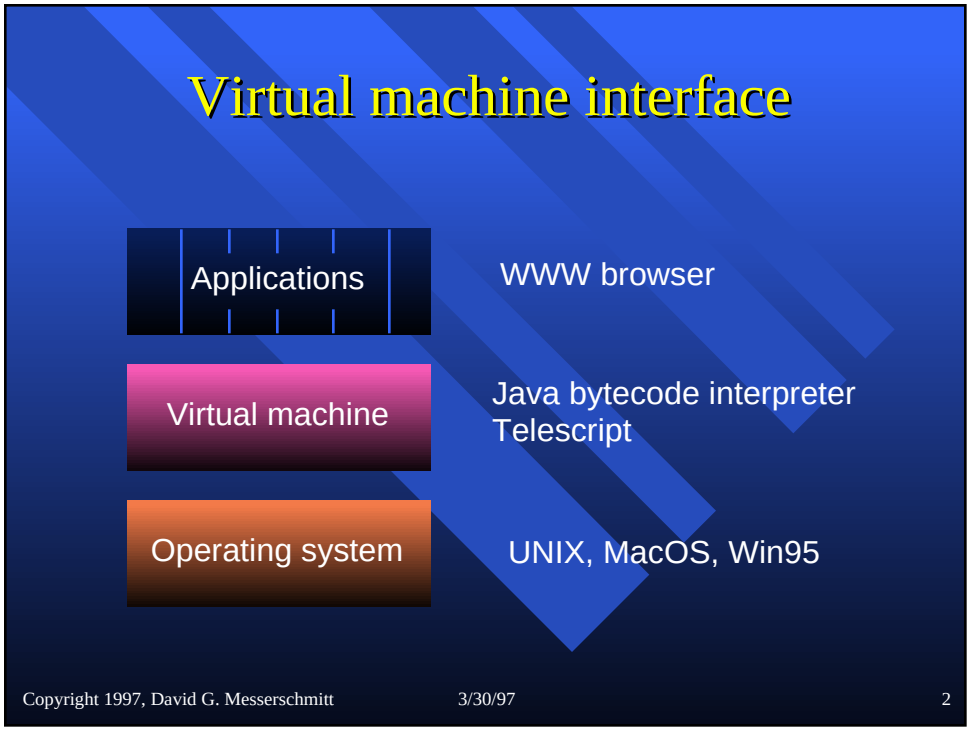
Important future technologies

- Java
- CORBA

David G. Messerschmitt

Department of Electrical Engineering and
Computer Sciences

In this lecture, we will describe two new technologies that may have a major impact -- Java and CORBA -- and then relate these to the NC.



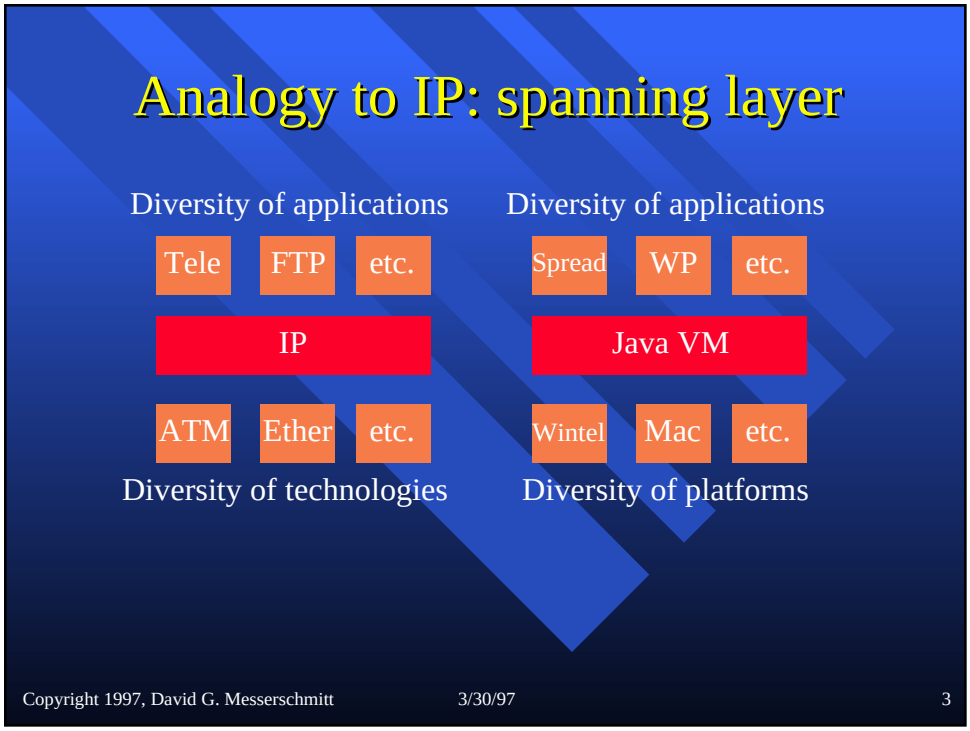
The virtual machine addresses some problems in network distribution of applications:

- Different instruction sets and operating systems (portability)
- Network and GUI APIs (portability)
- Security in downloading executables from untrusted sources (policies)

A price that is paid:

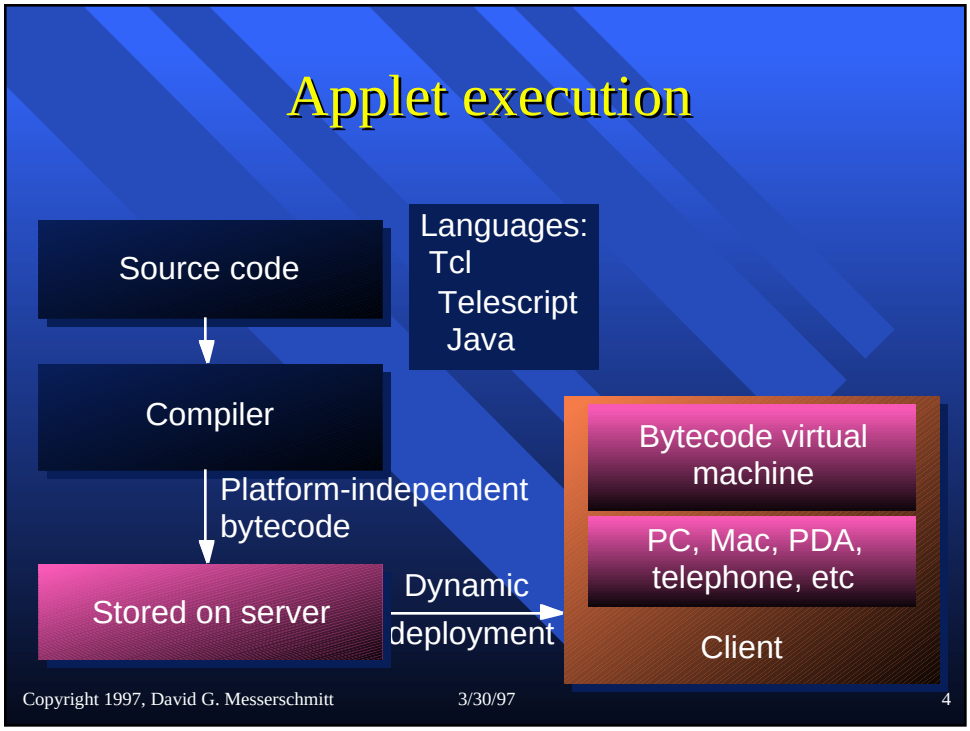
- 10X (perhaps down to 3X) performance hit for interpreted execution. This can be reduced using just-in-time (JIT) compilers and native Java processors.

Typically applications are written in Java, which offers features (such as threading and garbage collection) coordinated with the VM definition. However, in principle an application could be written in a different language as long as an appropriate compiler is available.



A “spanning layer”, as defined by David Clark of MIT, is a horizontal interface which can be assumed to be almost ubiquitous. Its significance is that it offers a layer of infrastructure that can be reasonably targeted by new products (from either above or below) with a large potential market. It separates the infrastructure into two modules (above and below) and attempts to make them independent of each other.

Of course, it cannot separate them from a performance perspective, only functionally. Typically there is an adverse performance impact due to introducing a new spanning layer.



