课程信息

- **Instructor**: 赖永炫
- Office: Room 305a, Haiyun A
- Telephone: 13616046216
- Email: laiyx@xmu.edu.cn; laiyongxuan@hotmail.com

- **Textbook:**

  《中间件技术原理与应用》
  张云勇 清华大学出版社

- **References**：

  《CORBA原理及应用》朱其亮, 郑斌编著 北京邮电学院出版社
  《分布式对象技术》 李文军等 机械工业出版社
  《Engineering Distributed Objects》Wolfgang Emmerich, Wiley
  《Learning DCOM》 Tbuan L. Thai, O'reilly
  Google / baidu

相关课程

Java / C++;

Data structure(数据结构);

Operating system(操作系统);

Object-oriented program analyzing and design(面向对象程序分析与设计);

Background in this course

Issues of this course

软件发展的现状

- 软件的出现还只有50多年的历史，是一种新兴的产业。在没有条条框框、标准协议的情况下，各路软件天才各显神通，呈现出百花齐放的局面、没有考虑整合、兼容、互操作问题。
  - 操作系统：Unix, DOS, Windows, Machitosh, Linux;
  - 程序语言：Fortran, Cobol, Pascal, Bacic, C, C++, Java, Object Pascal, C#, Script language.
  - 函数调用规范：_cdecl, _stdcall, _fastcall , thiscall, _pascal, _fortran, _syscall.
  - 编程模式：Procedure-Oriented, Object-oriented, Component-oriented, Service-oriented.
  - 开发工具：Borland，Microsoft, IBM, Sun, Apple.

软件技术的发展是一浪接一浪、后浪催前浪

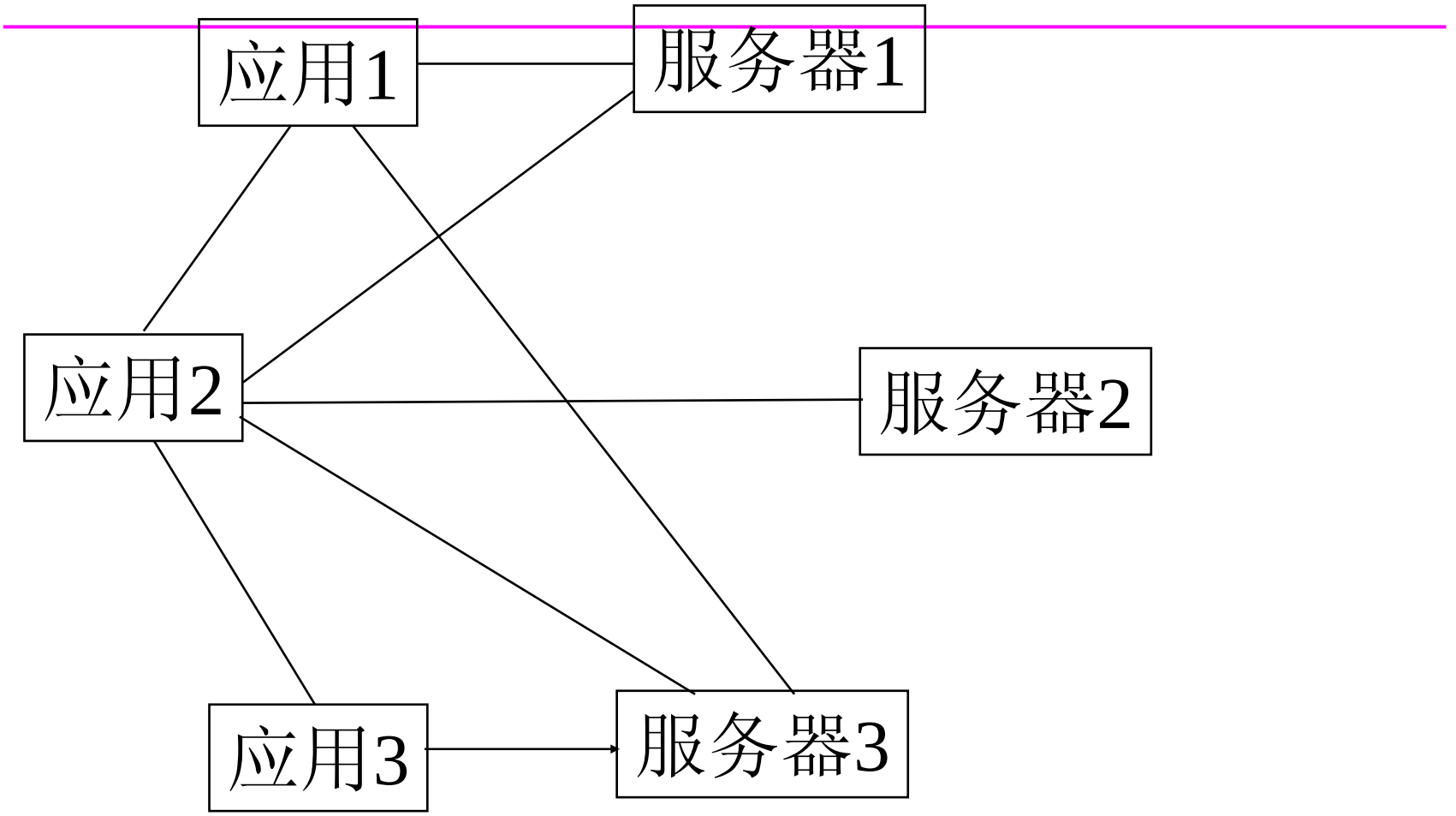
- 当一种软件技术打下一遍江山后，它又面临着新问题，被新的软件技术所取代。
- 新技术并不意味要推翻和抛弃已有的应用系统，因为**用户习惯**，**成本**，**系统可靠性**等等原因。
- 已有的应用系统继续发挥着不可替代的作用，是宝贵的资源。
- 新的应用系统不是要抛弃旧的应用系统，而是要**集成旧系统**，达到**旧貌换新颜**的功效。
  - 高级程序语言屏蔽了操作系统API以及执行模式的差异：C++, Java.
  - 虚拟平台的出现瓦解了编程中的机器和操作系统概念；
  - SOA则把软件的重心从计算（功能实现）移到了互操作的语义上；

现代应用系统的基本特征

◆分布　任务已不只是在单机上运行，而是由网络中多台计算机上的相关应用共同协作完成，需考虑网络传输、数据安全、数据一致性、同步等诸多问题；

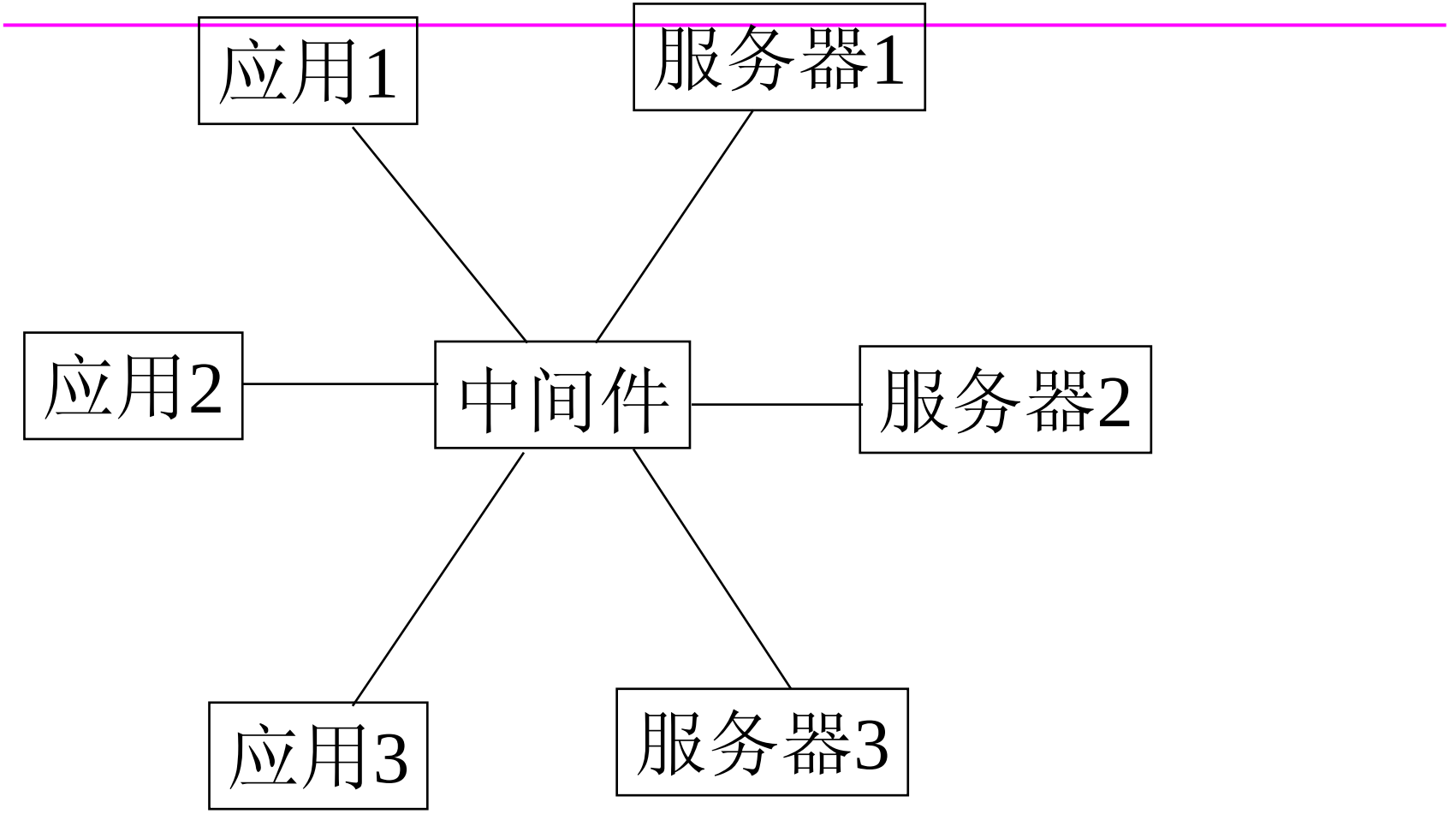
◆异构　计算机硬件、操作系统、网络协议、数据库系统以及开发工具种类繁多，需考虑数据表示、调用接口、处理方式等诸多问题；

