



**Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom**

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (01223) 515010
+44 1223 515010
+44 1223 359779
apm@ansa.co.uk**

Training

ANSAwise - Introduction to Distributed Systems [CNET]

Chris Mayers

Abstract

Organizations are aware that distributed systems may offer business benefits, but may be unsure exactly when is on offer, what the difficulties are, and whether they can actually deliver the promised benefits.

This is the first module of the ANSAwise training programme, and outlines the business case for distributed systems. It also introduces the ANSA principles.

This module covers a lot of ground rapidly; all the points are expanded in later modules.

[This is a variant of APM.1328 developed for CNET.]

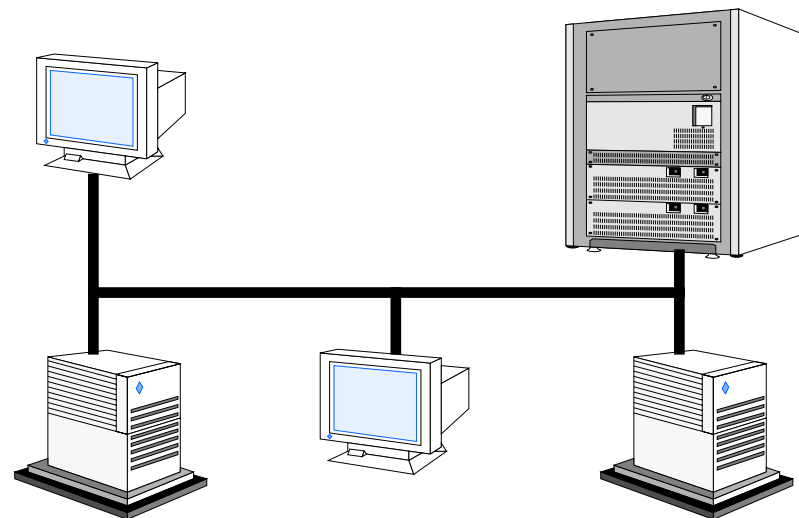
APM.1627.01

Approved
Briefing Note

17th March 1996

Distribution:
Supersedes:
Superseded by:

Introduction to Distributed Systems





In this session

- *Explain the business issues surrounding distributed systems*
- *Explain in what ways distributed systems are different*
- *Explain a general approach that helps you build distributed systems*



What's the real business challenge?

