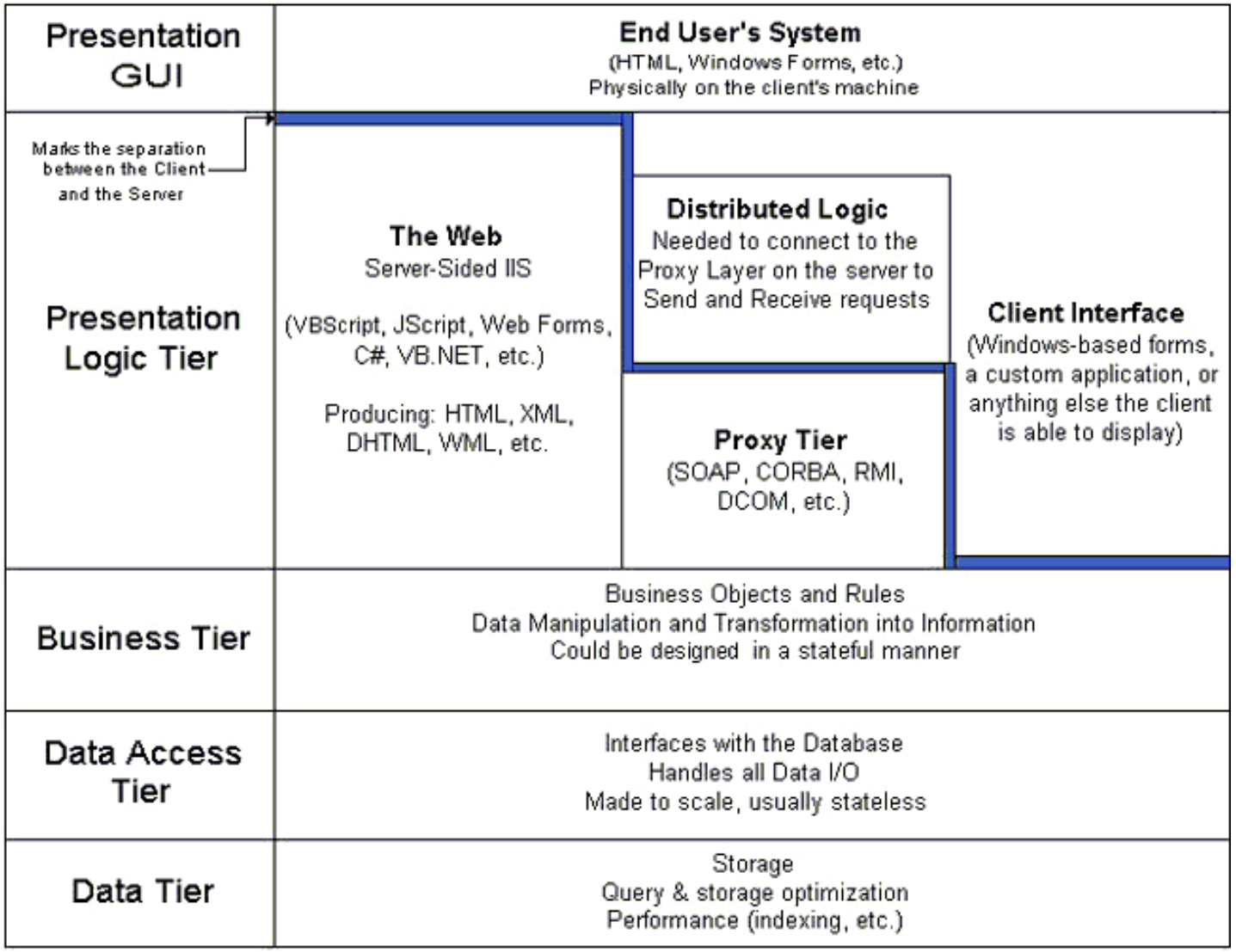
2-tier client-server architecture



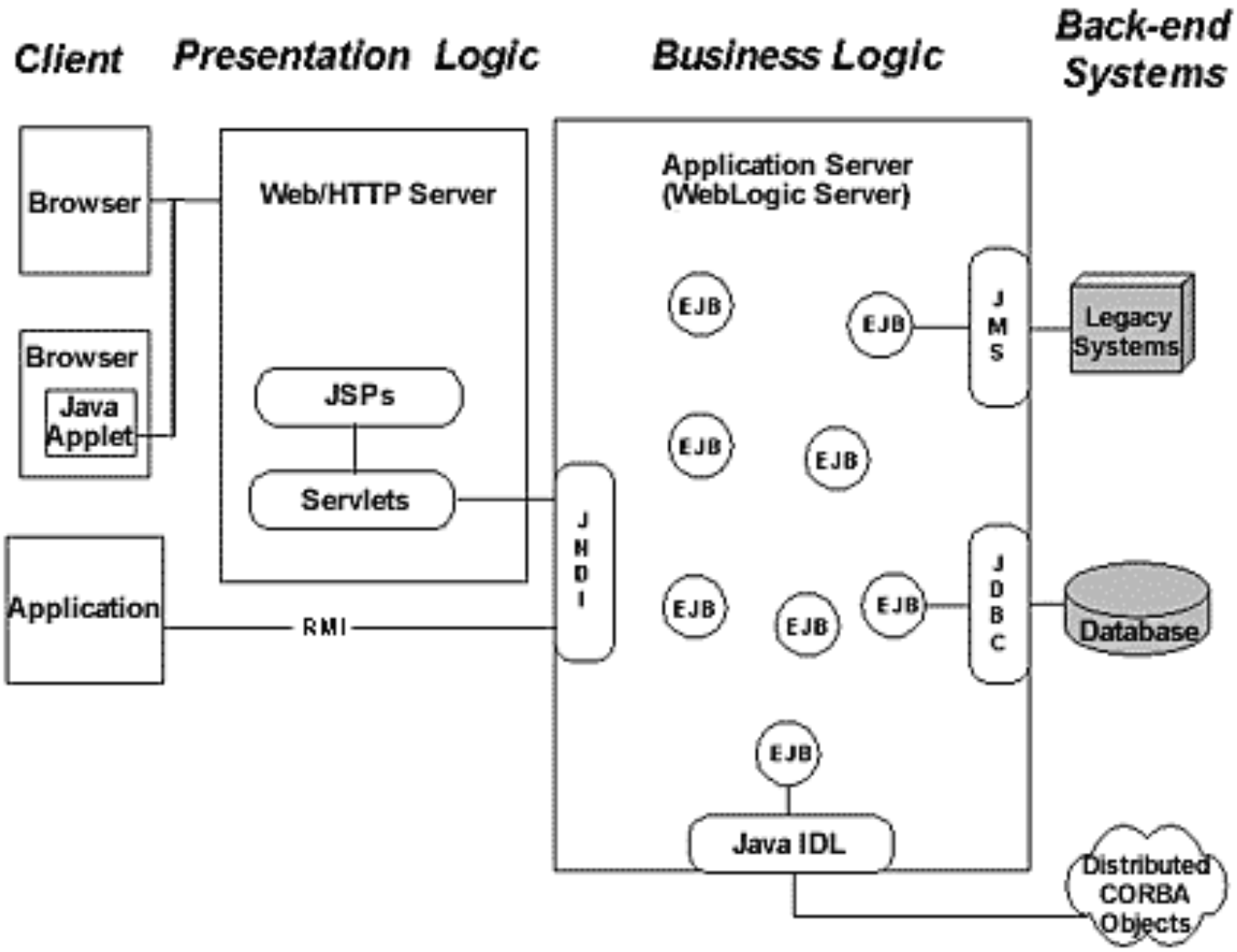
3-tier architecture



N-tier Application Architecture



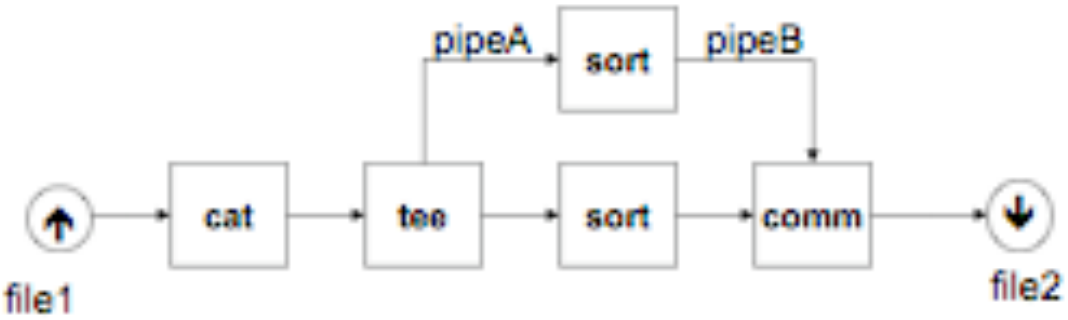
N-tier architectures with J2EE technology



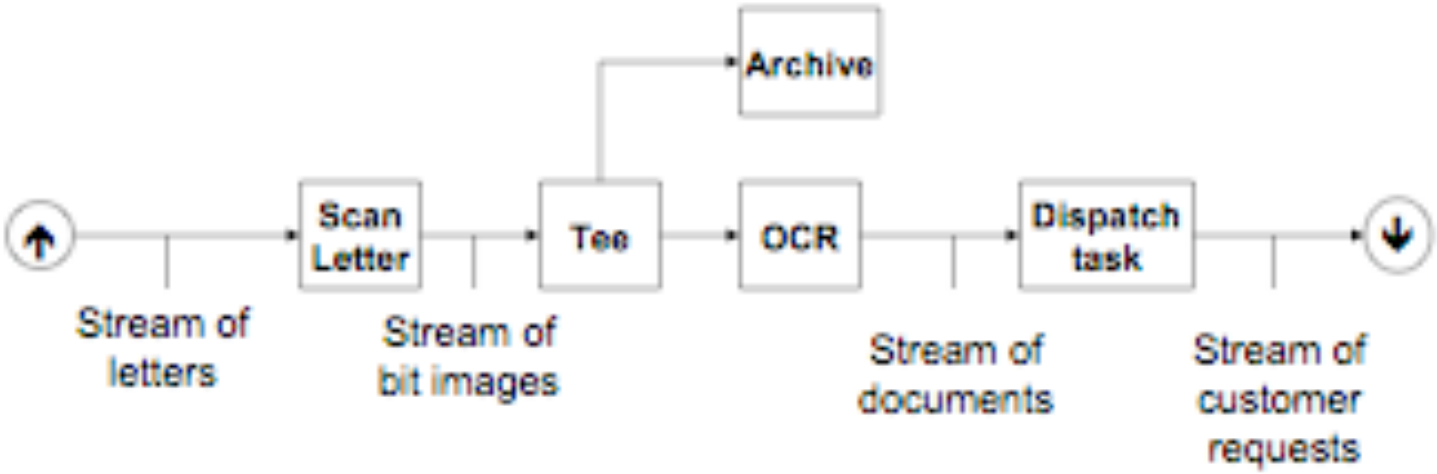
Unix Tee and Join

Task: Print a sorted list of words that occur more than once

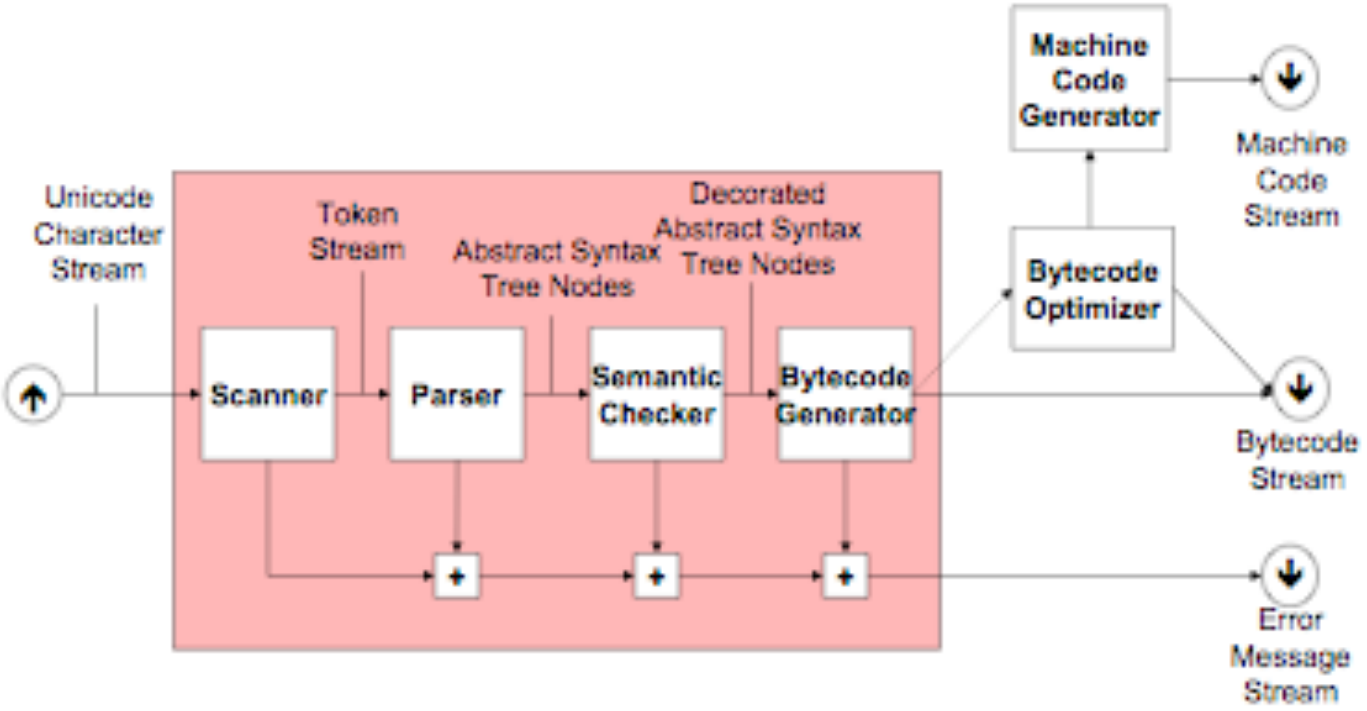
```
mknod pipeA p

mknod pipeB p

sort pipeA > pipeB &
cat file1 | tee pipeA | sort -u | comm -13 - pipeB > file2
```



