What is middleware?

A layer of software between the network and the applications that provides certain services …

Which services, and where to draw the “network” line, depends on what community you ask. What is commonly agreed on?

Middleware services are “distributed system services that have standard programming interfaces and protocols” Bernstein, “Middleware”

“The intersection of the stuff that network engineers don’t want to do with the stuff that applications developers don’t want to do.”

http://middleware.internet2.edu/overview/
What is the middleware sitting on?

A platform is “a set of low-level services and processing elements defined by a processor architecture and an OS’s API, such as Intel x86 and Win32…”

Bernstein, “Middleware”

Buzzword compliance

- What are the critical functions of middleware?
  - Who agrees on them?

- Everyone seems at least to agree that
  - There are different communication styles including
    - RPC / remote procedure call
    - OO (came from RPC)
    - MOM / message-oriented middleware
    - Transactional
  - There is a place that middleware “belongs”

What should middleware provide?

- Services
  - Information transport
  - Hiding heterogeneity & network
  - Directory/name services
  - Finding who you want to talk to
  - Security services
  - Authentication and authorization

- Properties
  - Reliability vs. performance
    - (Tradeoff?) (Is reliability a service?)
  - Scalability…
    - (Of increasing interest, but what does it mean?)

Issues – information transport

- Hiding “distributed” nature
  - Intrinsic limitations of each approach
    - E.g. RPC call-by-...

- Hiding heterogeneity: language, platform...
  - (More limitations here)

- Characteristics of transport
  - Suited for large or small information?
  - Latency?
  - TCP, HTTP, RPC, Mail, Instant messaging ..
Issues – continued

- Directory/name services
  - Separates addressing from naming (Grapevine82)
  - Distribution
  - Consistency
  - Scalability
  - Separate naming provider from naming service...
- Security
  - (Where to begin?)

Where did the “–ware” come from?

Before “–ware”

- White 1976, Liskov 1979
  - How can you communicate, generically?
- Nelson 1984
  - How can you make it efficient and secure?
- Navaratnam 1988
  - How can you make it reliable (but still efficient)?
- Oki 1993
  - How can you make it extensible w/out downtime?

“Implementing RPCs” Birrell, Nelson

- Information transport
  - Mimics local procedure call in Mesa on Dorados
- Directory/name services
  - Grapevine for binding importer to exporter
- Security services
  - Grapevine as authentication service; encryption
- Reliability, performance
  - Keep failure semantics similar to procedure call?
  - Optimized for small, frequent calls
- Scalability – ?
“Information Bus” Oki et al.

- Information transport
  - Subject-based publish/subscribe. Adapters convert data from heterogeneous applications.
- Directory/name services
  - Per-host daemon keeps local subscription lists
  - Pub/sub discovery for (quasi) RPC
- Security services – ?
- Reliability, performance – discussed
- Scalability
  - Integrate multiple instances of the bus
  - Extensibility (related to scalability?)

Delivering the promises

- Middleware products need to “get it right”
  - See standard sketch, they’re unavoidable
  - Where do the claims fall short?
- Critical review, formal analysis of products
  - Kevin Sullivan on COM (more than once)
  - JINI-on-CAN – it’s not independent of network
  - CORBA Type Safety – Crawley and Duddy
  - …

STOP