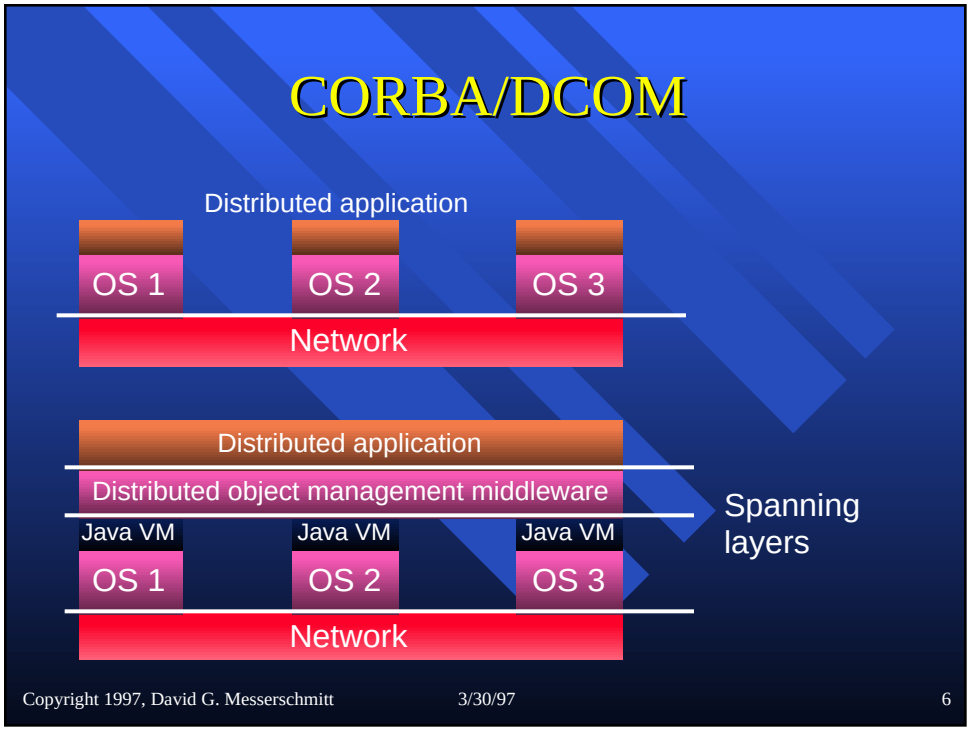
This illustrates how Java-described applications can aid scalabiity in client-server applications by transporting computation from server to client on demand.

Significance of Java

- Scalability
- Interoperability of distributed programs
  - Platform makes no difference
  - Version alignment
- Java VM offers a new spanning layer
  - Network effects reduced

In addition to scalability, Java contributes to interoperability by allowing distributed tasks to originate from a common code source.

Java also reduces network effects, since the universe of available users for a new application is the set of all compliant platforms, not users already possessing a pre-installed interoperable version of the application.

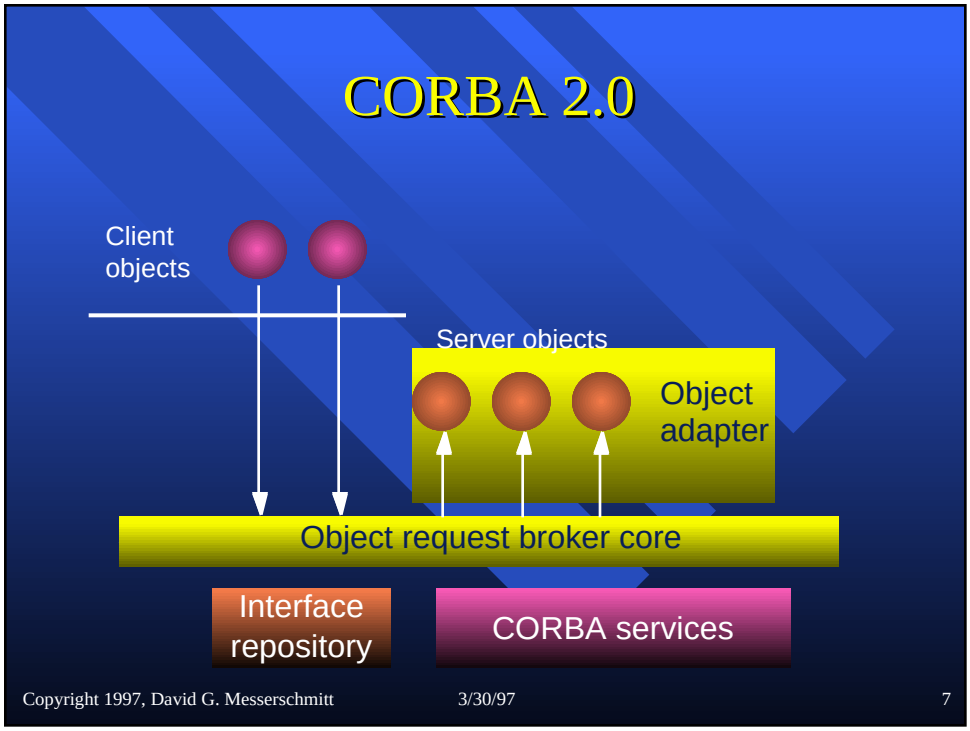


CORBA is an acronym for “Common Object Request Broker Architecture”, and is being advanced by the Object Management Group (OMG), a consortium of over 700 software companies.