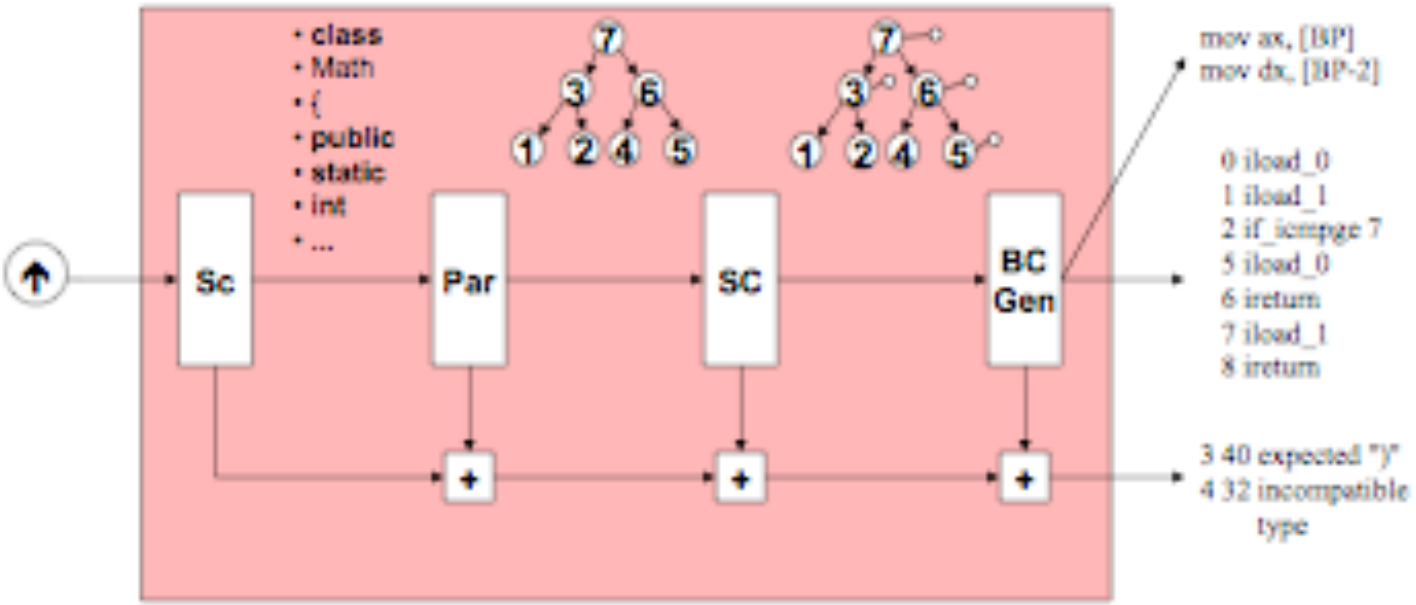
Document management



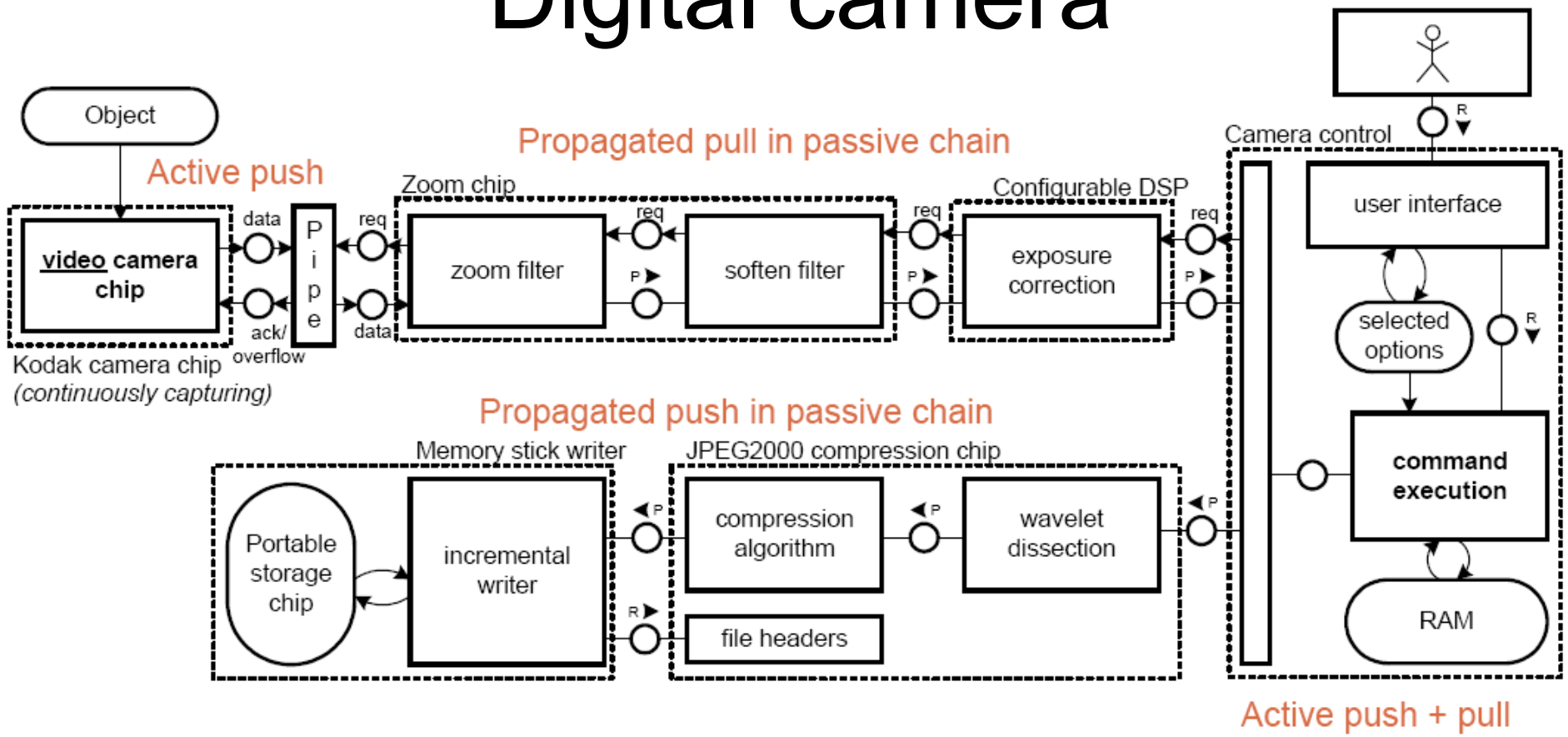
A compiler


Unicode
Character
Stream
Token
Stream
Abstract Syntax
Tree Nodes
Decorated
Abstract Syntax
Tree Nodes
Scanner
Parser
Semantic
Checker
Bytecode
Generator
Machine
Code
Generator
Bytecode
Optimizer
Machine
Code
Stream
Bytecode
Stream
Error
Message
Stream

Another compiler



Digital camera



Apache web server

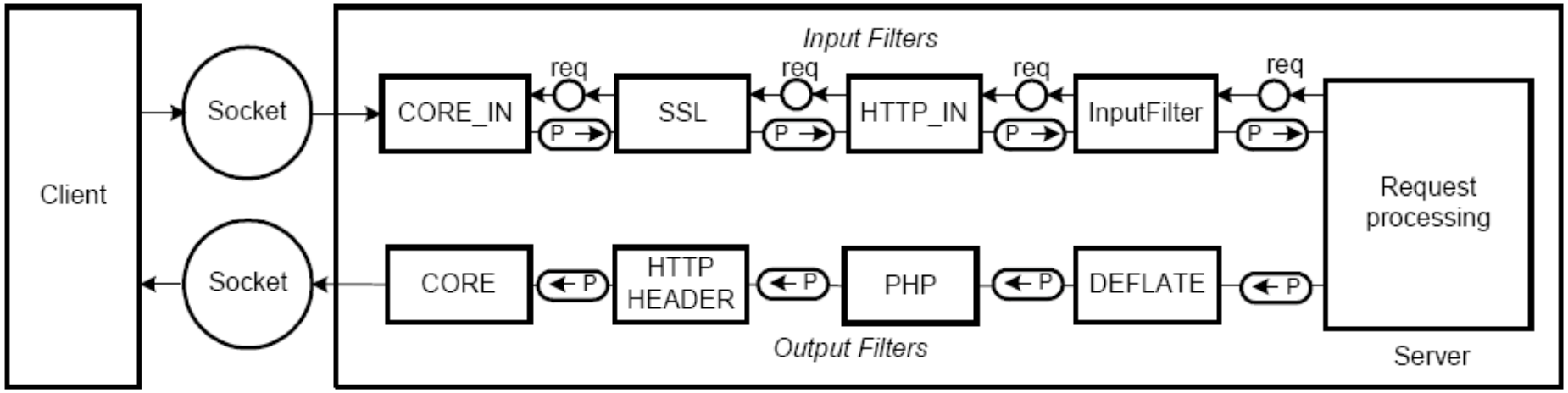
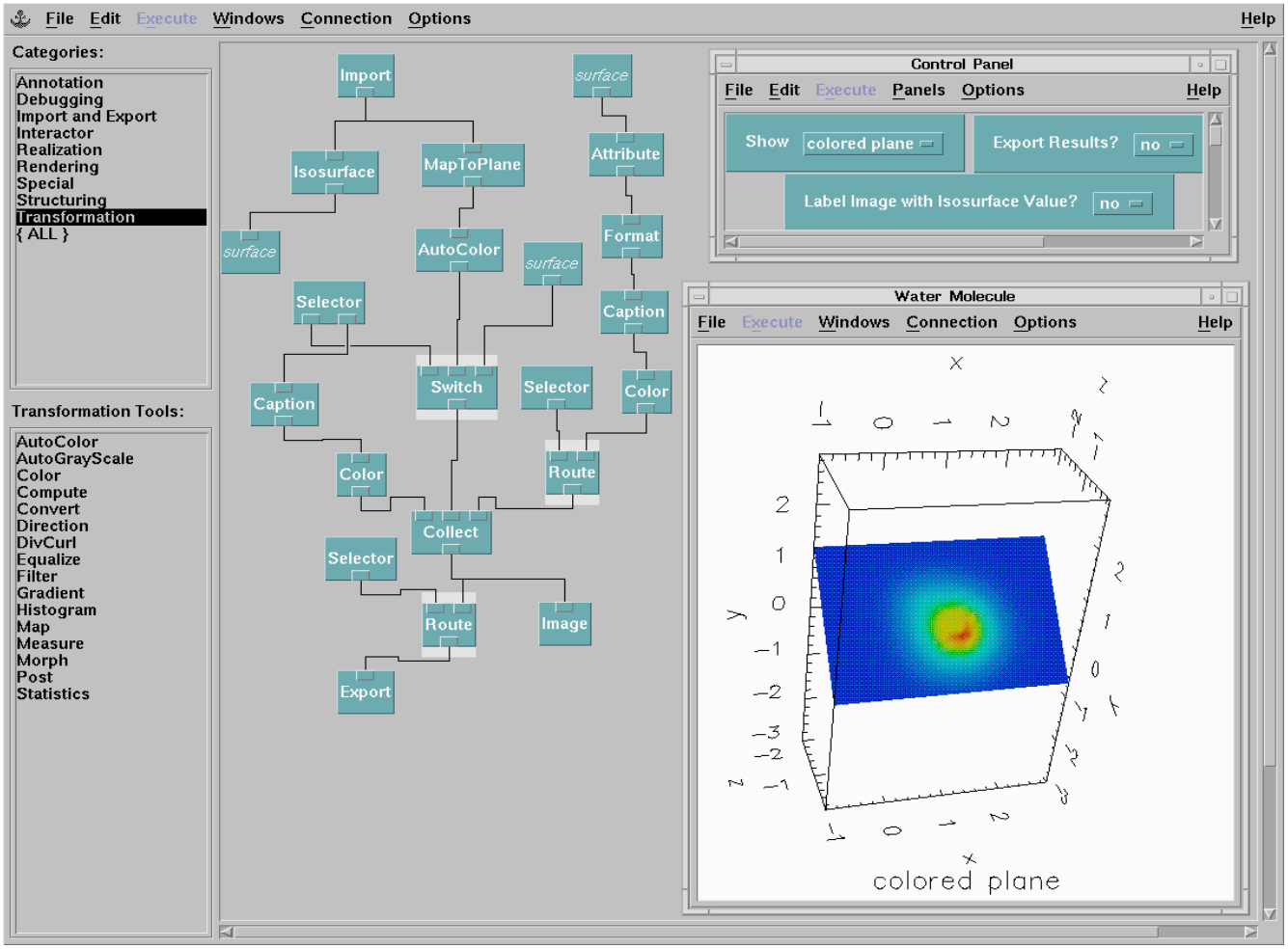
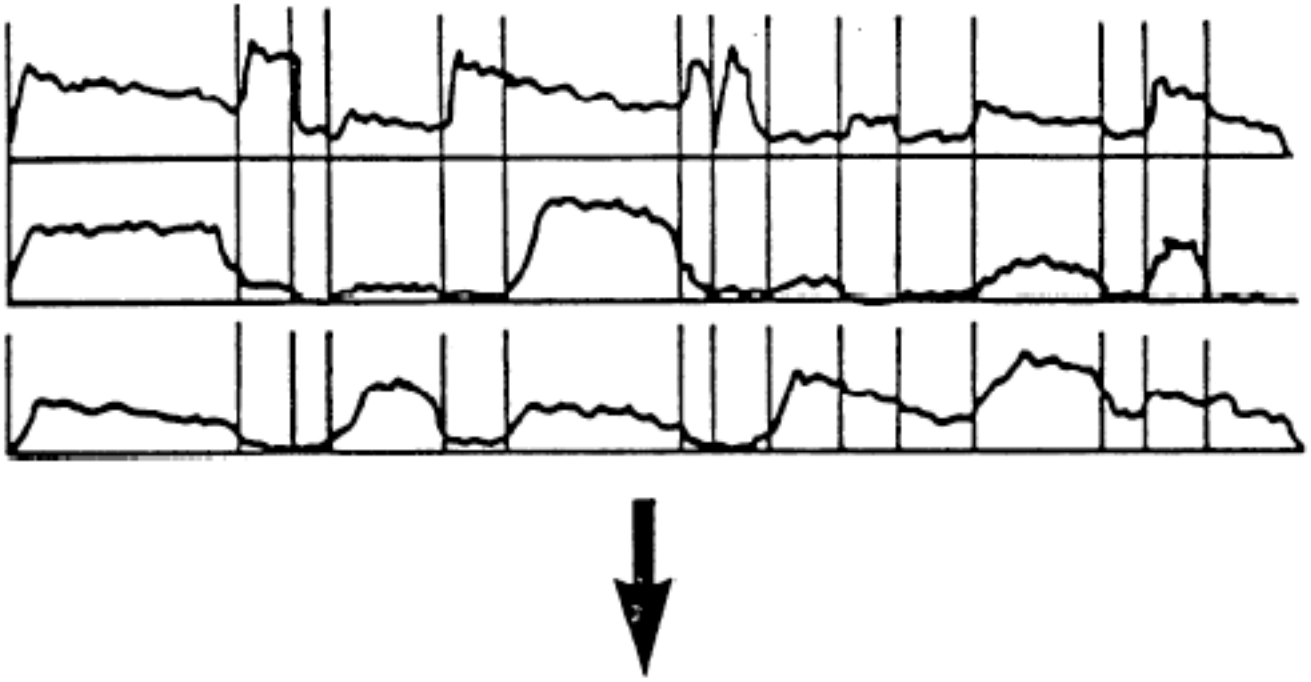