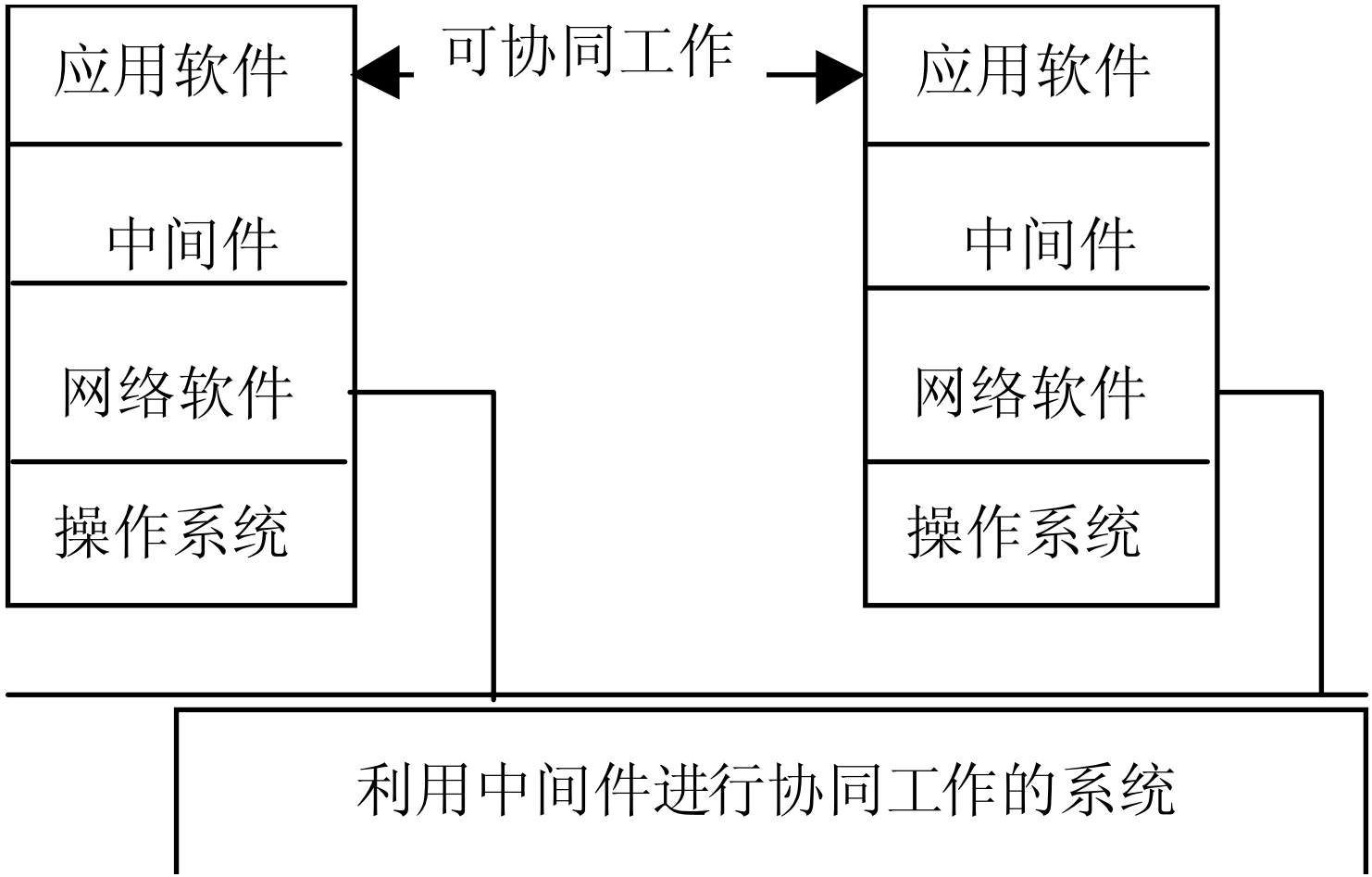
◆动态协作　参与协作的应用允许位置透明性、迁移透明性、负载平衡性等需求。



中间件

- 中间件(Middleware)是一种软件，处于系统软件（操作系统和网络软件）与应用软件之间，它能使应用软件之间进行跨网络的协同工作（也就是互操作）
- 允许各应用软件之下所涉及的“系统结构、操作系统、通信协议、数据库和其它应用服务”各不相同





组成部分

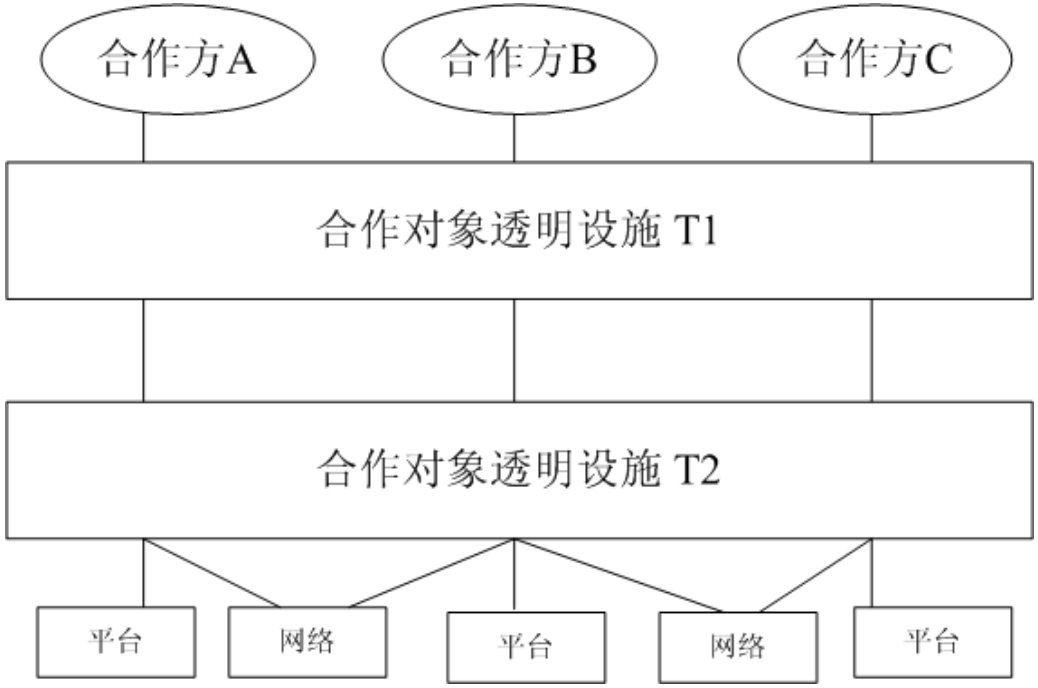
- 执行环境(Execution Environment)软件

  如果一个网络的各个节点安装了EE软件，各节点的应用软件就可以实现相互合作

- 应用开发(Application Development)工具

  一组工具，用来帮助开发内含"透明动用对方"成分的应用软件，或改造原有的无透明动用能力的应用软件

中间件对于应用之间协同工作的贡献



中间件的层次结构

- 提供了合作对象透明设施 T1

  合作一方不必知道另一方是谁和在何处，只说明自己需要怎样的服务，T1就能为其物色一个合适的合作方

- 提供了下层设备透明设施T2

  合作一方不必关心合作的另一方所用的节点设备(机器和操作系统)与本节点的差异

软件技术发展的四次飞跃

- Procedure-Oriented;
- Object-oriented;
- Component-oriented;
- Service-oriented;