Coping with change

The pressures for change

- *Political, economic, social, and technological...*

- **Globalization**



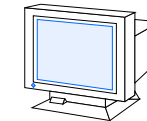
- **Rapid organizational change**



- **Increased customer expectations**

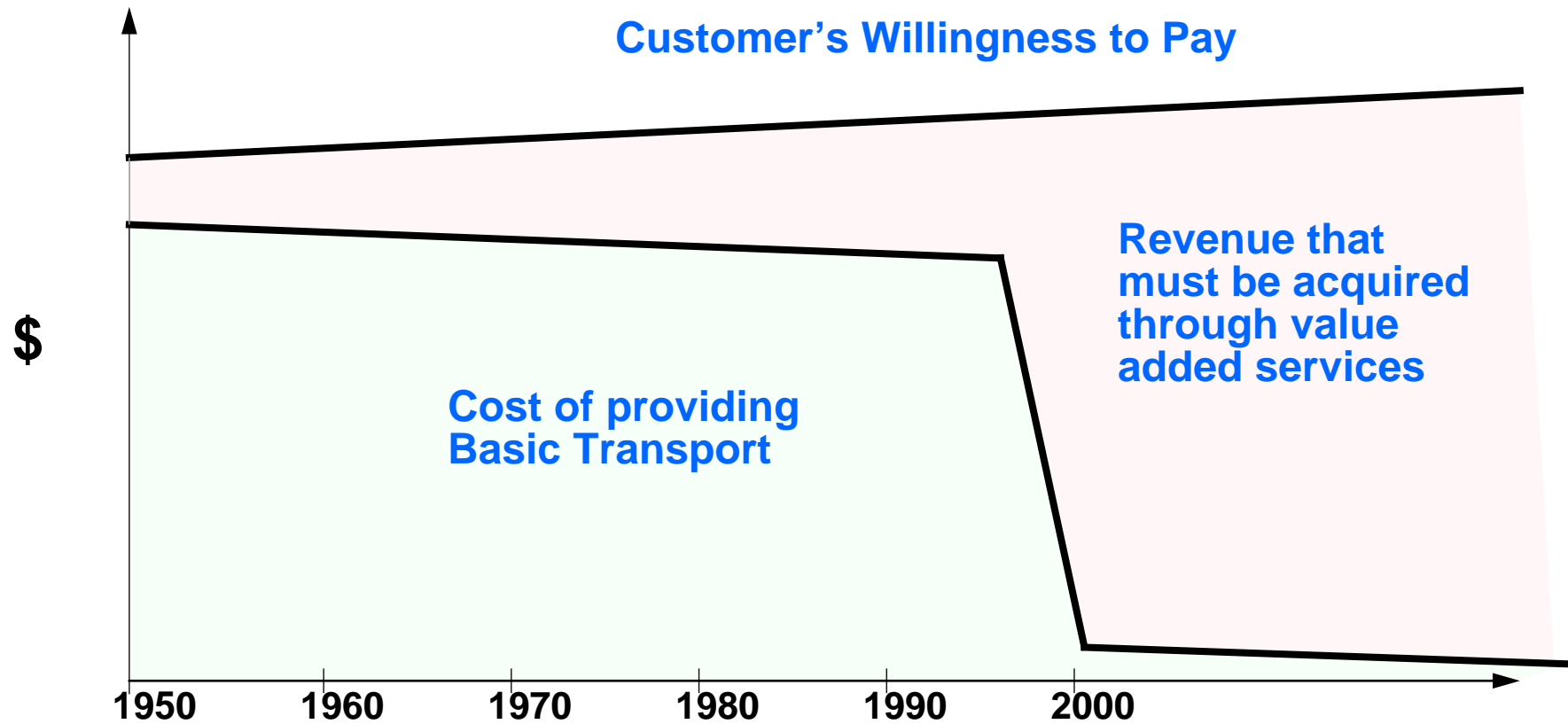


- **Inexpensive computing and telecommunications**





The business challenge for telecommunications





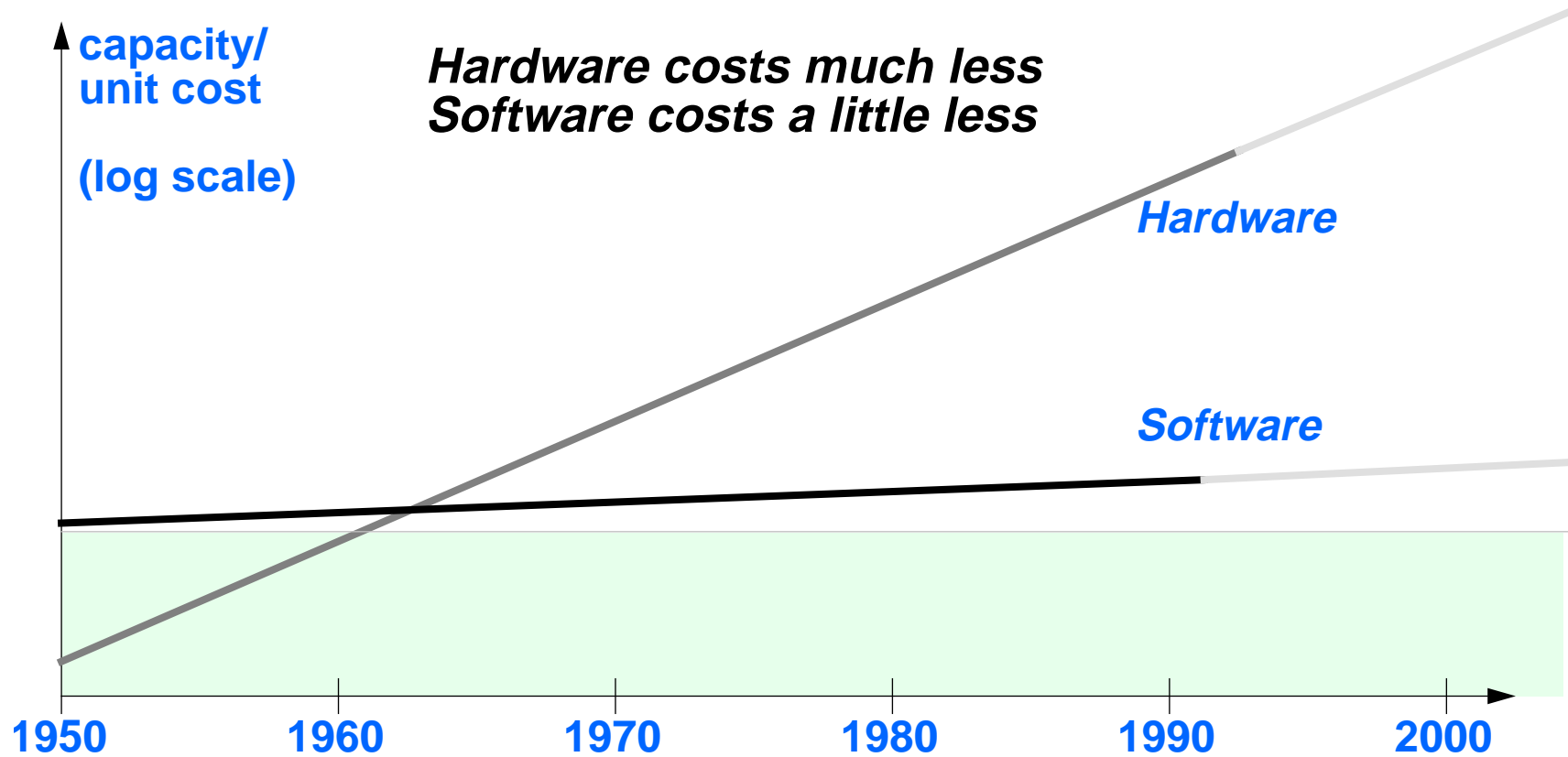
Meet the customer's service expectations...