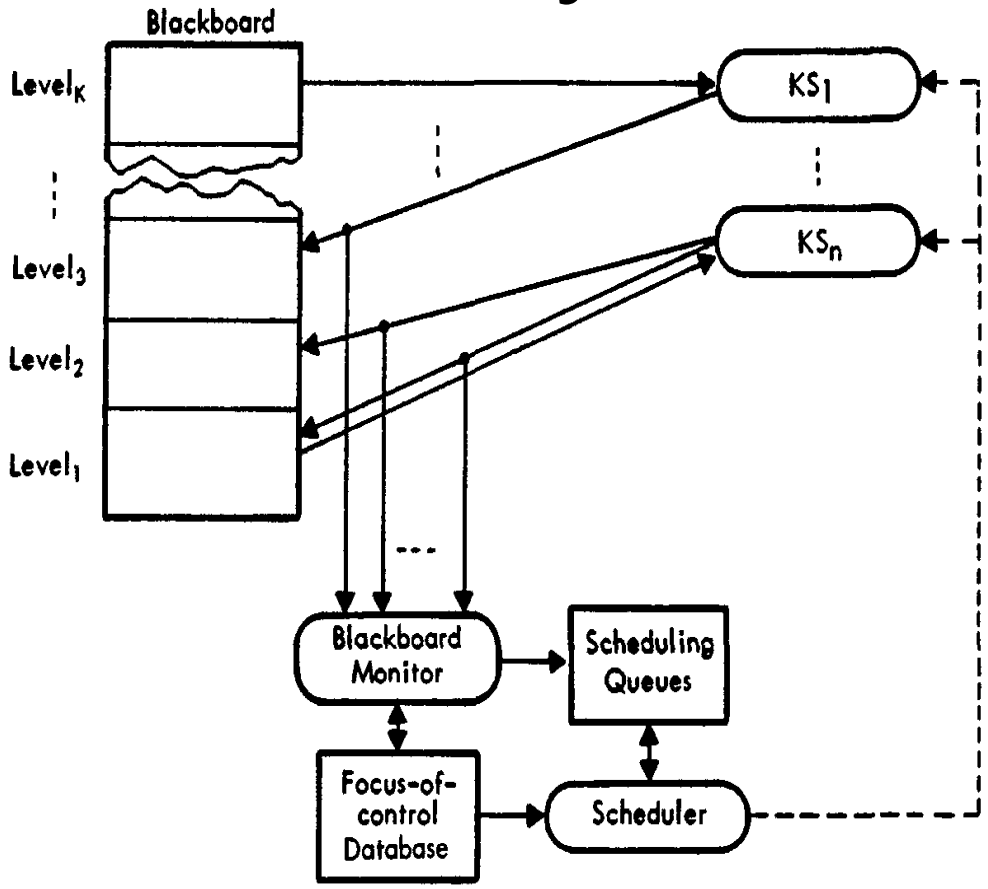
Image processing



Hearsay-II task



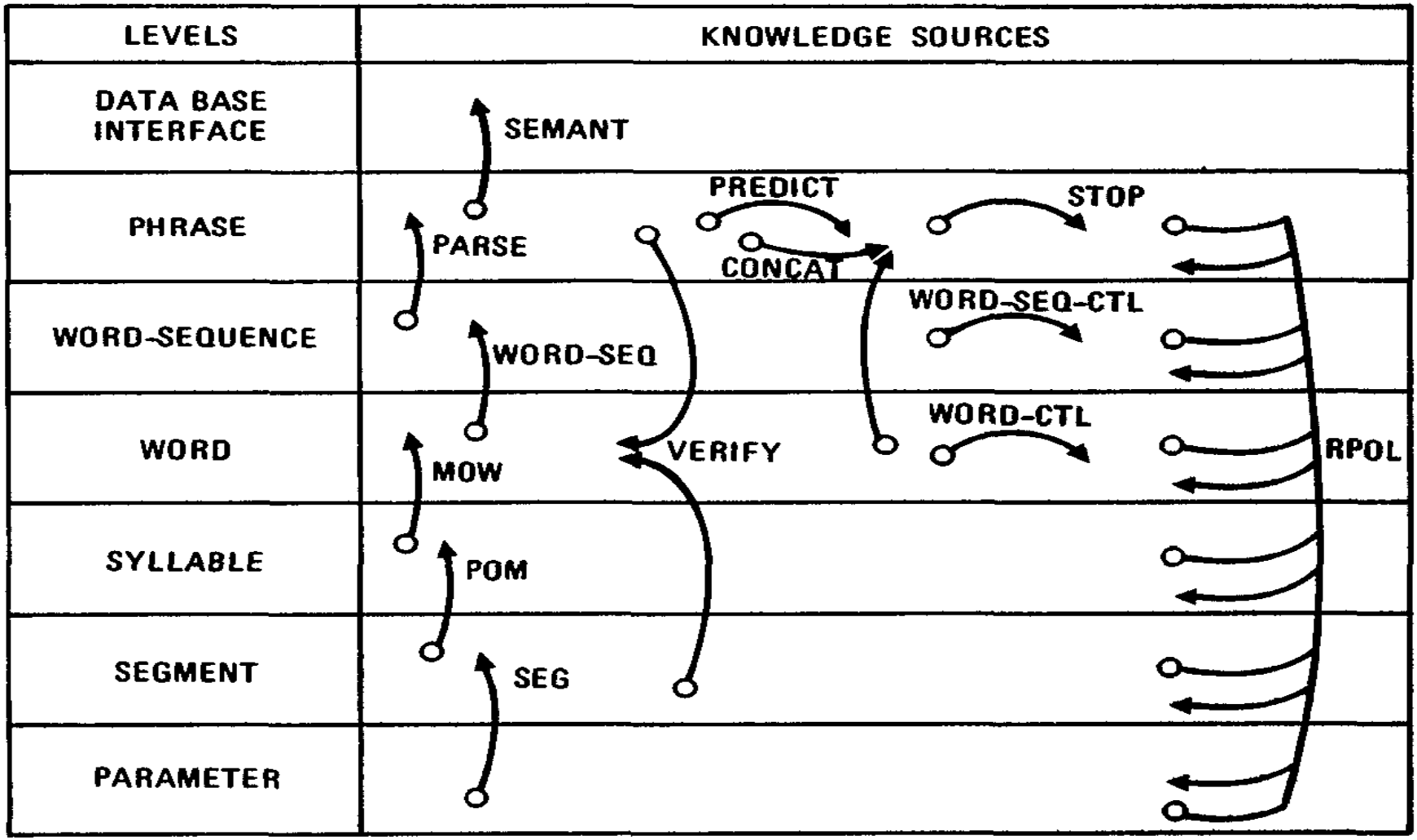
Hearsay-II architecture



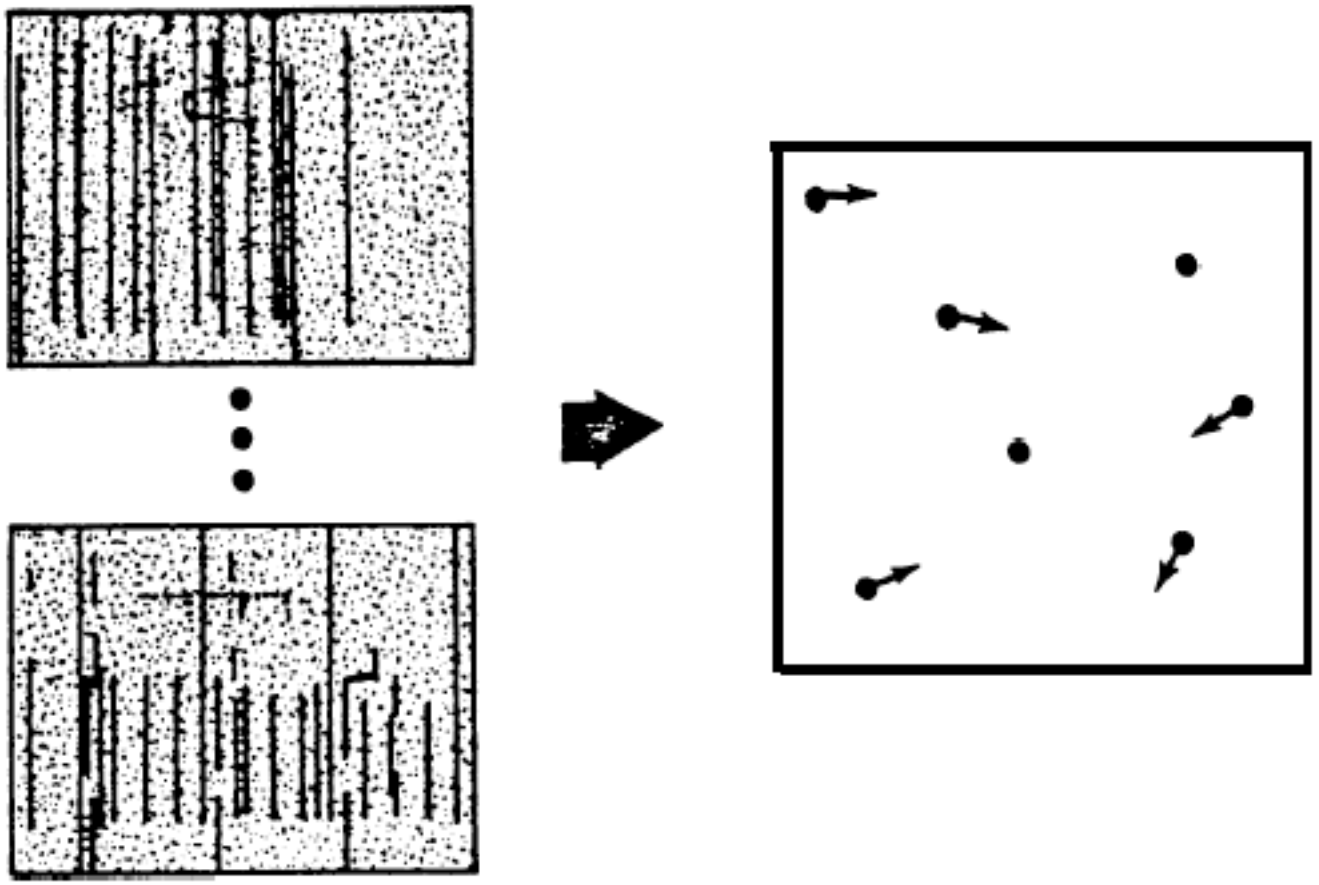
Hearsay Control

- Data driven
- Asynchronous
- Opportunistic
- Top-down AND bottom-up
- Focus of attention