软件发展面临的挑战、发展的源动力

- 开发周期；
- 开发成本；
- 软件质量；
- 系统可维护性；
- 产品竞争和开发风险；
- 资源的利用和共享：

最小化成本

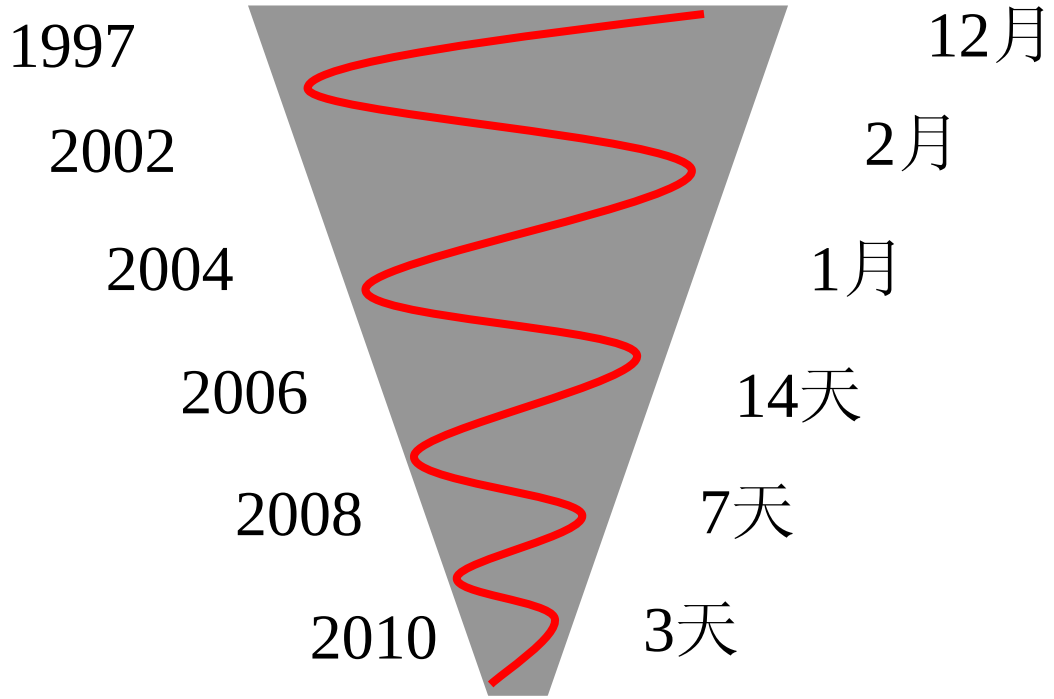
最大化收益；

最强的竞争实力；

软件发展面临的机遇

- Internet的普及又为软件的发展提供巨大机遇：
- 资源的共享使得最大化收益、最小化成本，最快的开发时间成为伸手可及的事情；

快速的开发周期



产品竞争和开发风险

- 第一，软件平台和技术的不断地改朝换代：DOS，Windows，.NET（Java），SOA。
- 第二，随着软件技术愈来愈多，应用也愈来愈多元化。这使得产品的潜在客户群被分散，软件利润趋于下降；
- 第三，信息技术的多元化，产品开发面临难以抉择的状态。面临着虚拟平台、程序语言和信息架构等众多因素的组合变量。

软件危机解决的出路

- 编程的自动化（程序代码的自动生成: 开发工具）
  - 应用的框架代码由开发工具自动生成，程序员只需要编写功能代码；
- 实现程序的通用性 （操作系统与机器）
  - 操作系统级的源程序；（一个程序要写多遍，多次编译链接）；
  - 与操作系统无关的源程序；（一个程序写一遍，基于操作系统和机器要进行多次编译链接）；
  - 与机器无关的源程序；（一个程序只要写一遍，编译链接一遍）；
- 实现软件之间的互操作:
  - 二进制级的软件互操作：同构软件的互操作，异构软件之间的互操作；
  - 一个机器中，同进程内互操作；
  - 同一机器中，不同进程间的互操作；
  - 不同机器间的软件互操作；

编程的自动化：开发工具

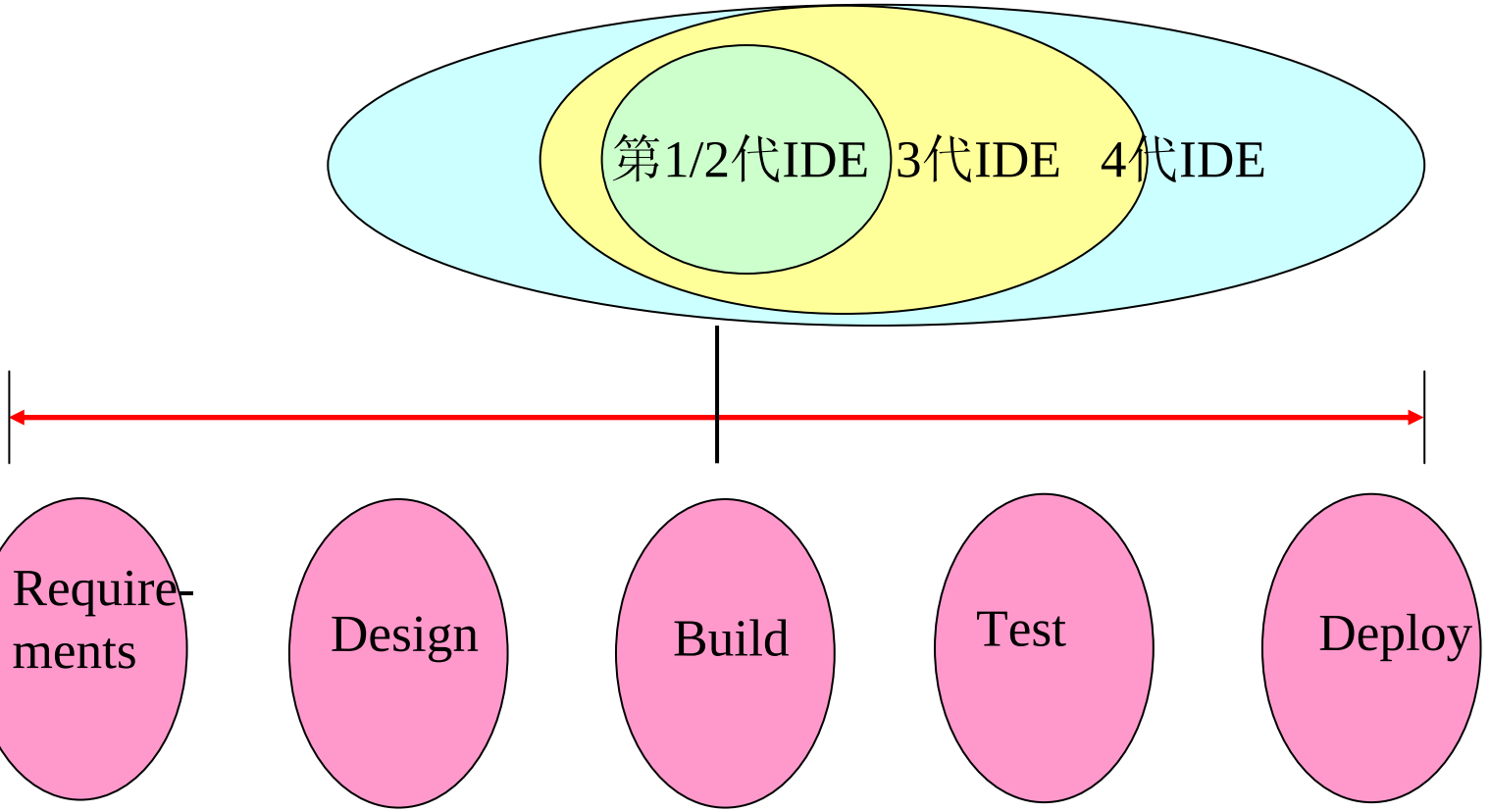
集中开发环境（IDE)从编辑器＋编译器＋连接器， 演变到同时面对专业开发人员，应用开发人员，分析设计人员以及整个开发团队。