DCOM is an acronym for “Distributed Common Object Model”, and is promulgated by Microsoft.

CORBA and DCOM are motivated by the needs of electronic commerce, where network effects are strong.

CORBA emphasizes interoperability, but currently does not emphasize portability. In this sense CORBA and Java are complementary, since Java emphasizes portability.



Client objects in CORBA do a “remote method invocation (RMI)” on server objects. (Under different circumstances, a single object instance can be either a client of server.)

The object adapter manages the life cycle of objects in its domain, such as instantiating them, destroying them, directing RMI’s at them, etc.

The core is a “universal software bus” that represents a spanning layer at the level of RMI.

CORBA offers many services that are of wide applicability, such as naming, security, etc.

The interface repository stores browsable descriptions of all object interfaces, argument and return value interfaces, etc. This is called object metadata, meaning “data about the data”.

What CORBA provides

- Java
  - Language bindings
  - Transportable objects
- Inter-galactic software bus
  - Cross-platform and language
  - Interoperability (but not portability)
- High levels of abstraction
  - Remote method invocation on objects

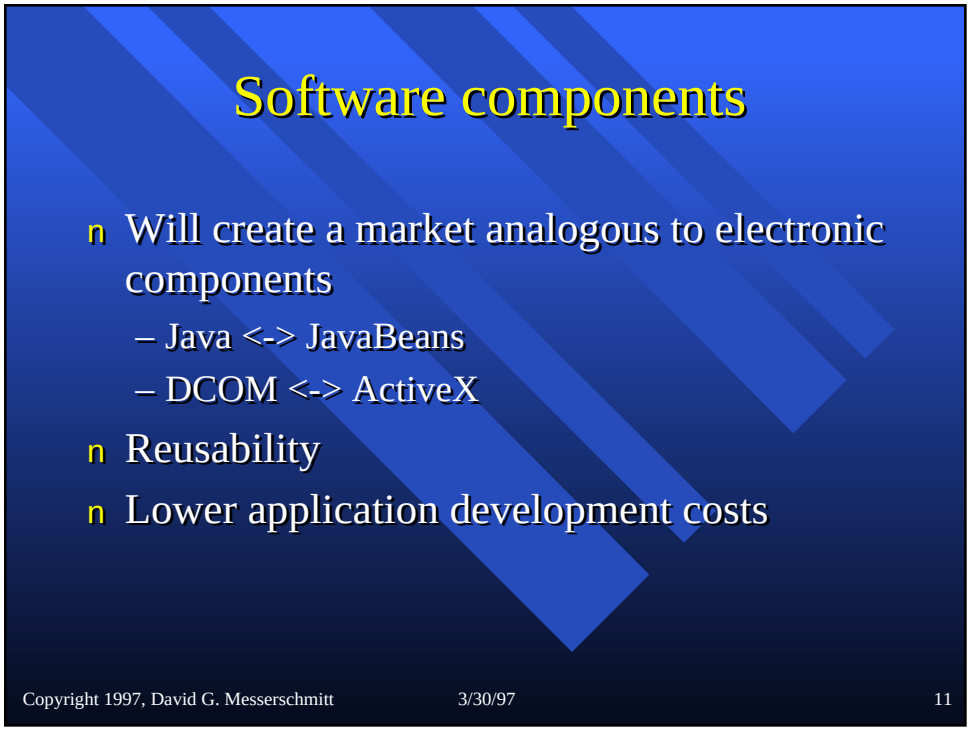
What CORBA provides (con't)

- Run-time flexibility
  - Everything self-describing
  - Interface discovery
  - Dynamic data structures and binding
- Useful services
  - Naming
  - Security
  - Many others

CORBA includes the ability to discover and dynamically bind to objects at runtime. This potentially provides great flexibility across a heterogeneous infrastructure.

Importance of CORBA

- Inter-enterprise computng
  - Platform and language independence
  - Electronic commerce, network management, etc
- Reduction of network effects
  - Another spanning layer
  - Significance of platform reduced



Software components add a new layer to the infrastructure by defining objects that provide functionality common to many applications. The objective is reusability through the sharing of common components, similar to the structure of the hardware industry for many years.

It is expected that a separate industry in components will develop.

Are Java and CORBA competitive or complementary?

- Both offer interoperabilty across different platforms
- Java offers portability and transportability
- CORBA offers heterogeneous language bindings
- CORBA offers many services, metadata, etc.
- Bottom line: they are complementary!

The major overlap between CORBA and Java is that they both enhance interoperability across a heterogeneous infrastructure, all they do this in very different ways. In other respects, they are complementary.