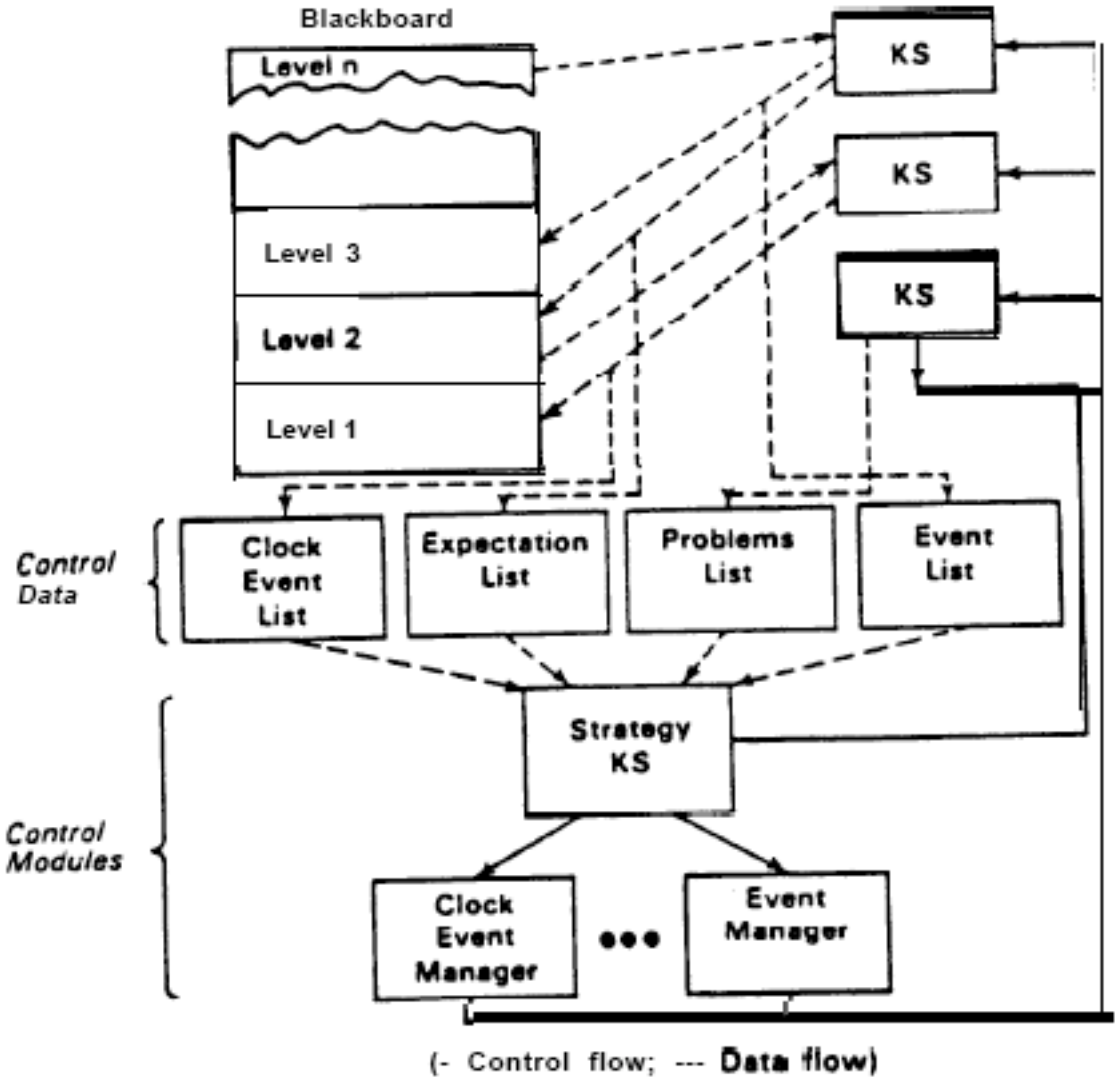
Levels and knowledge sources in Hearsay-II