集中开发环境仍然是开发周期的核心。提供Framework，以及其他的工具（设计、编程、测试、部署）和Plug-In等。

开发工具提供：

- 基于Framework的代码自动生成；
- 支持库（.h文件，.cpp文件，*.lib(dll) 文件）；
- 编译、链接；

开发工具的演进

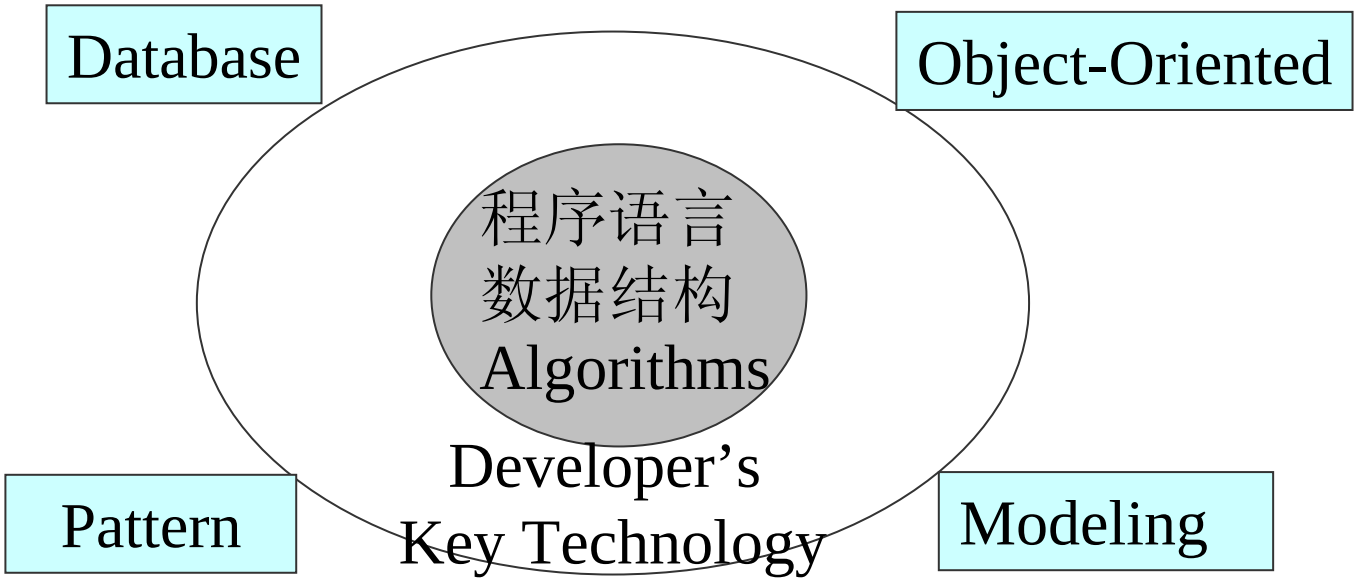


开发工具的演进

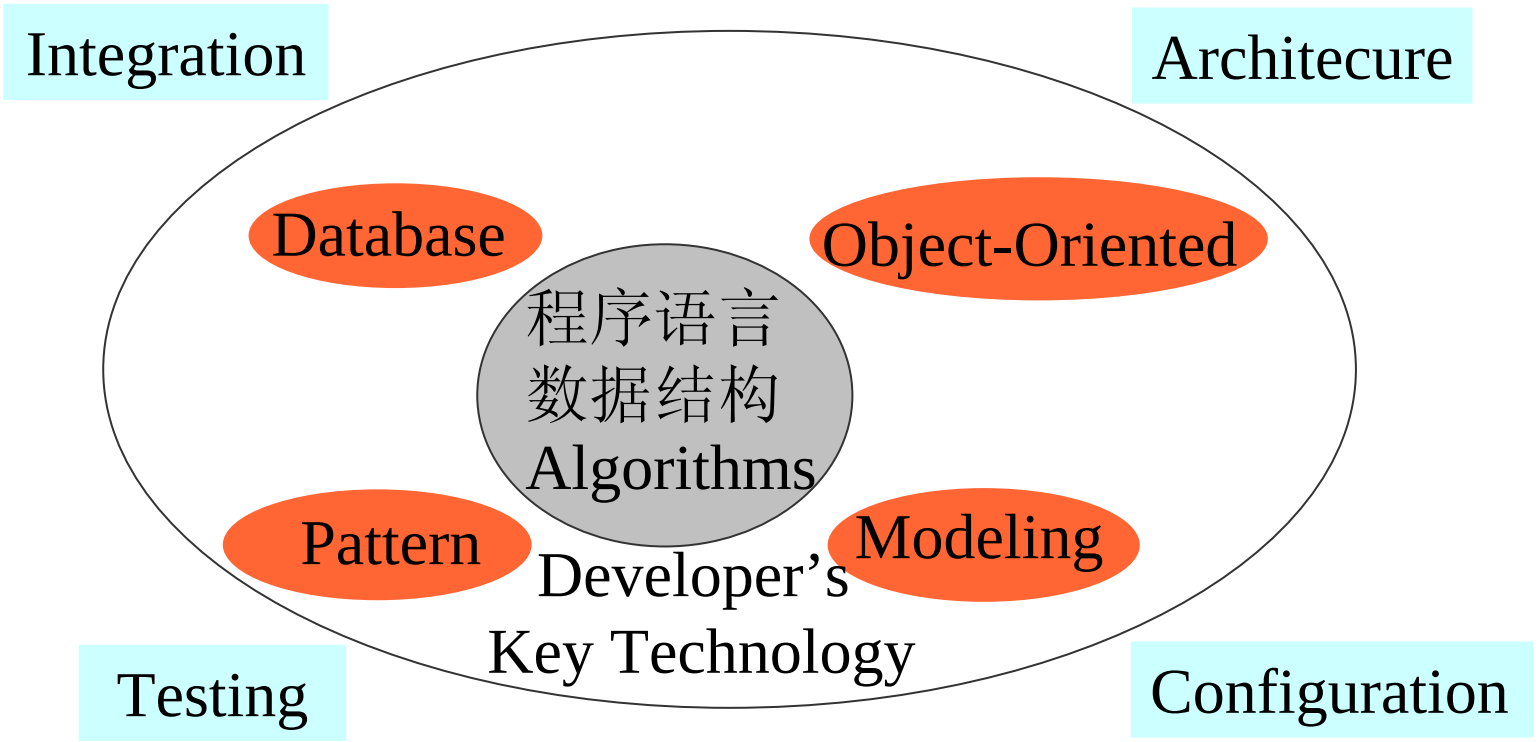
高级语言	第一代IDE Editor + Compiler + Linker	1984	专业工程师
语言/IDE	第二代IDE DOS模式下的IDE	1985 Borland Turbo系列	专业工程师 应用工程师
IDE/Framework MFC/OWL)	第三代IDE GUI模式下的IDE	1993 Borland系列 Microsoft系列	专业工程师 应用工程师 开发团队
语言 /IDE/Framework Java/.NET Modeling/软件工程	第四代IDE IL IDE	2003 Java IDE .NET IDE	专业工程师 应用工程师 开发团队 分析/设计人员

软件开发的知识面

单一程序语言



软件开发门槛的提升



由点到面的全方位软件技术

每一个时代都有主导的软件技术在影响着当时的产品及软件公司的兴衰。不同年代中，不同信息技术成了当时的主宰力量：

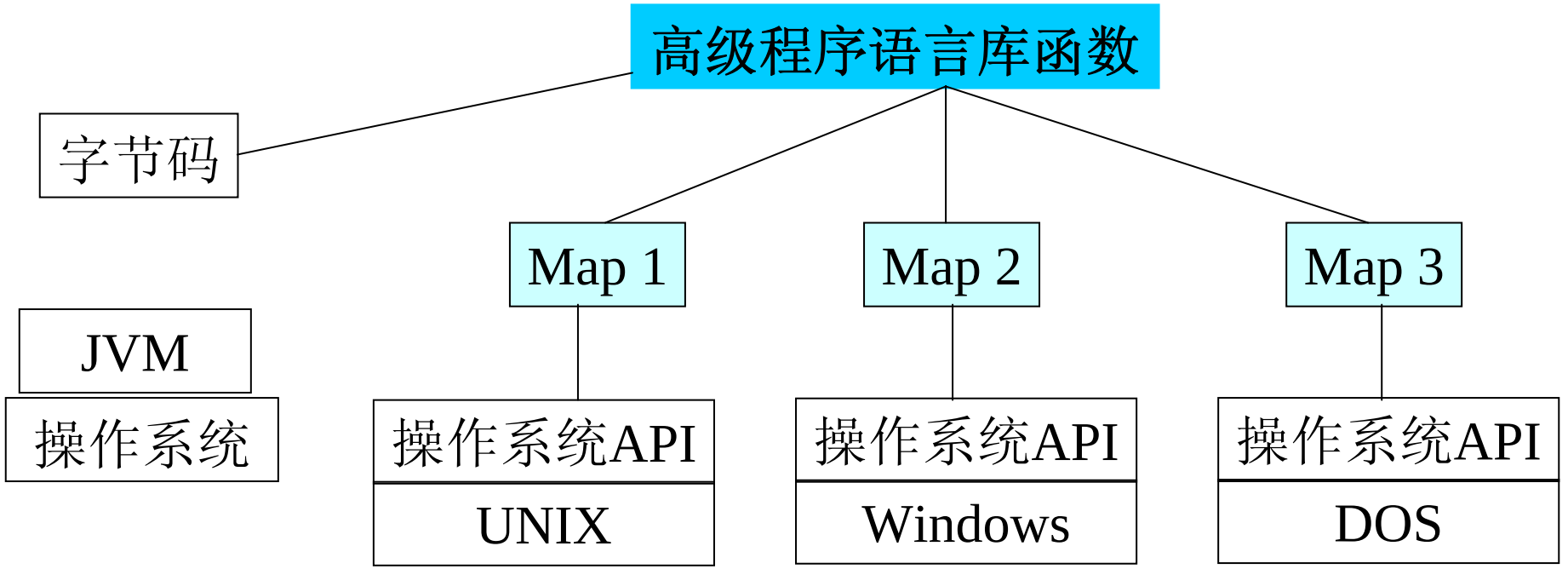
- 60/70年代，**数据处理**和**程序语言**独领风骚；
- 80年代，便由**数据库**当家作主；
- 90年代，各种**组件**和**中间件**又主导了系统架构；
- 00年代， 平台**整合**、**集成**时代。

随着**面向对象**和**中间件技术**对于信息系统的影响愈来愈大，信息技术的演进逐渐**从点形成了面**，在2001年之后主要的软件力量来自平台的整合和竞争、以及**全方位的软件技术**。

发展趋势的归纳

- 应用程序之间的整合受到愈来愈多的重视。
  - ✓ Internet/Intranet，Web的影响非常深远；
  - ✓ 百花齐放的软件技术和程序语言，没有考虑日后整合；
  - ✓ 整合/集成要使用一种松耦合（loose coupled）构架并借助标准的沟通方式和数据来整合已有系统是必经之路。
- Web Service 提供了跨平台的能力，XML则是跨平台的数据交换标准。整合分散应用的最终选择是Web Services和XML.
- 整合应用系统须要自动化工具以及一个整体构架（SOA Service-oriented Architecture）——（开发工具，和系统支持环境）。