- *timely:* *I want it immediately*
- *personalised:* *I want it to meet my needs*
- *competitive:* *I want to pay as little as necessary*
- *dependably predictable:* *I want it to be reliable*
- *integratable:* *I want it connected to my PABX, PC, ...*

.... before your competitor does



Costs of providing service





Software cost in providing services – a new problem?

- *We have already tried these solutions....*
 - Project management
 - Requirement analysis
 - Analysis and design methodologies
 - Informal and formal approaches
- *....they work, but not for complex systems*
 - they do not *scale*



The service provider's problem - Summary

- **Providing networked information services**
 - not simply the physical transport of data
- **Satisfying the Critical Success Factors**
 - services must be *developed rapidly*, to meet market windows
 - new services must *interwork* with existing services
 - services must be *easy to deploy*
 - services must be *easy to manage*
- **Meeting the customer's expectations**
 - before your competitors
 - at a price the customer will pay



About distributed systems

- *Distributed systems are those which consist of interconnected cooperating components*
 - there is no central machine or group of machines
- *Distributed applications are those written for a distributed system*
- *Distributed processing is the method for designing and building distributed applications*
- *Distributed computing is the technology we use in distributed systems*



Examples of distributed systems

- ***Diverse business areas***
 - **Telecommunications**
 - **Airline reservations**
 - **Retail point-of-sale**
 - **Banking**
 - **Command and control**
 - **... and many more**

- ***Built at the limits of the technology***



Features of distributed systems

- *Diversity: many types of machines in the same system*
- *Legacy: evolution and interworking of existing systems*
- *Scalability: low cost of computing per machine*
- *Decentralization: no single point of control*
- *.... these differences are fundamental*



Distributed systems are fundamentally different - Separation

- **Separation...**
 - remoteness
 - migration
 - no shared memory
 - partial failure
 - weak global consistency
- **...for example, the customer information systems of a regionally-organized company**