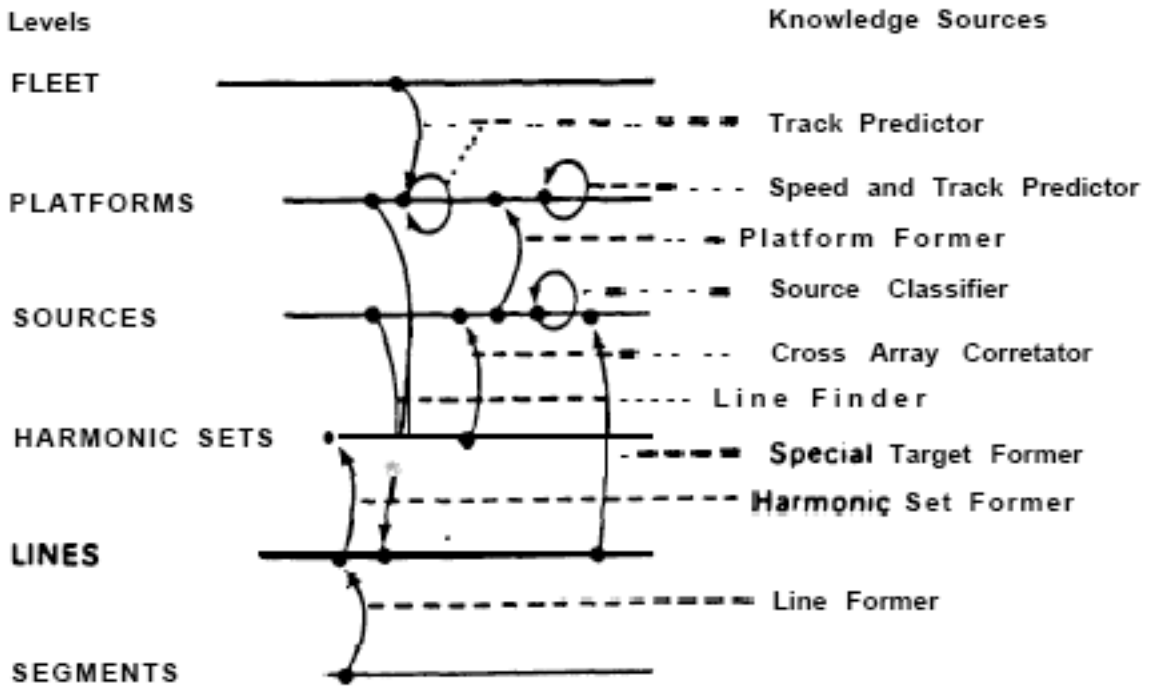
Hasp task



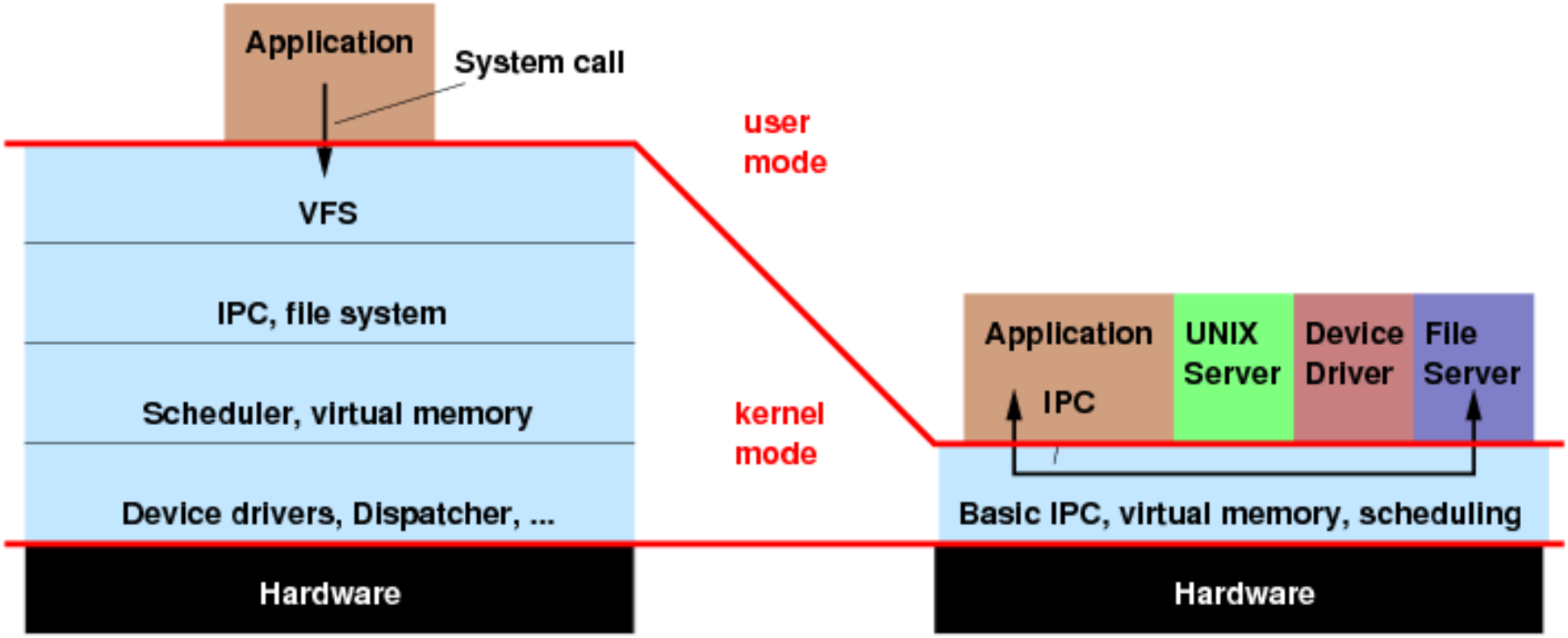
Hasp architecture



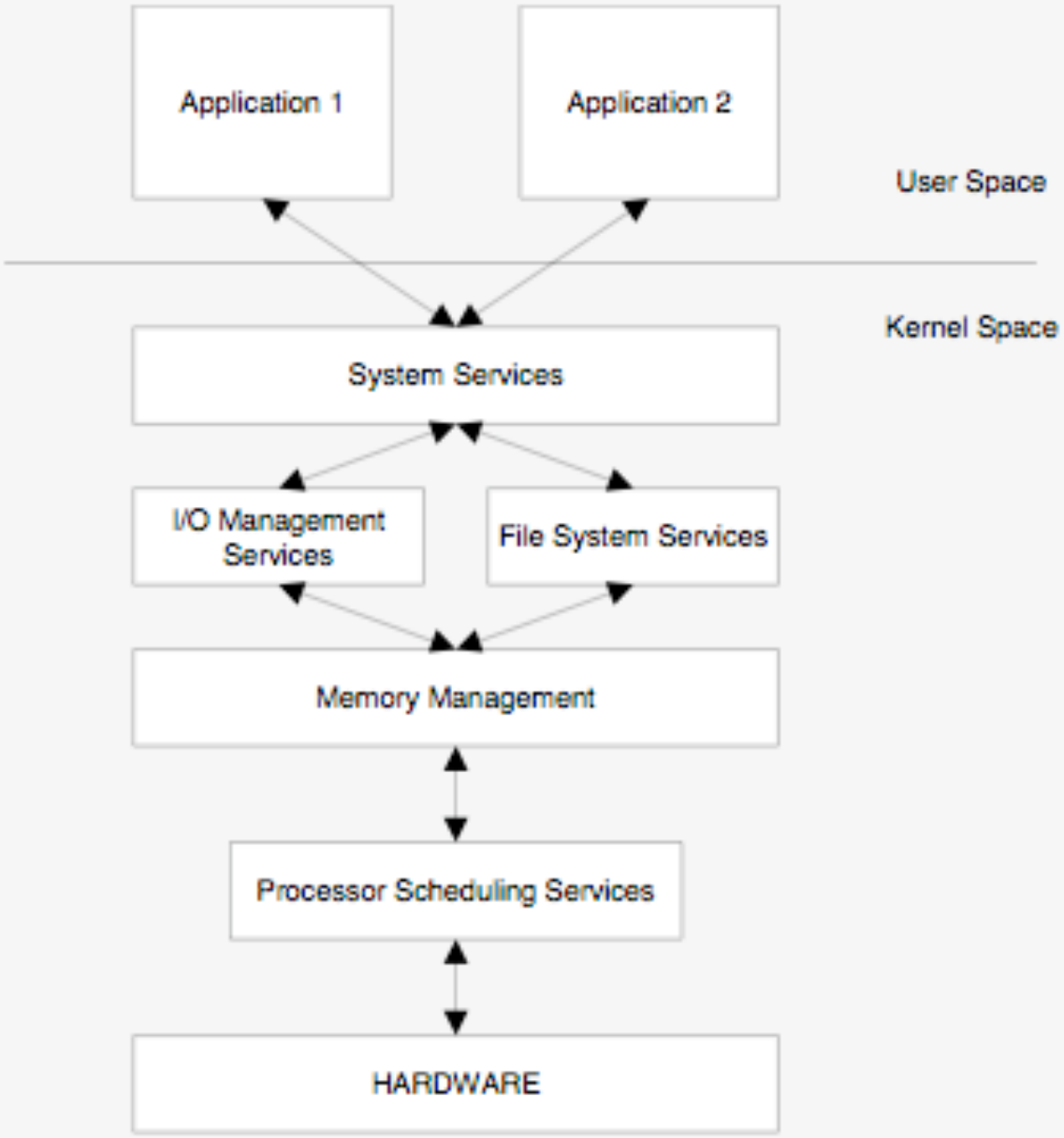
Levels and knowledge sources in Hasp

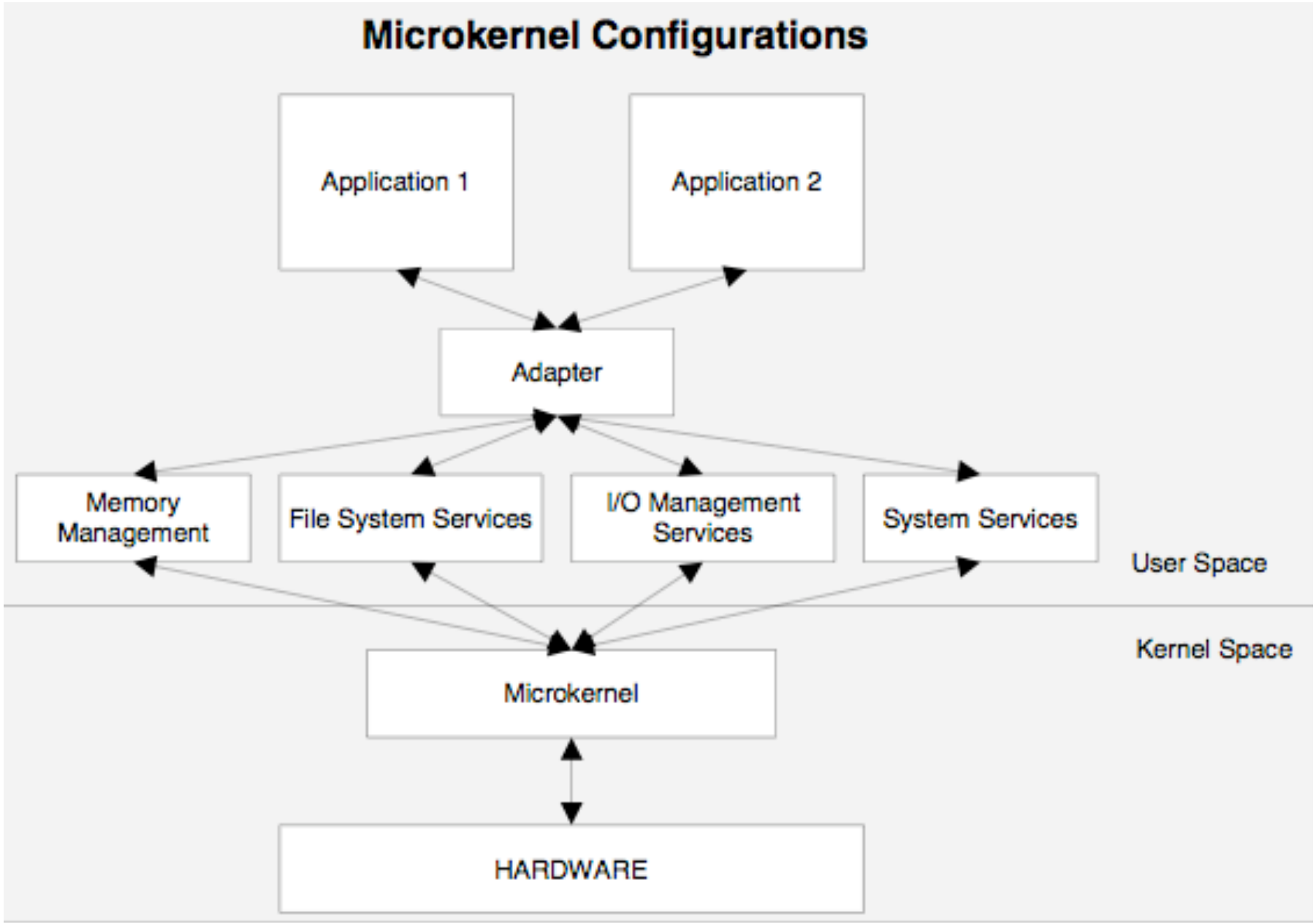


Monolitic versus microkernel OS

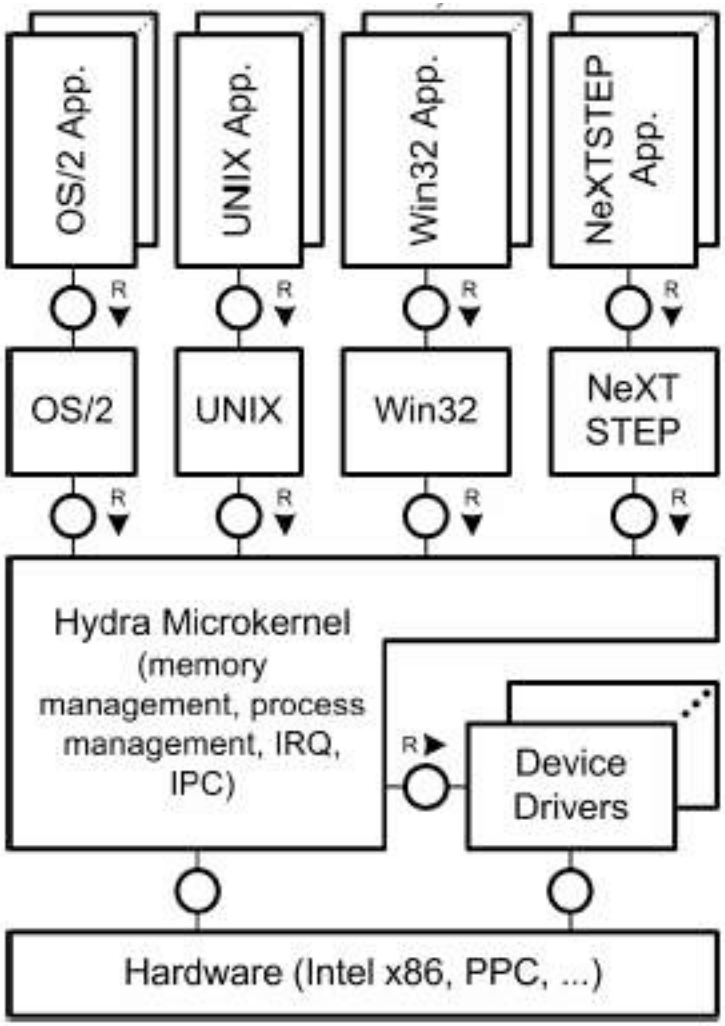


Traditional Kernel Configurations

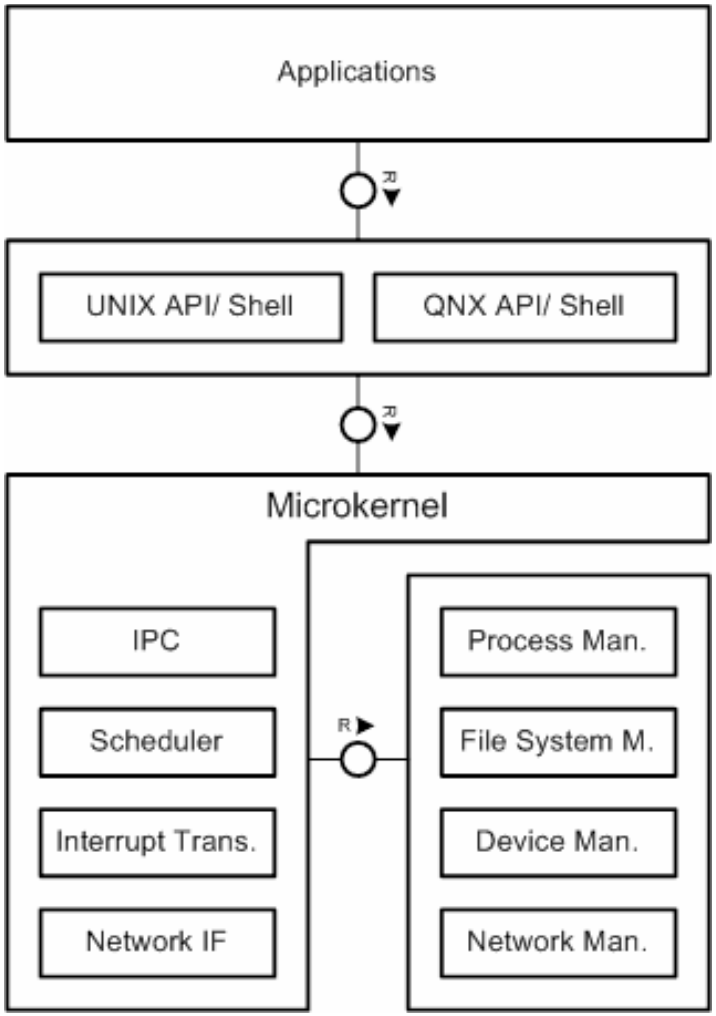



Microkernel Configurations
Application 1
Application 2
Adapter
Memory Management
File System Services
I/O Management Services
System Services
User Space
Kernel Space
Microkernel
HARDWARE

HYDRA a microkernel OS

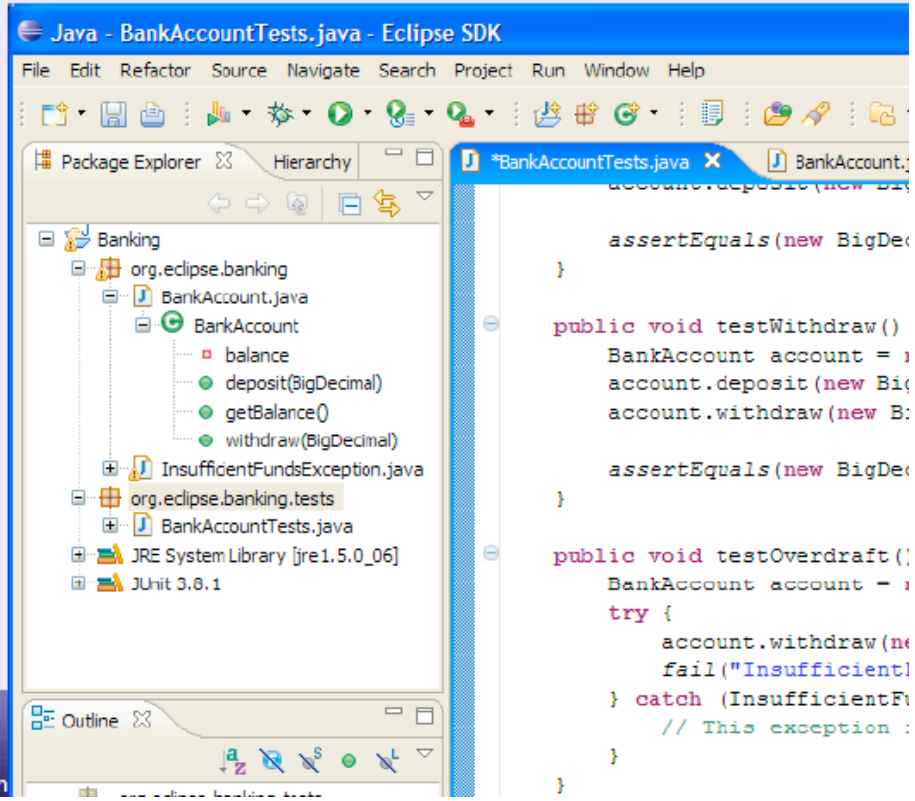


The QNX real time OS



Eclipse is a Java IDE

- Language-aware editors, views, …
- Refactoring support
- Integrated unit testing and debugging
- Incremental compilation and build
- Team development support

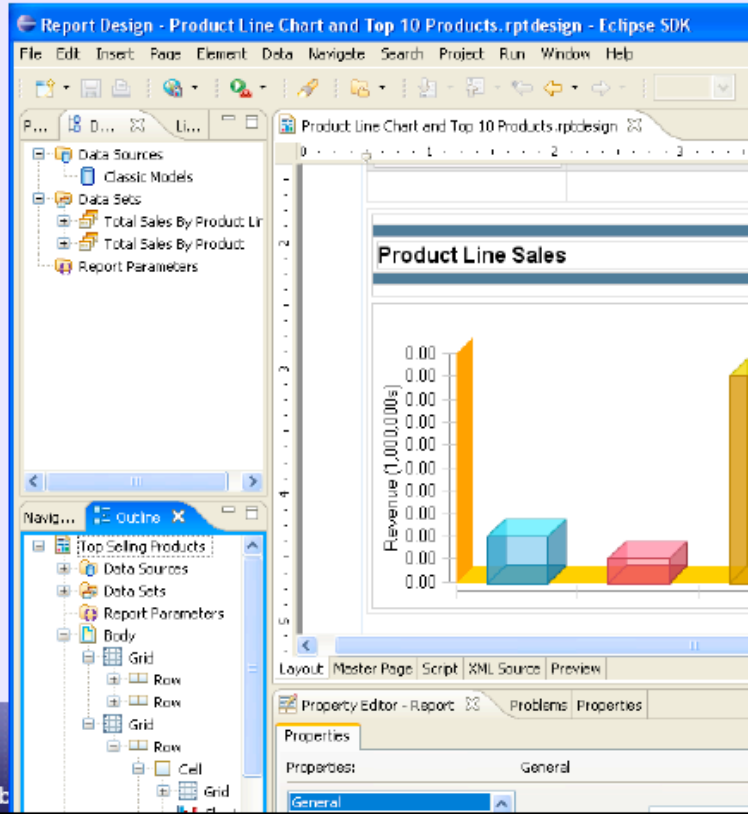


Eclipse is an IDE Framework

- Eclipse + JDT = Java IDE
  - First class framework for Java, language aware editor, incremental build, integrated debugging, ...
- Eclipse + CDT = C/C++ IDE
  - First class framework for C/C++, language aware editor, refactoring, search
- Eclipse + PDT = PHP IDE
- Eclipse + JDT + CDT + PDT = Java, C/C++, PHP IDE
  - Ruby, TCL, JavaScript, ...