实现源程序的通用性



实现源程序的通用性

- C语言中的库函数：

```
malloc(size_t size);
```

- 在Windows中的Map实现：

```
malloc(size_t size)  {
    HANDLE h = GetProcessHeap( );
    Return HeapAlloc(h, 0, size);
}
```

- Java语言：

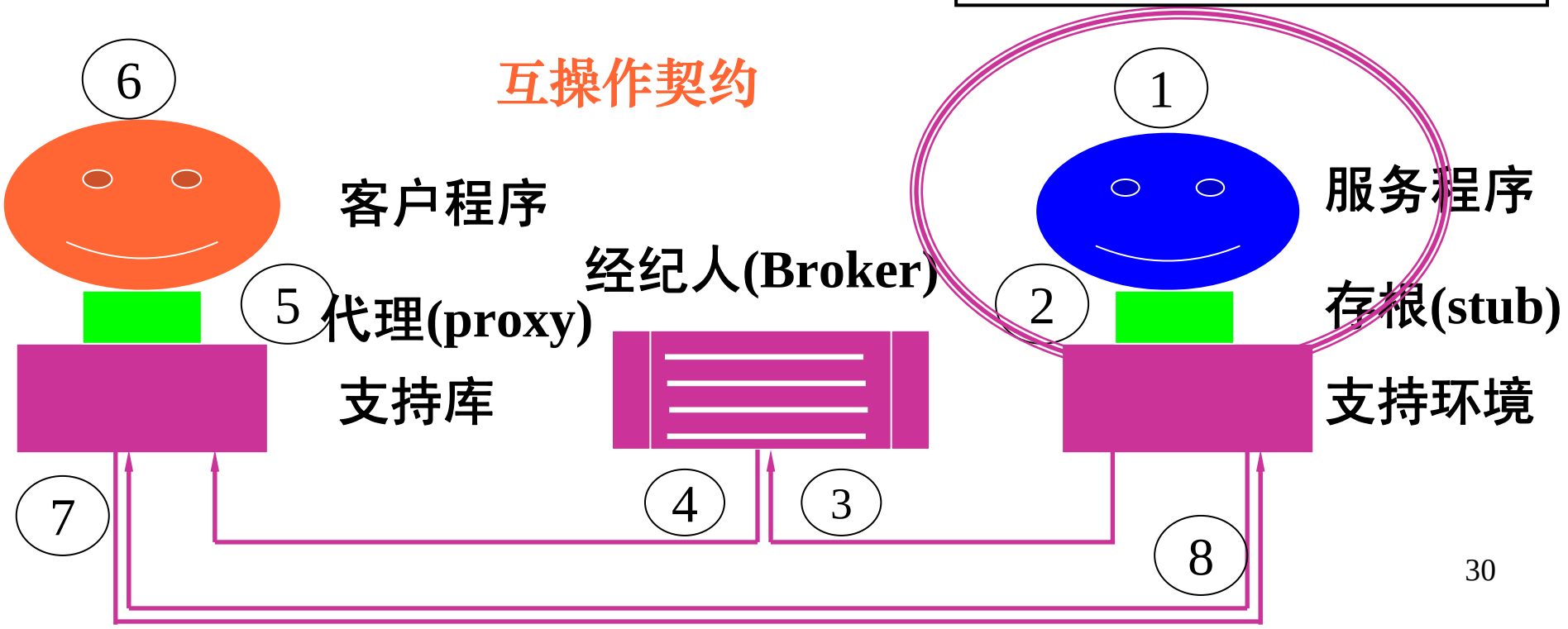
```
CObject p = new CObject;
```

互操作的基本概念

1）分离性
2）透明性

1）运行环境
2）开发工具
3）客户软件；
4）服务软件；

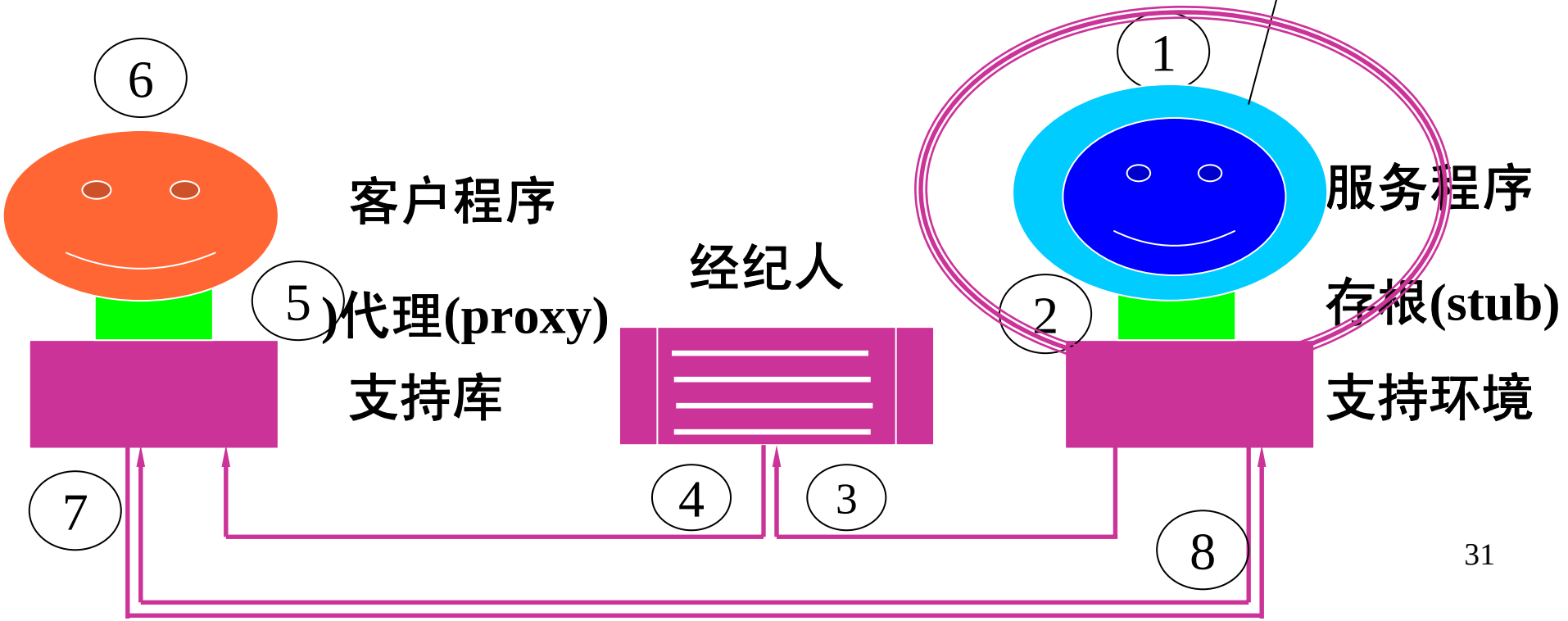
动态性：
1)直接交谈：无
2)同一语言交谈：电话
3)不同语言交淡：翻译



互操作的基本概念

1）运行环境
2）开发工具
3）客户软件；
4）服务软件；

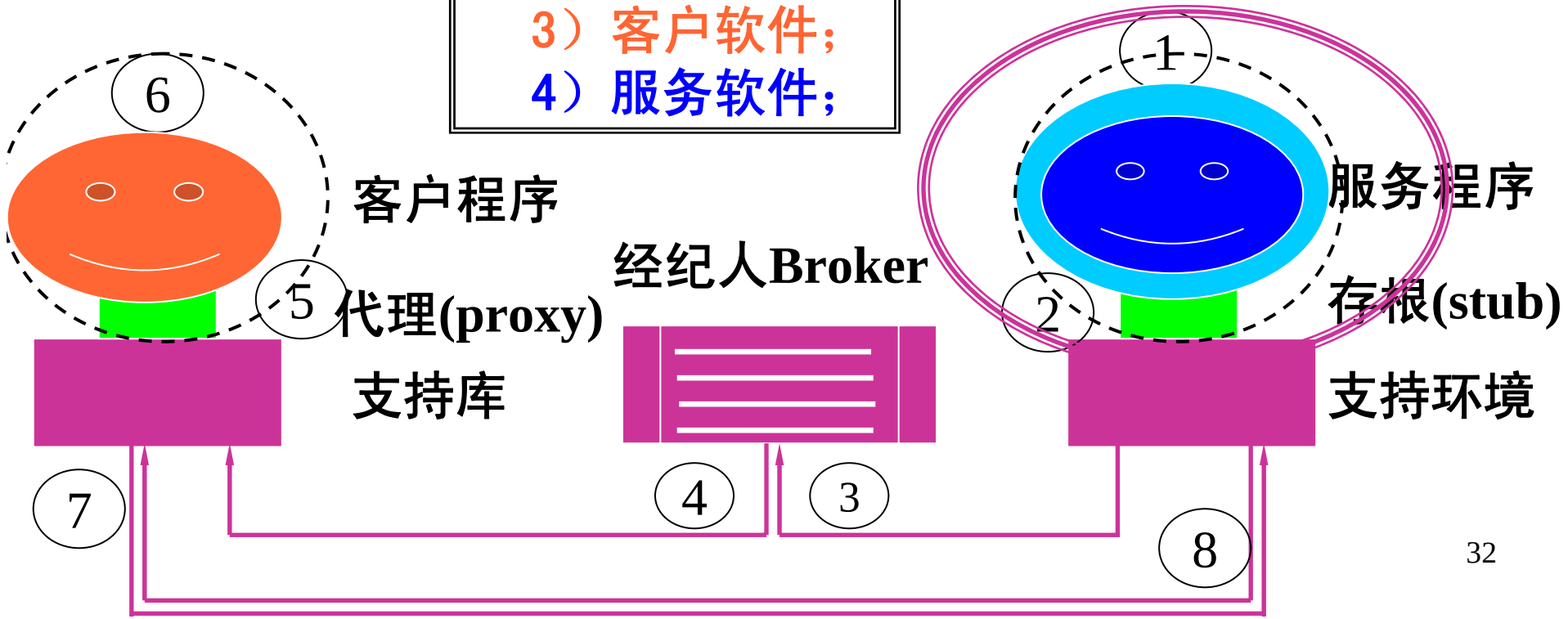
基于互操作规范
加装接口



中间件技术的含义

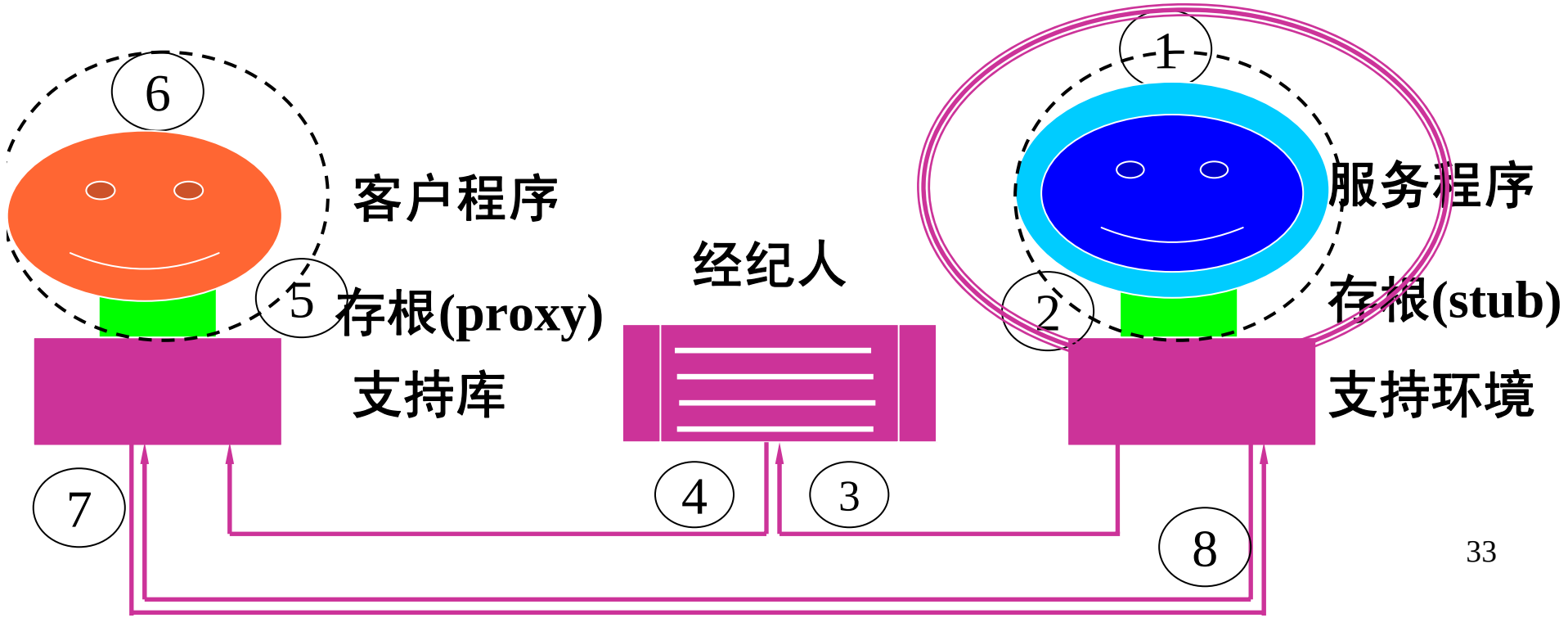
运行环境
开发工具
→
应用软件
的开发
→
应用软件
的部署
→
应用软件
的运行

1）运行环境
2）开发工具；
3）客户软件；
4）服务软件；



中间件特性

1）二进制级的互操作；
2）服务程序有多个客户；
3）客户和服务都会升级进化，无约束，可组合性；
4）对应用程序员，保持编程模式的不变性；
5）客户和服务之间的契约既要具有永恒性，又具有可延伸性；



互操作的范围

有三种形式：它们都要考虑**服务程序的标识，动态适配**

- 同一进程内不同模块之间的互操作；

  可以直接完成；

- 同一机器中不同进程　之间的互操作；

  要求进程之间通信，不用考虑Data encoding；