Distributed systems are fundamentally different - Diversity

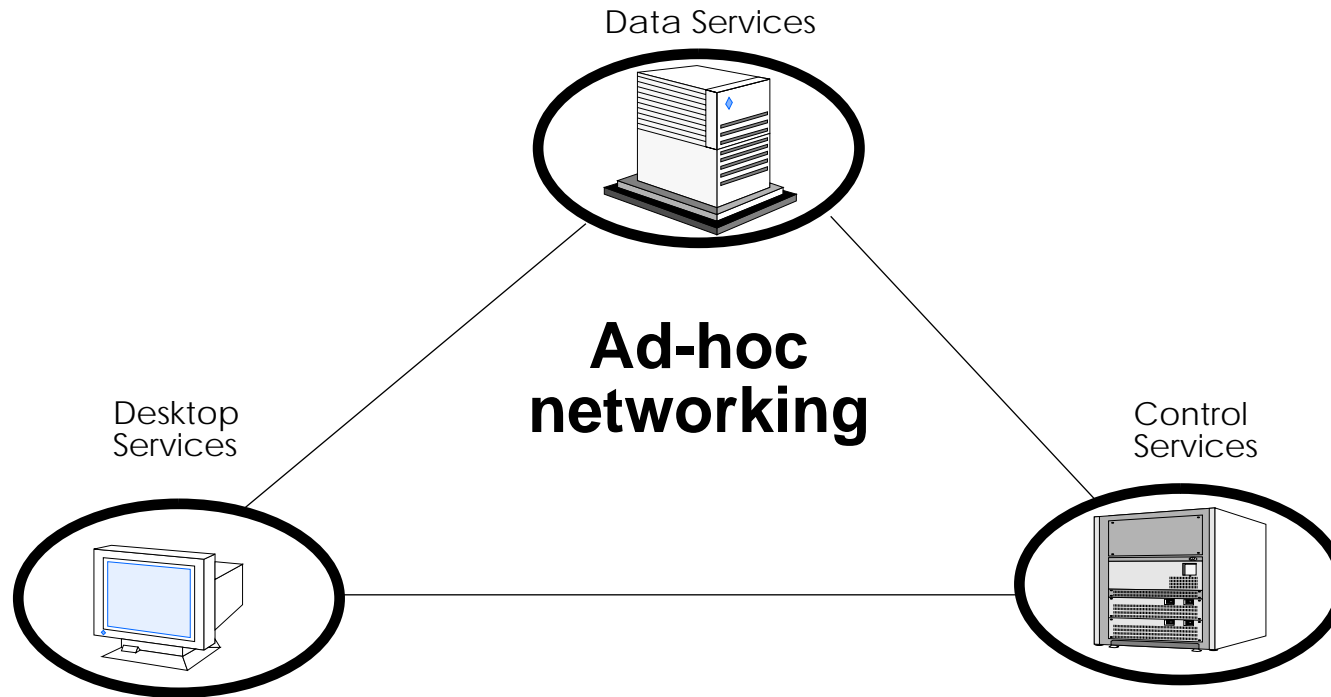
- ***Diversity...***
 - **diversity of scale**
 - **diverse data representations**
 - **diverse naming schemes**
 - **diverse hardware and software**
 - **diverse communications mechanisms**
- **...for example, the corporate LAN**



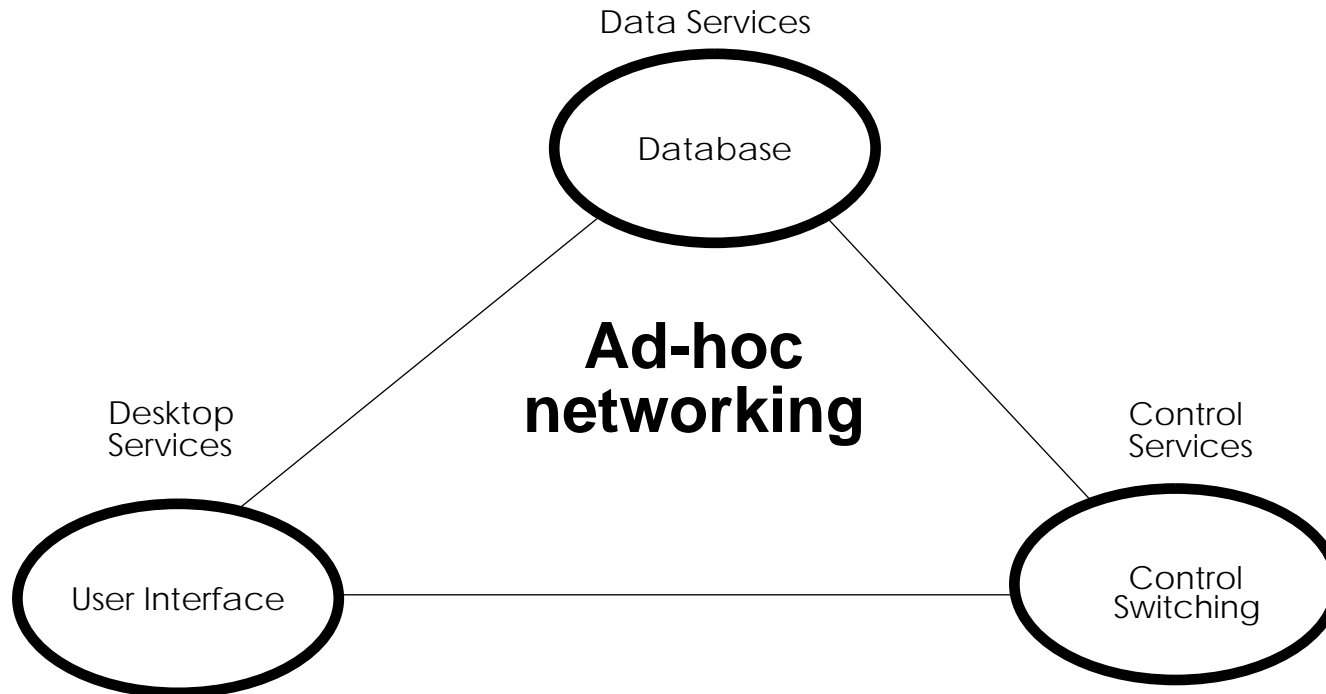
Distributed systems are fundamentally different - Federalism and Concurrency