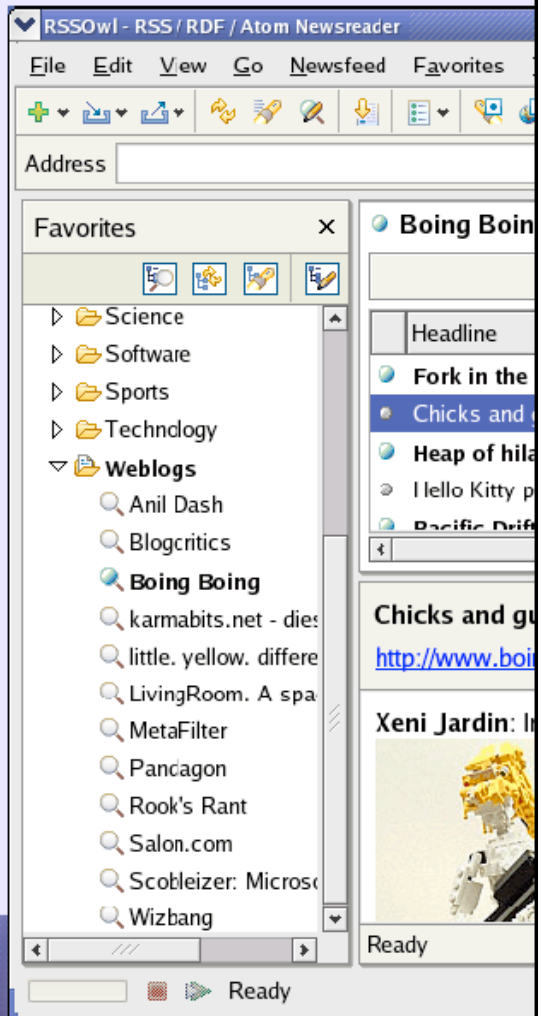
Eclipse is a Tools Framework

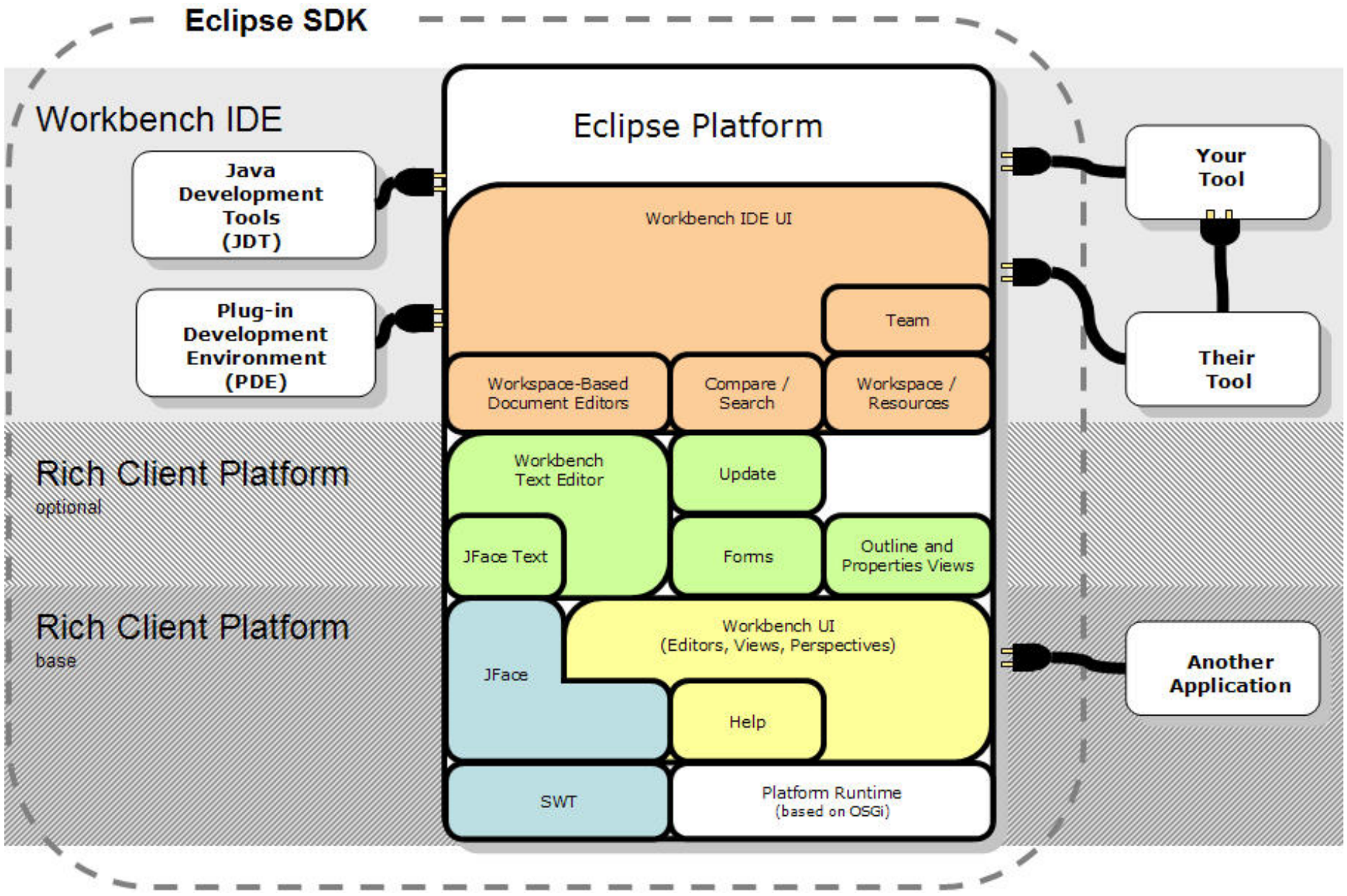
- Plug-ins make Eclipse whatever you need it to be
- Platform of frameworks and exemplary tools
- Tools extend the platform using bundles/plug-ins
  - Business Intelligence and Reporting Tools, Web Tools Project, Data Tools Project, Eclipse Modeling Framework



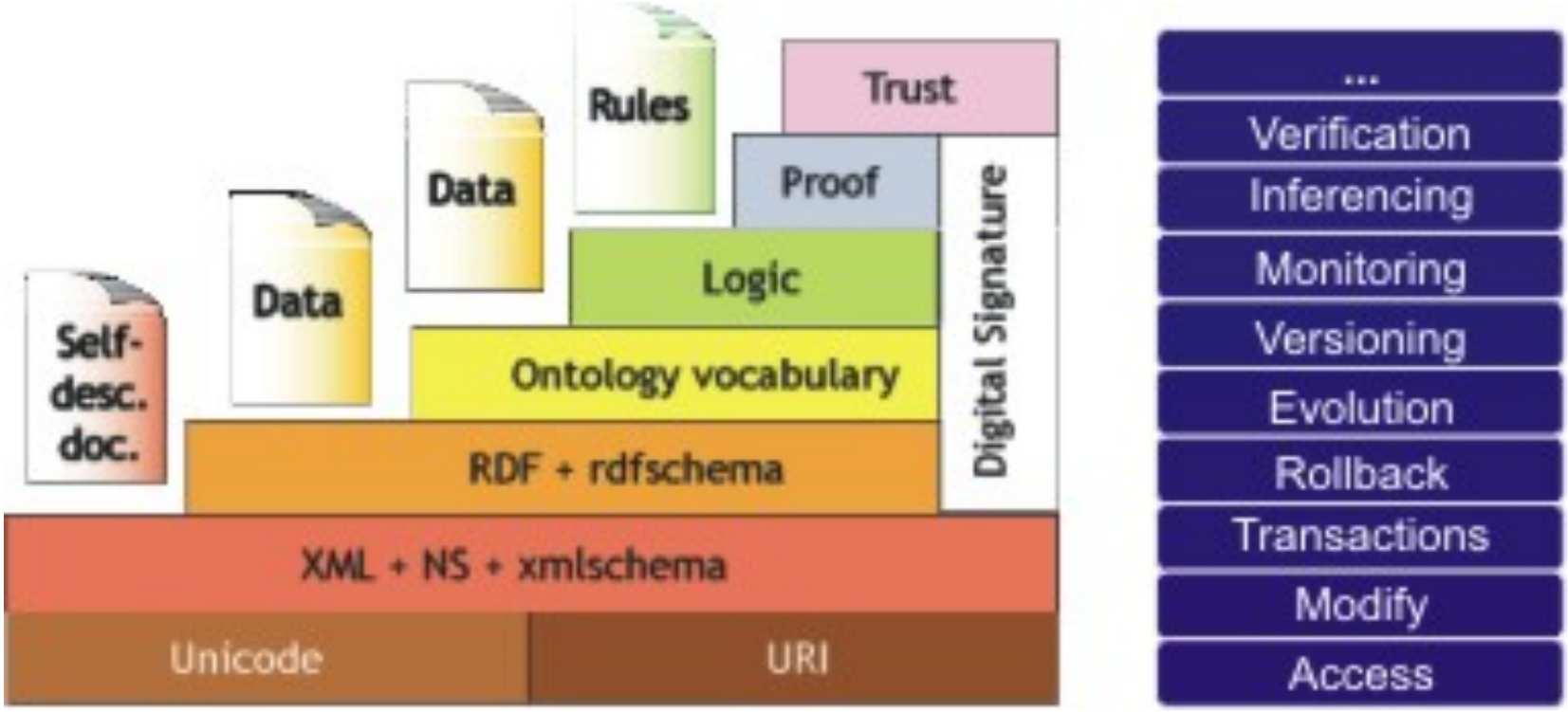
Eclipse is a Application Framework

- Remove the IDE elements; you're left with a general-purpose application framework
  - Linux, Windows, Mac OSX, UNIX, embedded
  - Rich widget set, graphics
  - Native-OS integration (drag and drop, OLE/XPCOM integration)
- A platform for rich clients

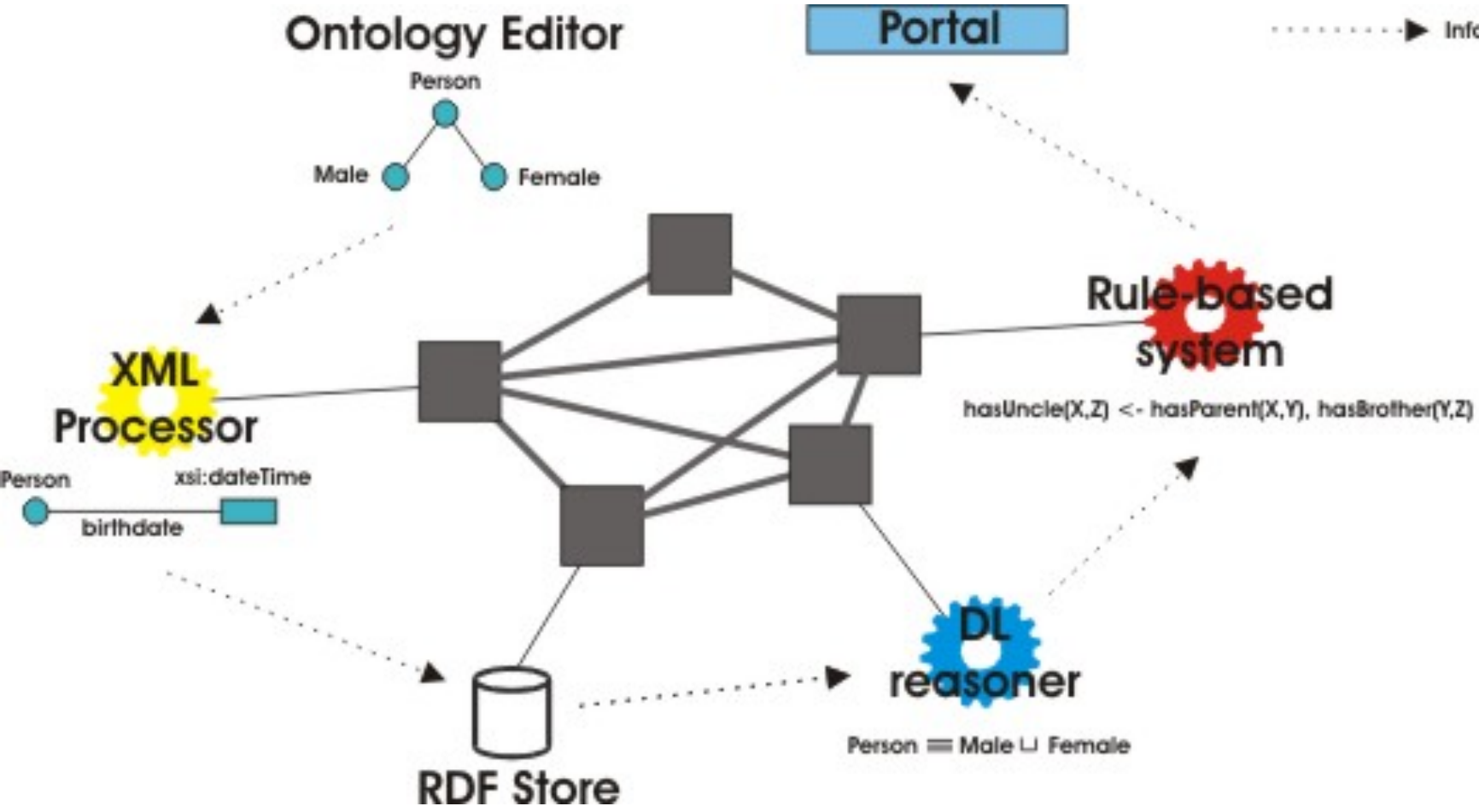



Eclipse SDK
Workbench IDE
Java Development Tools (JDT)
Plug-in Development Environment (PDE)
Eclipse Platform
Workbench IDE UI
Team
Workspace-Based Document Editors
Compare / Search
Workspace / Resources
Workbench Text Editor
Update
JFace Text
Forms
Outline and Properties Views
Rich Client Platform
optional
Rich Client Platform
base
JFace
Workbench UI (Editors, Views, Perspectives)
Help
SWT
Platform Runtime (based on OSGi)
Your Tool
Their Tool
Another Application