- 不同机器之间模块之间的互操作；

  要求机器间通信，要考虑不同Data encoding之间的转换，因此须要中间翻译；还要事先约定传输协议；

中间件技术的优点

✓使用中间件技术之后，软件的开发的重心从功能实现变成了功能组件的组合；解决了**成本问题、开发周期问题、质量问题，提高了软件的利用率**。

✓软件的功能的扩充不再需要重新编译和链接，可以在**二进制级完成软件功能的扩充**；因此谁都可以对一个软件进行扩充，进行剪裁；

✓总之中间件技术使得软件**从封闭变得开放**，变成可为**别人服务**，**自己也可以扩充**，取得了**互利互赢**的功效。

中间件技术的优点

- 由于标准接口对于可移植性和标准协议对于互操作性的重要性，中间件已成为许多标准化工作的主要部分。

- 中间件提供的程序接口定义了一个相对稳定的高层应用环境，不管底层的计算机硬件和系统软件怎样更新换代，只要将中间件升级更新，并保持中间件对外的接口定义不变，应用软件几乎不需任何修改，从而保护了企业在应用软件开发和维护中的重大投资。

中间件的分类

- 事务处理中间件(TP Monitor: Transaction Process Monitor)
- 消息中间件(MOM: Message-Oriented Middleware)
- 数据库中间件(Database Middleware)
- 远程过程调用中间件（RPC: Remote Process Call）
- 对象请求代理中间件(ORB: Object Request Broker)
- J2EE中间件

中间件相关的技术

- RPC/RMI
- COM/COM+/
- .net
- J2EE
- CORBA
- JINI
- WEB SERVICE
- ……

远程过程调用

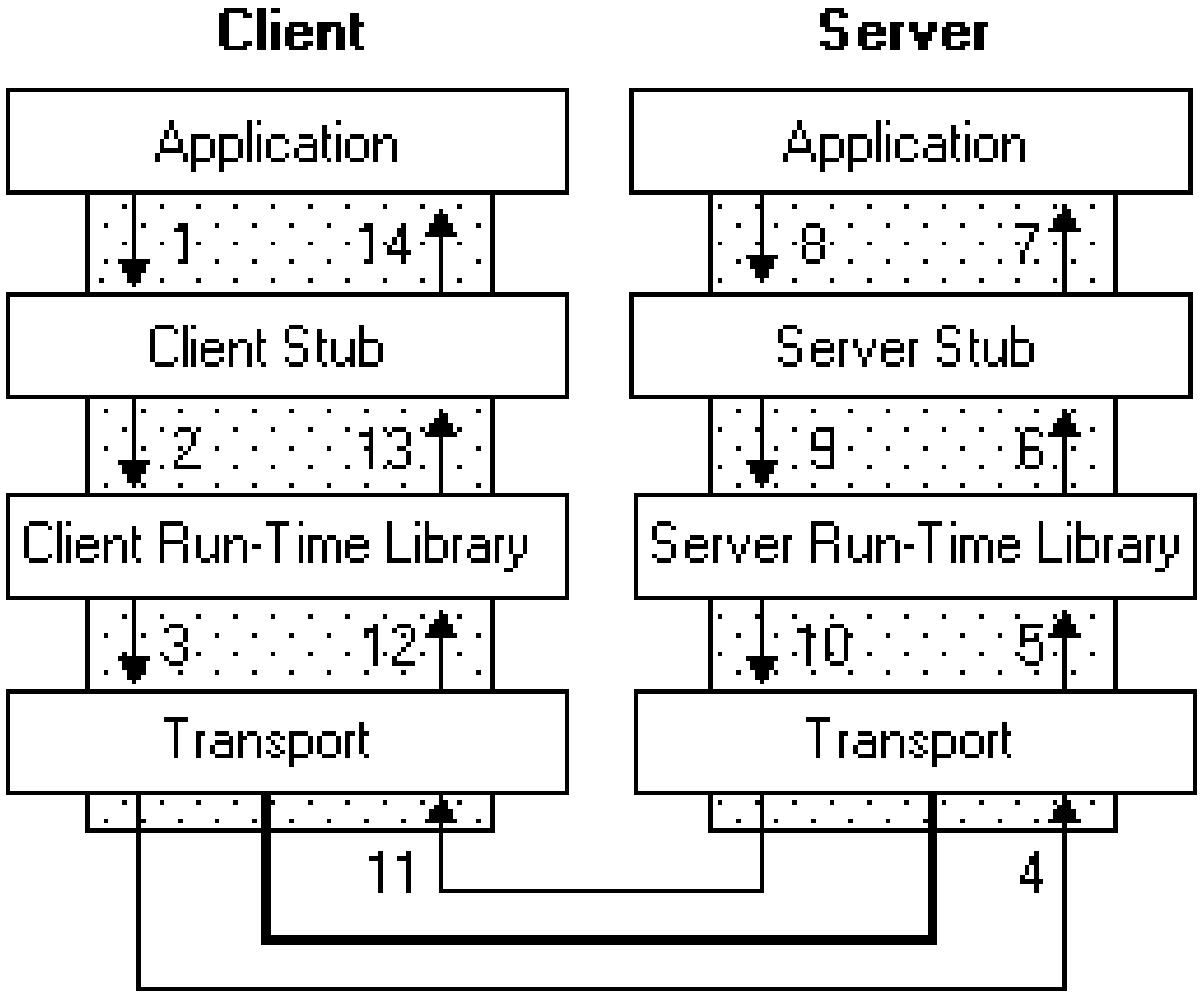
- 远程过程调用协议(Remote Procedure Call Protocol):