- ***Federalism...***
 - no central authority
- ***Concurrency...***
 - simultaneous operation
 - multiple copies
- ***... for example, the Internet***

How distributed applications are built now



Typical skills needed to build them





Skills in the Data culture

- *Remote data access*
- *Distributed databases*
- *Stored procedures*
- *Object repositories*



Skills in the Desktop Culture

- *Individual PC productivity services*
- *Group PC productivity services*
- *File and printer sharing*
- *Mobile computers, universal personal digital communication*



Skills in the Control Culture

- *Device control*
- *Workflow*
- *Robust messaging*
- *Intelligent networking*



Possible solutions on offer?

- *Client-server*
- *Object-orientation*
- *Open systems*
- *Rightsizing*
- *... no single approach or technology will dominate*

These are not complete solutions, but they are useful



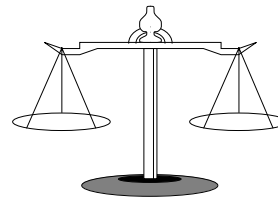
Different policies for different applications

- *Availability versus Consistency*
- *Autonomy versus Uniformity*
- *Security versus Convenience*
- *... and many other unavoidable trade-offs*



Distributed systems and coping with change

- *How do distributed systems help businesses cope with change?*
- *How do distributed systems balance*
 - *the demands of change...*
 - *...and the demands of continuity?*





The demands of change

- *Pressures for change make distributed computing necessary, as well as possible*
 - in the near future, most systems will be distributed
 - world-wide business requires world-wide systems
- *Information networks are the starting point...*
- *... how to build systems to coordinate information from many sources?*
 - diverse sources: old systems, new systems, and other organizations' systems
 - separate sources: from different places at different times



The demands of continuity

- *Preserving investment*
 - in people, and the legacy systems they use

- *Bridging the old and the new*
 - evolution not revolution



Distributed Systems have many aspects

- *Distributed systems involve many different people (the stakeholders)*
 - business managers, users, IT managers, IT developers,...
- *These people are concerned with different aspects of the system*
 - they see the system from a different viewpoint
 - each viewpoint is important
- *We need to be able to separate out these concerns when describing distributed systems*
 - so that each stakeholder can see that their needs are satisfied...
 - ... without being overwhelmed by descriptions of aspects that are irrelevant to them



The technical challenge

- ***Provide a framework for systems that:***
 - **integrate products from many vendors**
 - **are owned and managed by many organizations**
 - **can grow larger than the international telephone network**
 - **can evolve gracefully**
 - **allow different kinds of applications to interwork**
 - **preserve the investment in existing technology**
 - **have lower development and operating costs**
- ***... This framework is an architecture for Open Distributed Processing***