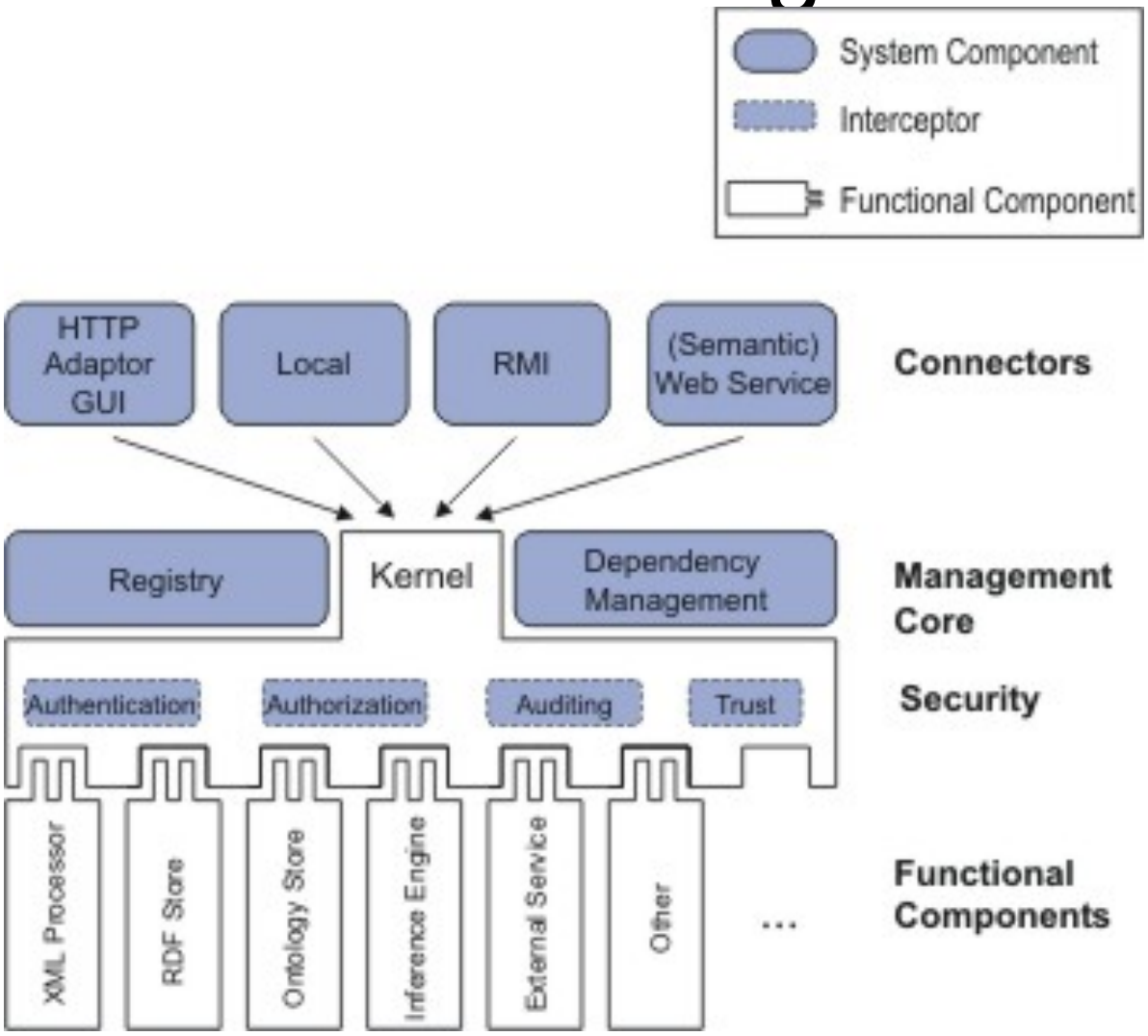
The semantic web



An application using ontology stuff



The KAON server architecture - A Semantic Web Management System



MicrowaveController (SPL)

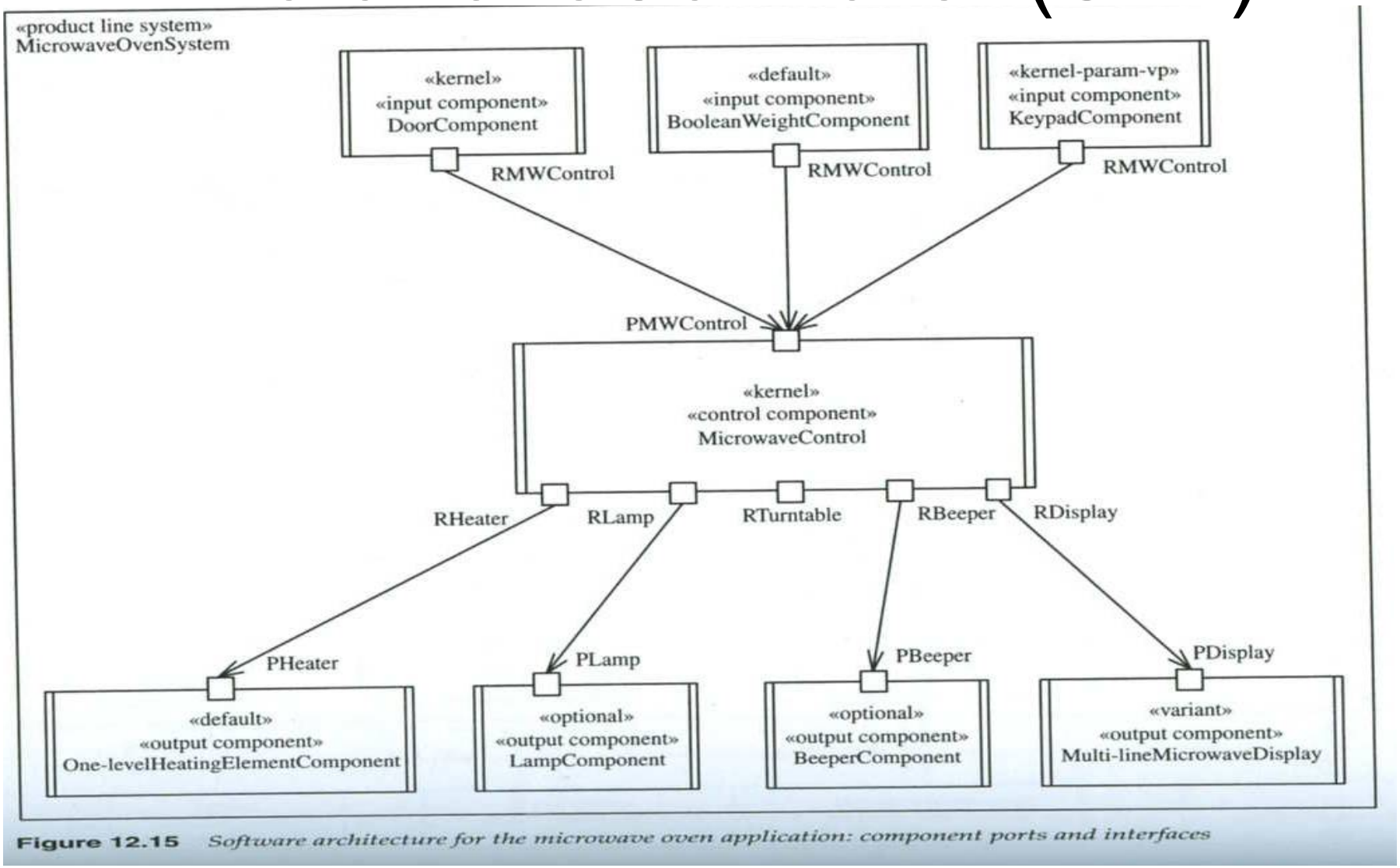


Figure 12.15 *Software architecture for the microwave oven application: component ports and interfaces*

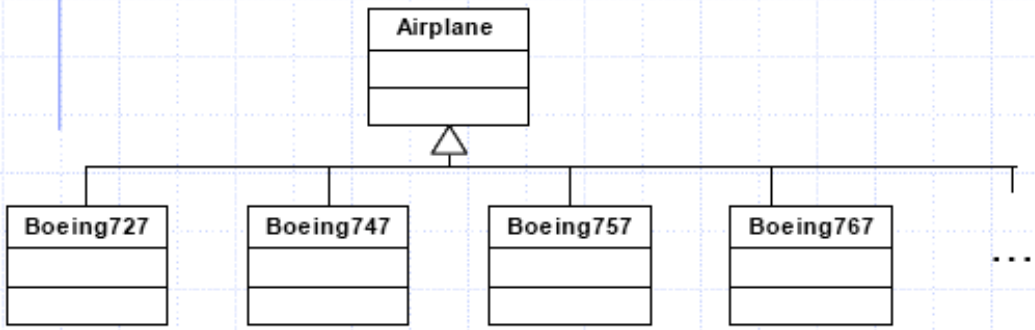
Adaptive Object-Model
(Active|Dynamic Object-Model)

- An ADAPTIVE OBJECT-MODEL is an object model that provides "meta" information about the domain so that it can be changed at runtime
  - explicit object model that it interprets at run-time
  - change the object model, system changes its behavior
- ADAPTIVE OBJECT-MODELS usually arise from domain-specific frameworks
- Business rules are stored as descriptive (meta) information in ADAPTIVE OBJECT-MODELS
- Sometimes called a "reflective architecture" or a "meta-architecture ".

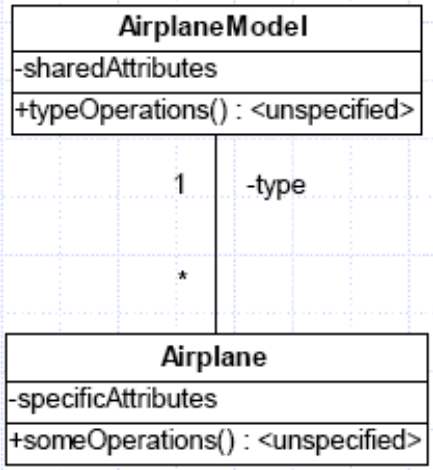
Type-Object

PLoPD3 - Johnson and Woolf

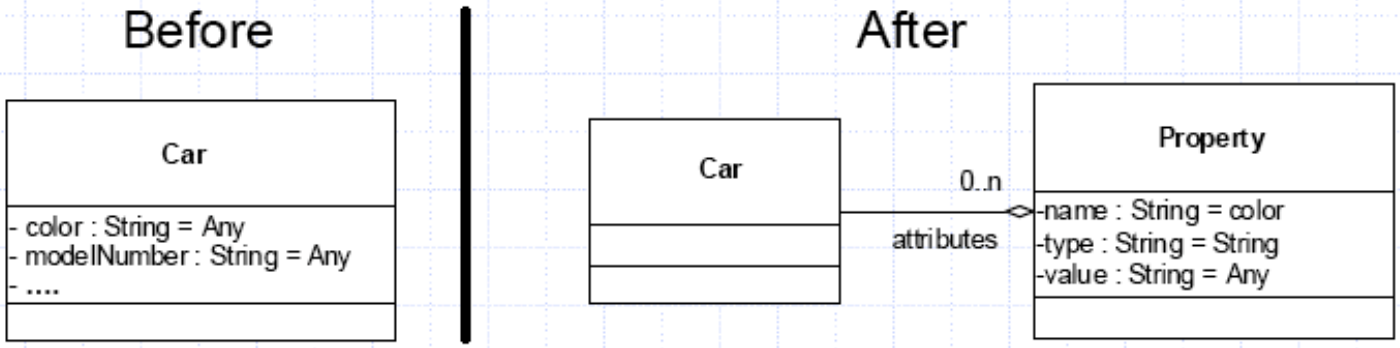
Before



After



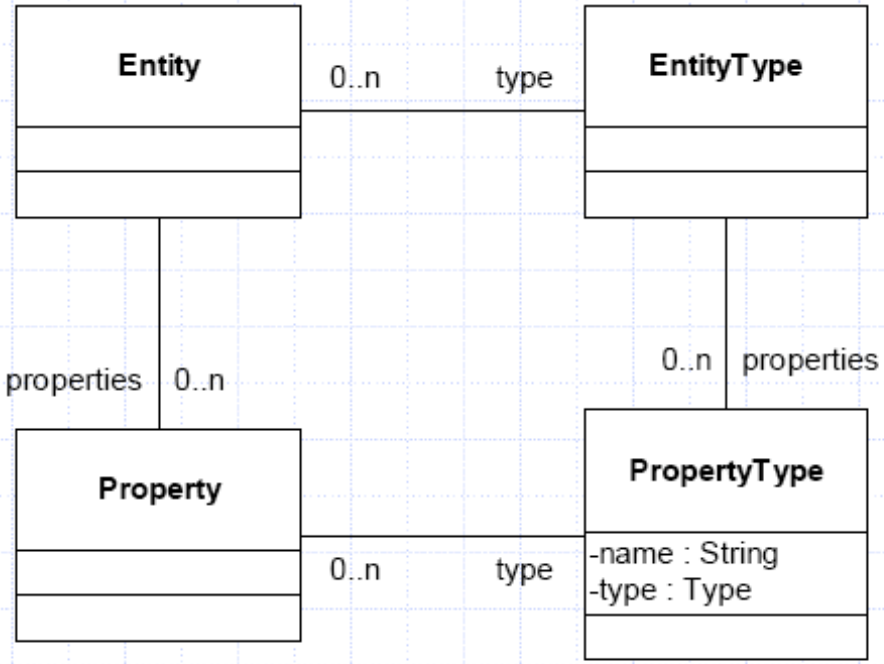
Properties



Example: A Store with Catalogue Entries

- Sweaters (size, color, material)
- Canoes (length, material)
- Video Tapes (name, rating, category)