  一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议

RPC

NDR format

传输协议

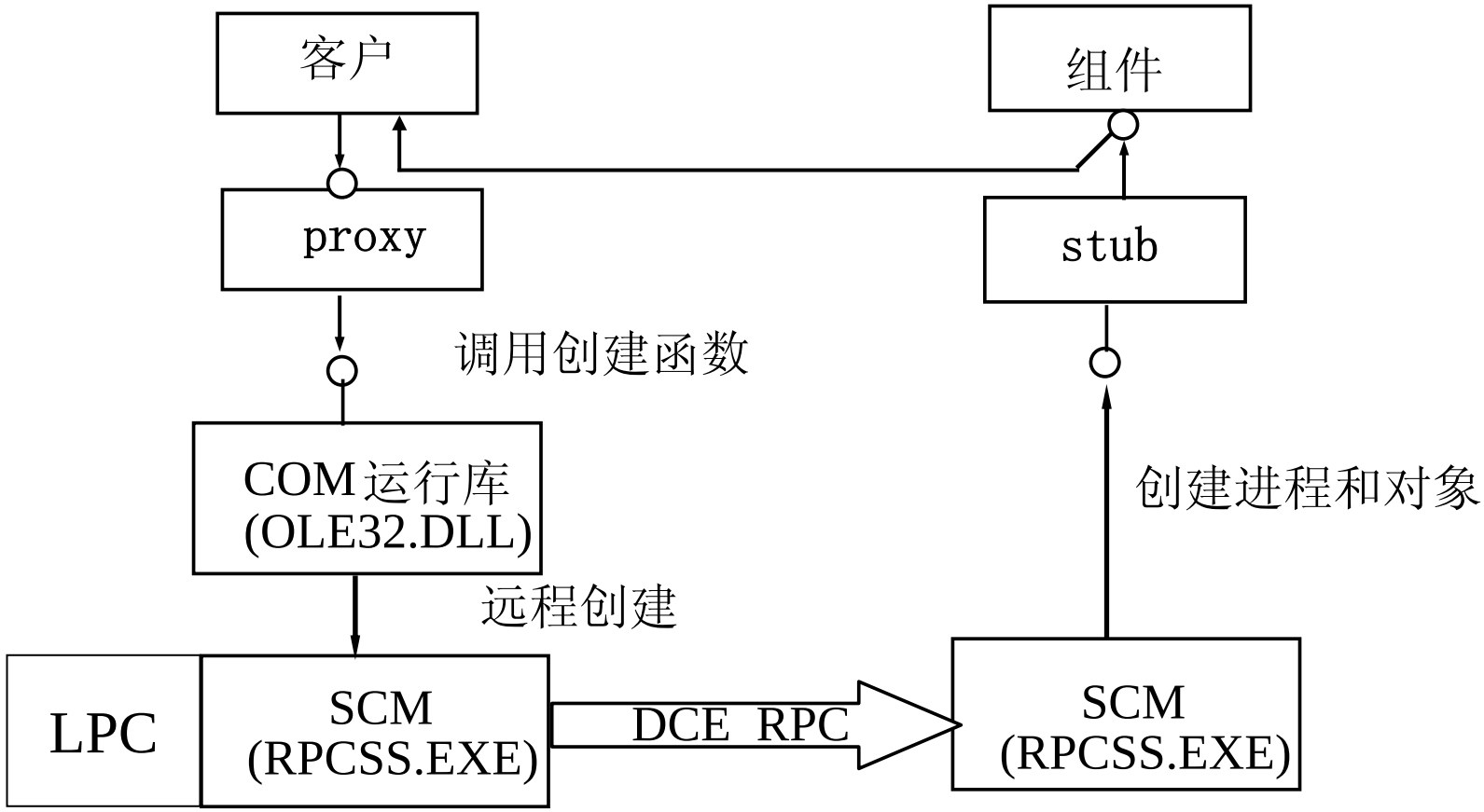


NDR: Naming and Design Rules

COM组件

- 组件对象模型（Component Object Model, COM）是: COM组件是以WIN32动态链接库（DLL）或可执行文件（EXE）形式发布的可执行代码组成，遵循COM规范编写，可以给应用程序、操作系统以及其他组件提供服务，可以动态的插入或卸出应用
- 在COM构架下，人们可以开发出各种各样的功能专一的组件，然后将它们按照需要组合起来，构成复杂的应用系统

COM

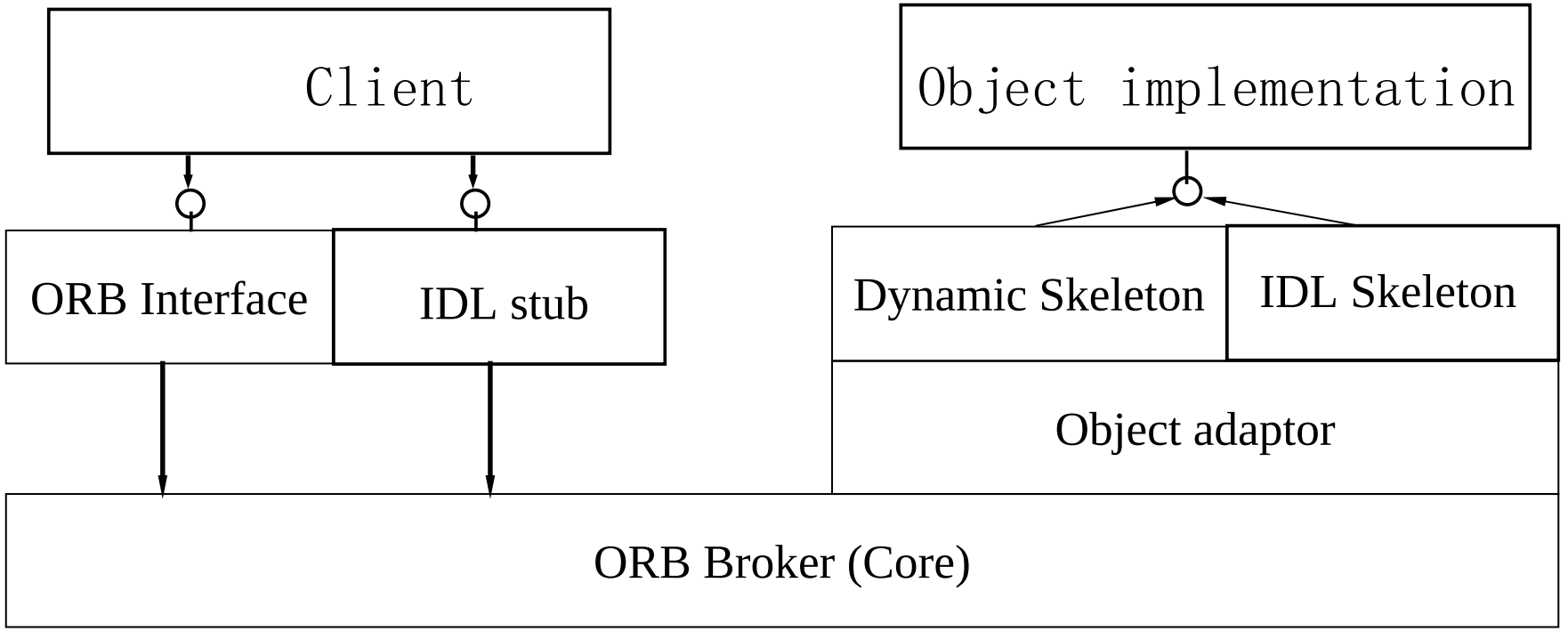


ORPC (Object RPC)

公共对象请求代理体系结构

- CORBA（Common Object Request Broker Architecture公共对象请求代理体系结构）是由OMG组织制订的一种标准的面向对象应用程序体系规范
- CORBA体系结构是对象管理组织（OMG）为解决分布式处理环境(DCE)中，硬件和软件系统的互连而提出的一种解决方案