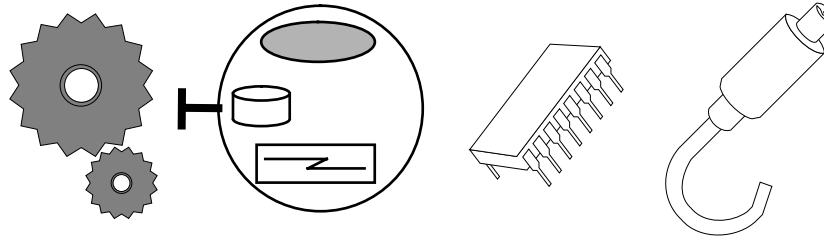


Other demands on the architecture

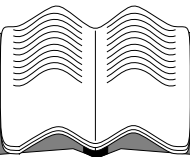
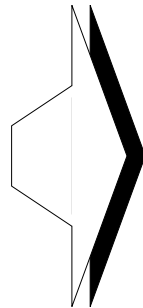
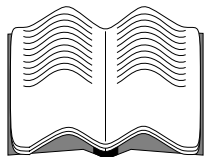
- *Must be easy to use and understand*
- *Must be widely applicable*
- *Must be durable and long-term*
- *Must be practical and proven*
- *Must be vendor-neutral*
- *Must be backed by the authority of international standards*

Architectural construction

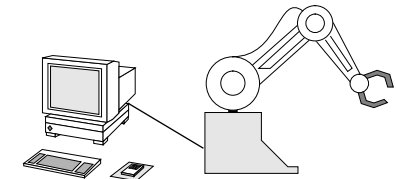
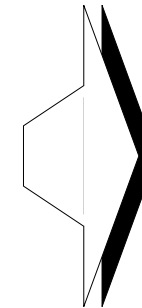
Basic building blocks