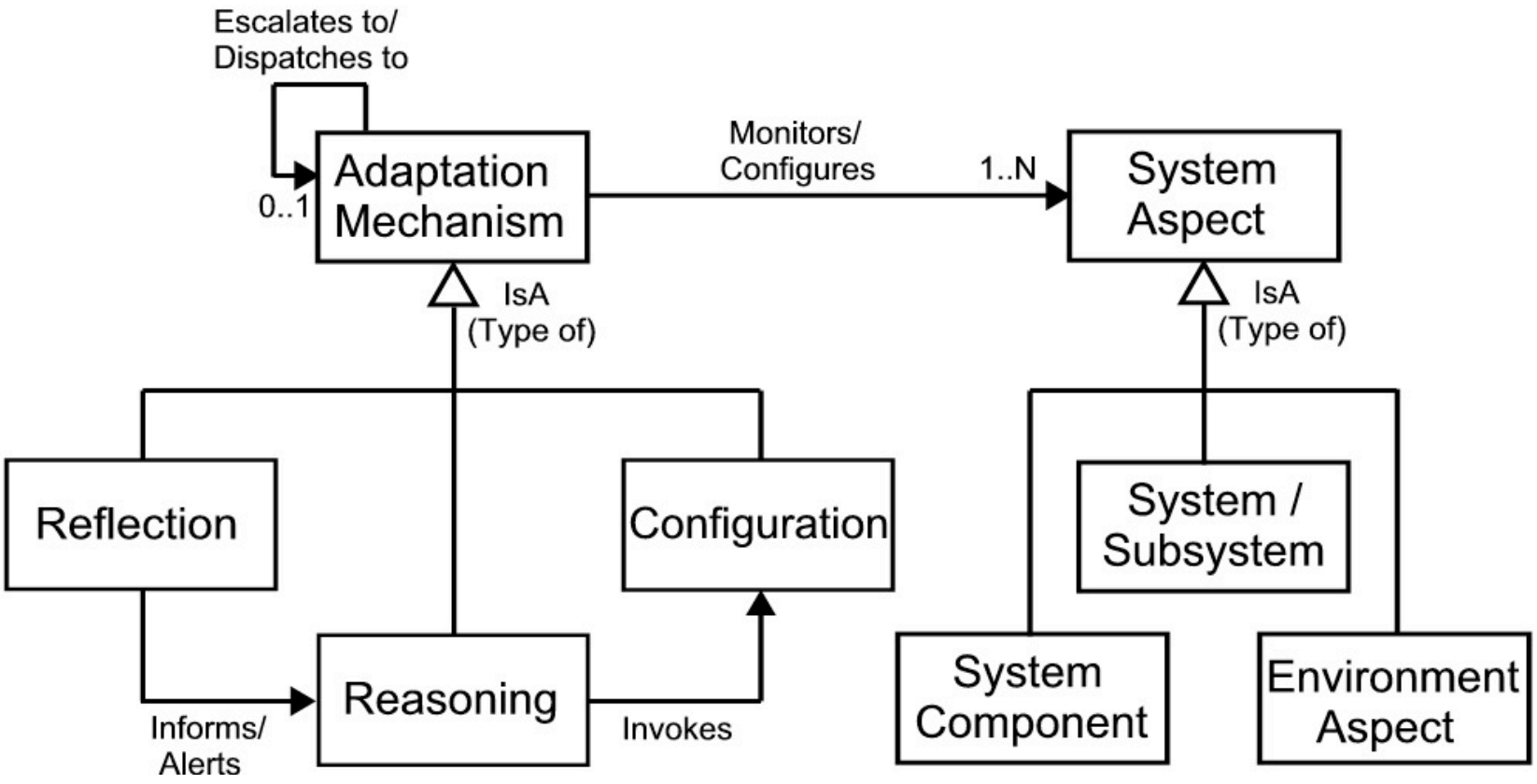
TypeSquare



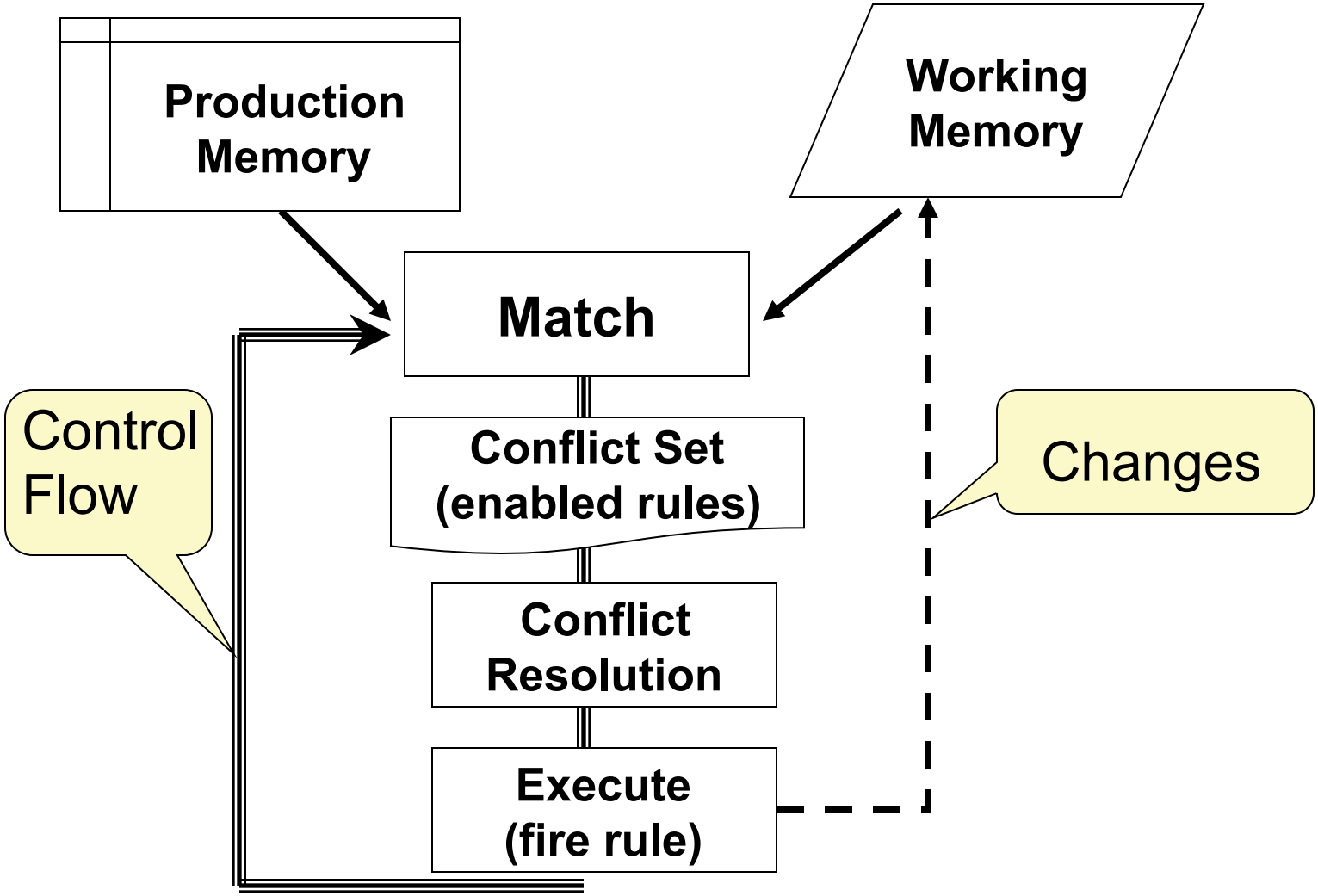
Example: A Store with Catalogue Entries

- Sweaters (size=(S,M,L,XL), color=(red,green,blue,yellow,...)
- Canoes (length=float, width=float)

Conceptual Model for Self-Healing Architectures



Production System Architecture



Mycin architecture

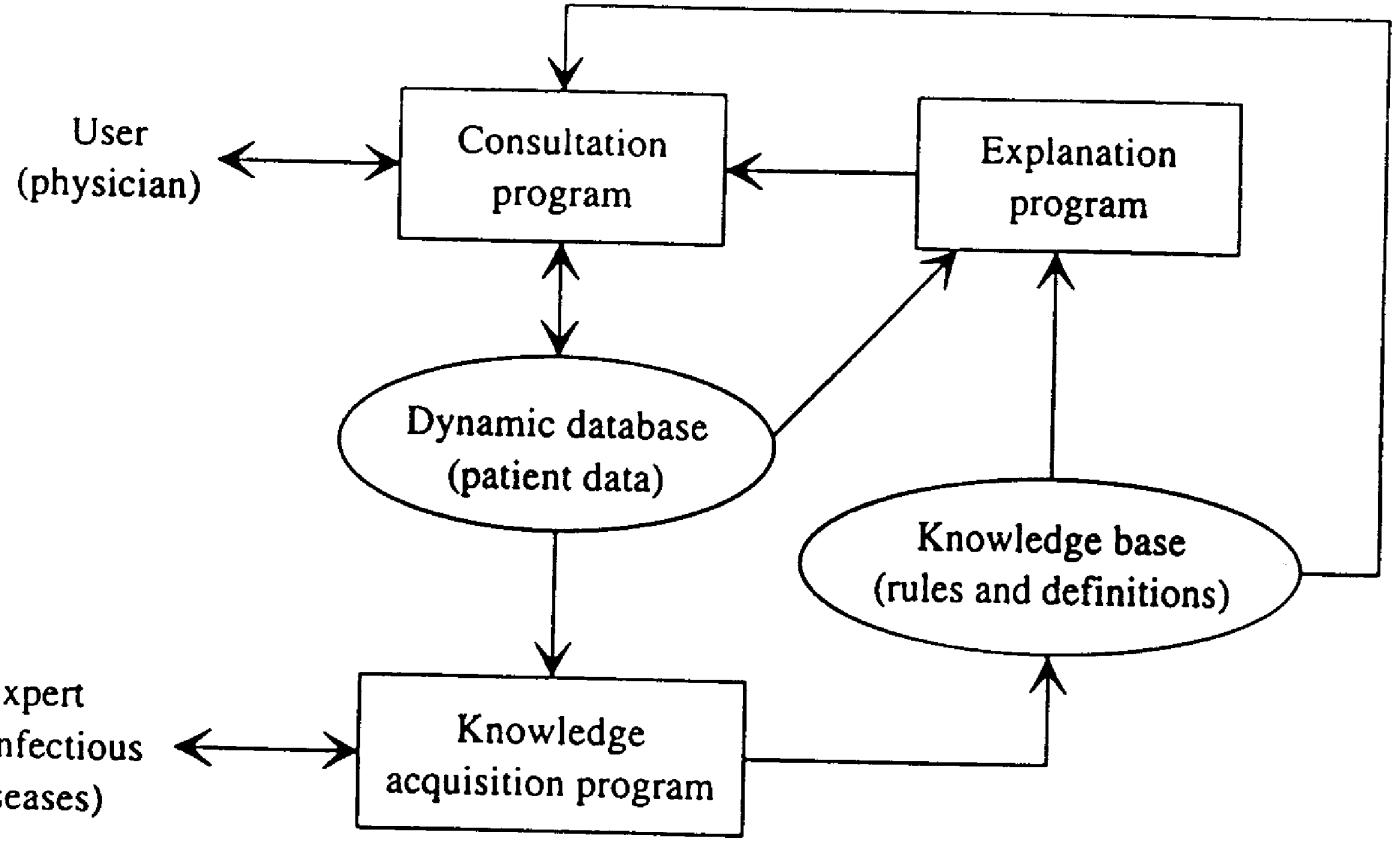


FIGURE 3.45. Overview of the subsystems of MYCIN.