CORBA

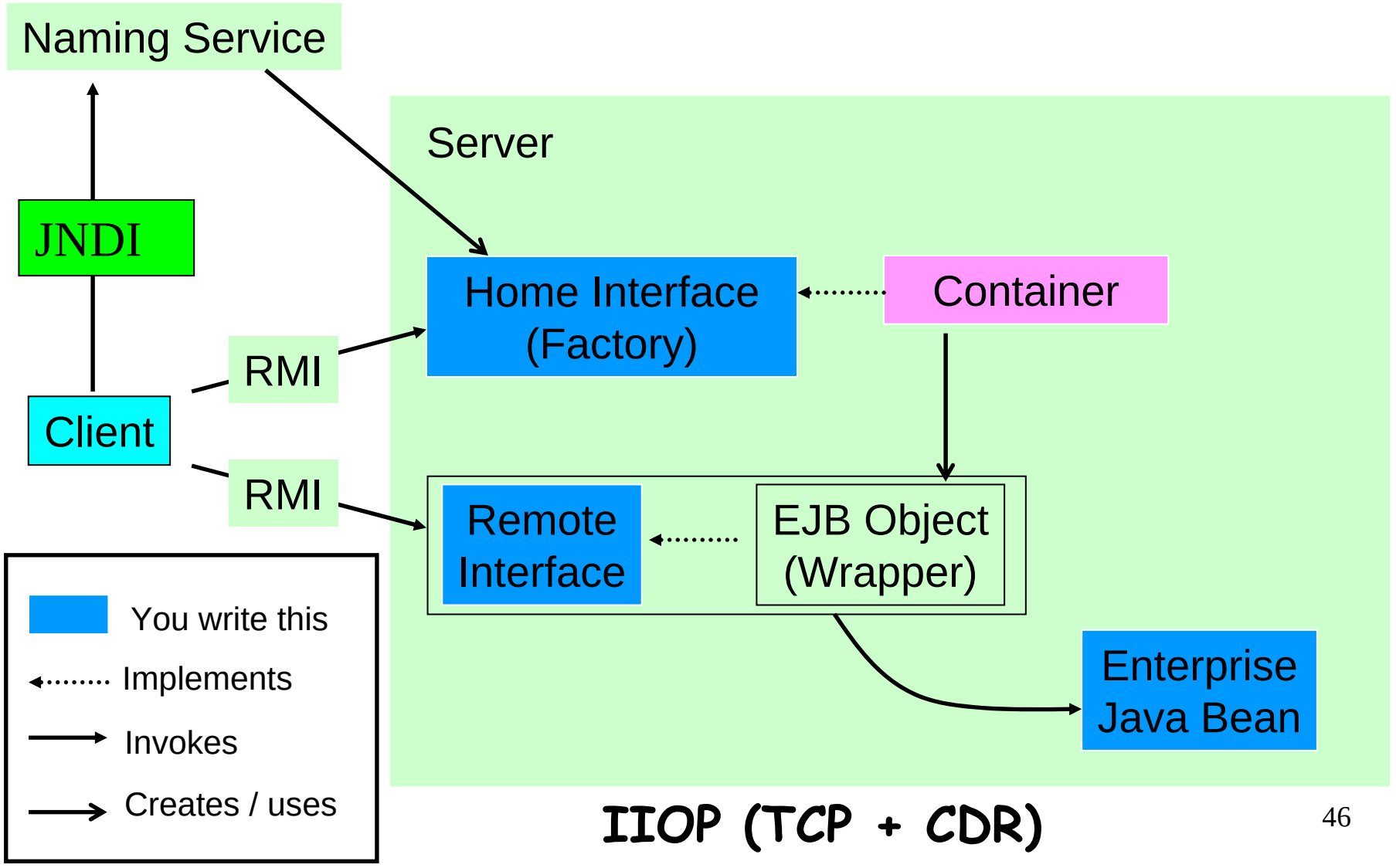


IIOP (TCP + CDR)

企业级Java Bean

- EJB(Enterprise Java Beans)是sun的服务器端组件模型，最大的用处是部署分布式应用程序。凭借java跨平台的优势，用EJB技术部署的分布式系统可以不限于特定的平台。
- 一个技术规范：EJB 从技术上而言不是一种"产品"

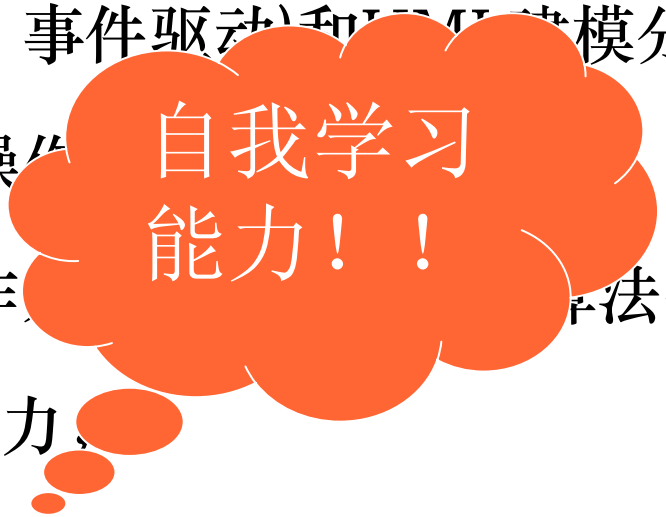
Enterprise Java Beans （ EJB ）



软件人才的基本专业素质

在竞争愈来愈激烈的软件业中，软件人员都必须具备：

- 精通面向对象技术(交互接口、事件驱动)和[illegible]模分析技术；
- 快速掌握系统架构的能力(互操[illegible]
- 快速掌握程序语言的能力(操作[illegible]法、视窗)；
- 快速查找资料的能力、英语能力；
- 精致化的开发能力；
- 也需要了解软件技术发展的规律，以便规划自身的发展方向。



软件技术的学习方法

- 软件行业的发展，终会像发展了上百年的汽车工业一样，从百花齐放，到走向成熟。最后只剩下少数的IT企业，从开发工具、服务器、到应用程序从源头提供成套产品。大部分软件人员将从事装配性、定制性、维护性工作，或者信息服务工作。
- 面对程序语言、数据库、组件模型，平台，以及Framework等不断涌现的新技术(新名词)，很多人似乎陷入了永远学不完新东西的梦魇。
- 不过，如果你仔细思考这些技术的本质，就会感悟它们都不过是对代码和数据的精致整理而已。抓住了本质，就不会陷入永无止境的学习梦魇，就会在学习中游刃有余，如鱼得水。

章节小结

- 软件技术发展从百花齐放，逐步过渡到共识和统一；
- 发展的源动力：最小化成本，最大化收益、最强的竞争实力；
- Internet的迅猛发展，使得系统集成/整合成为当前要解决的紧迫问题；
- 解决软件危机问题的途径：软件通用，软件互操作；
- 中间件技术就是软件互操作技术；
- 中间件技术包含运行环境、开发工具、应用程序三方面的内容；
- 中间件技术涉及到三方：客户，服务，经纪人；

课程目标

- 学习和掌握主流的中间件技术和软件互操作方法：
  - 掌握软件中间件的原则
  - 掌握软件互操作的框架
  - 了解如何设计中间件
  - 了解中间件技术的发展和软件发展的未来方向
- 提高个人竞争力(To improve Personal competences): speaking, demonstrating, communicating, collaborating, innovating, and learning.

课程安排

- 相关知识的介绍
  - RPC/RMI
  - CORBA
  - COM/COM+/ .NET
  - J2EE
  - JINI
  - WEB SERVICE
  - XML
  - ……

考核方式

- Participation: 10
  出勤率、课堂提问和讨论
- Presentation: 20
  5~6人/组 选组长
  在9月21日前将分组情况Email本班助教；助教会将分组情况在FTP上公布;
  zhenpengzhan@gmail.com 詹振鹏
- Experiment: 20
  3~4个
- Final Exam: 50

Presentation

- 报告内容：
  - 某一项技术/规范/思想
    - 历史、背景、与中间件技术的关系
  - 介绍技术以及细节(概要的)
    - 可根据具体的系统和例子来说
- 评价标准：
  - 材料丰富，介绍深入，正确的评价
  - 综合展示和表达能力
- 提问和讨论：
  - 问题的质量，回到是否正确
- 时间安排：
  - 4~6次课，穿插在课堂中（10月份开始）
  - 报告前一周提交资料和幻灯

- 书本知识的进一步扩展和介绍（针对某一侧面）：
  COM+, .net, corba, J2EE, jini
- 编程语言类介绍：
  C#，ruby，python, eiffel, tcl/tk, D programming language,
- 热门技术：
  Structs, Spring, Jboss, ajax, ATL, Hibernate, Eclipse
- 数据库相关：
  DAO, Content Management
- 流行概念：
  云计算，网格计算，普适计算，Mashup, SNS, Web 2.0, SaaS, p2p
- 协议
  XML, SOAP, IIOP
- 案例
- …… ……
- …… ……

谢谢

部分幻灯摘自 **Jin-Min Yang (杨金民） <Middleware technology>, Software School of Hunan university**