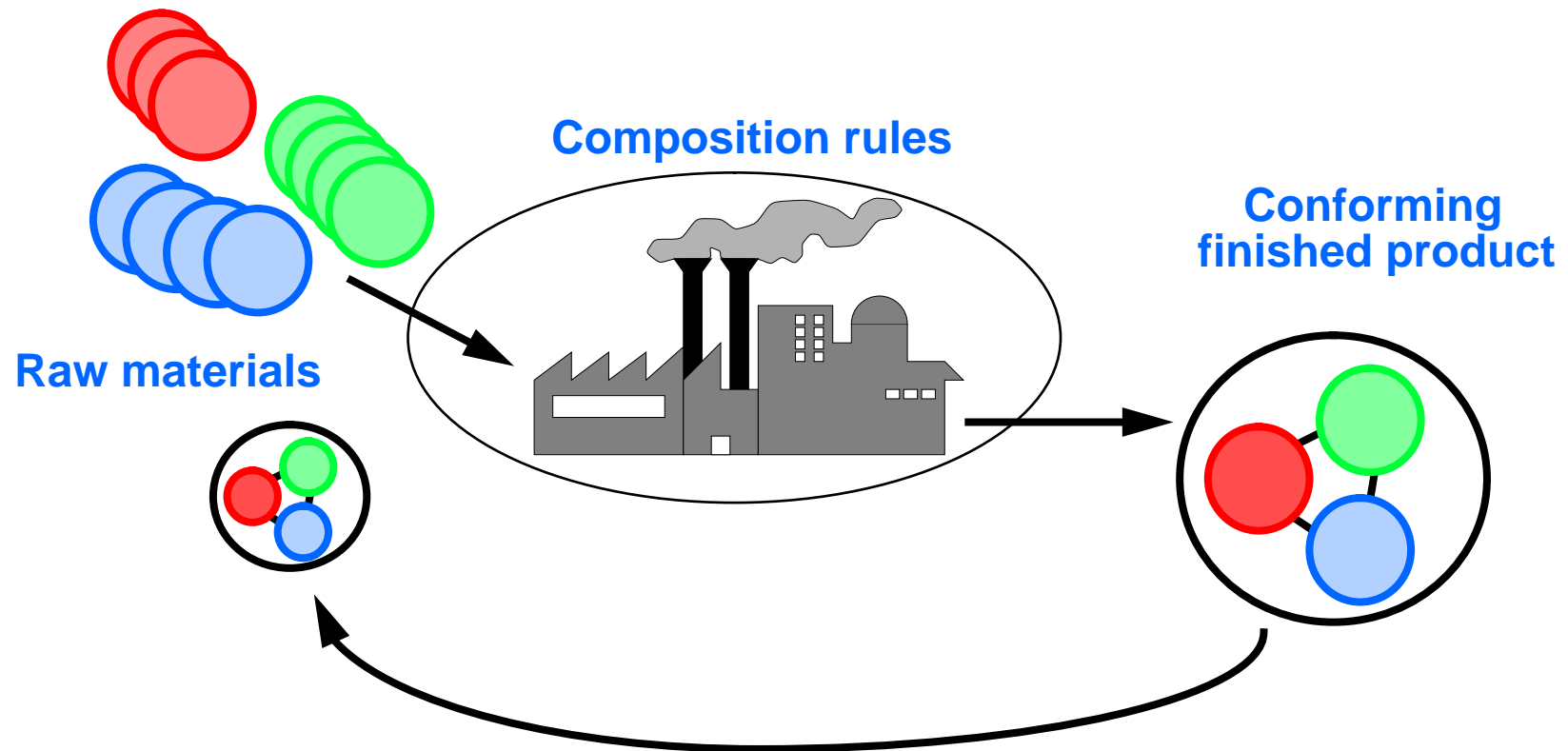
Combination rules



Recipes



Architecture for reuse





In the architecture there should be...

- ***Components***
 - standard functional building blocks, and tools to assemble them
- ***Rules***
 - embodying principles and assumptions
- ***Recipes***
 - for satisfying commonly-occurring requirements
- ***Guidelines***
 - for making design choices and trade-offs
- ***Concepts***
 - clearly defined and delineated



The architecture should leave you to decide...

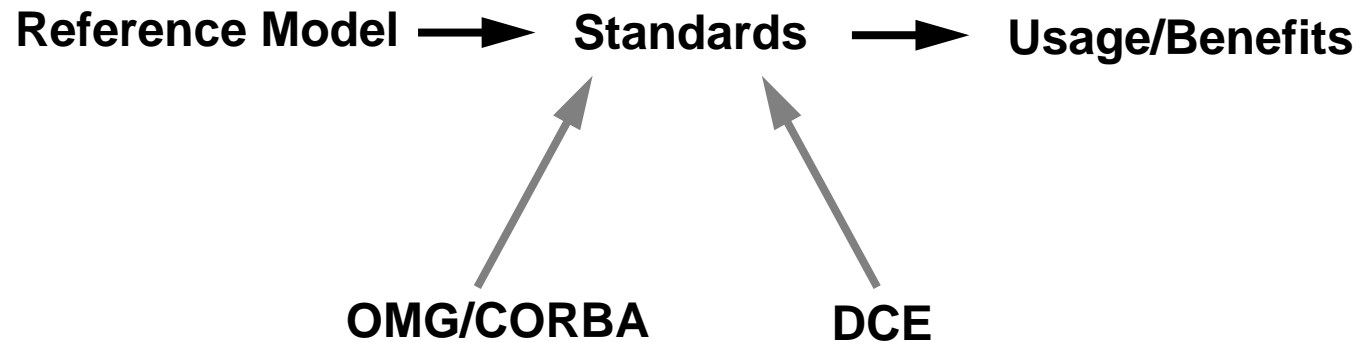
- *Which products to use*
- *Which software development methods to use*
- *Which user interfaces to provide*



RM-ODP

- *The Basic Reference Model for Open Distributed Processing (RM-ODP) is...*
 - an architectural framework for understanding the problems and concerns of distributed systems
 - a framework for assessing the conformance of a particular system
 - a forthcoming international standard
- *...a starting point for ODP standards*
- *Standardization is essential for openness to be achieved*

Relationship to other standards



- *Liaison is in place with OMG, and the framework is being populated*
 - *function correspondence has been identified*
- *CORBA and DCE are working bottom-up...*
- *...ODP is working top-down*



Architectural principles - Summary

- *Distributed systems have different properties to centralized systems*
- *Different applications need different solutions*
- *Unnecessary complexity should be masked from the applications*



Where next?

- *In this course we'll be exploring*
 - ...the ideas mentioned here
 - ...the CORBA distributed system architecture
 - ...and related fields