Meeting Performance and QoS Requirements with Embedded CORBA

Joey Garon
Senior Architect, Vertel Corporation
January 18, 2001

In embedded CORBA, can you have this:

You write only application logic

```
result = distObj->method(arg1, arg2);
```

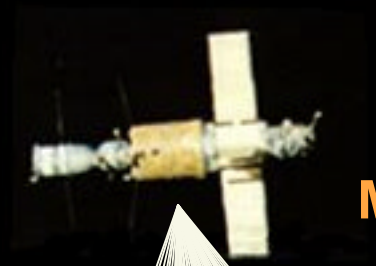
You don't write message encoding and decoding
You don't write transport APIs
You don't write request dispatching

…and still have these?

- efficient operation (speed)
- small footprint
- reliability (fault tolerance)
- scalability
- prioritization
- predictability
- security

vertel

2 Million Clients!?!?

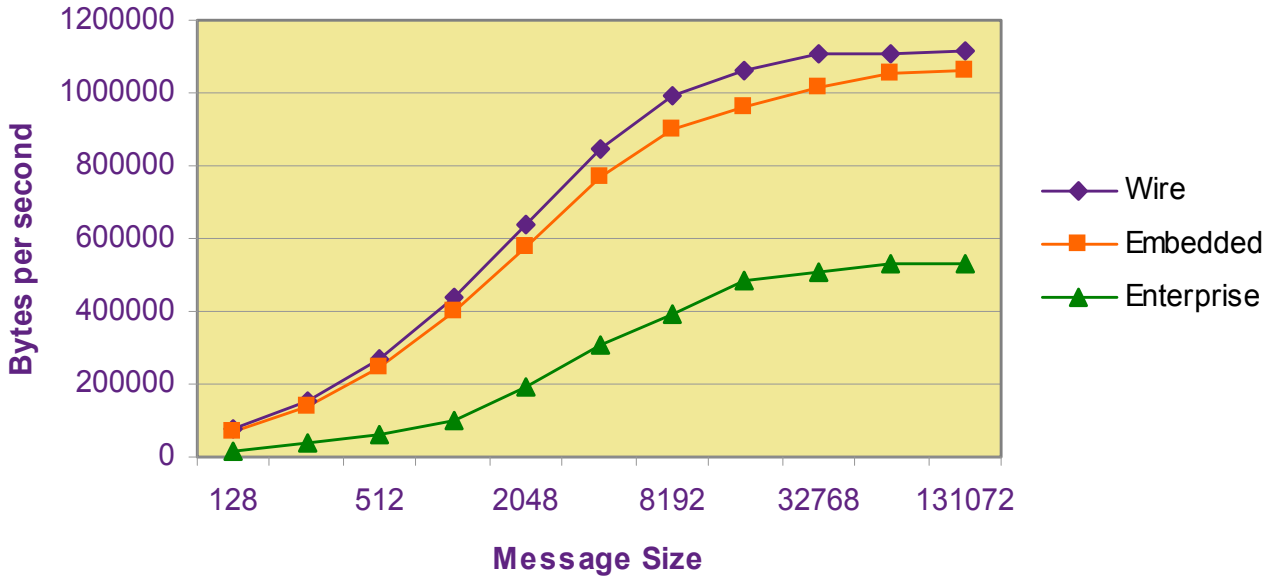


Massive load on server

Narrow bandwidth, and all the clients share the same spectrum

Set-top boxes need tiny footprint

Fast CORBA



- Embedded ORBs now run at 90% of wire speed
  - zero copy for fixed data types
  - constant-time demultiplexing on server
  - pluggable request-path features
  - platform-specific optimizations

vertel

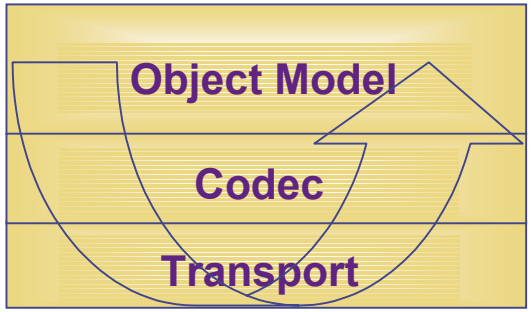
Small CORBA

- There are some *very* small ORBs now
  - Some are in the 100K range
  - Some are as small as 30K
  - Some ORBs are now available on PDA devices

- How is this possible?
  - Remove unneeded stuff: minimumCORBA
    - No DII, DSI, Dynamic Any, IFR
    - Reduced POA, ORB, and Object
    - Beyond minimumCORBA
  - ORB-designer cleverness
    - Platform-specific optimizations

vertel

Custom Transports, Codecs




```
result = distObj->method(arg1, arg2);
```

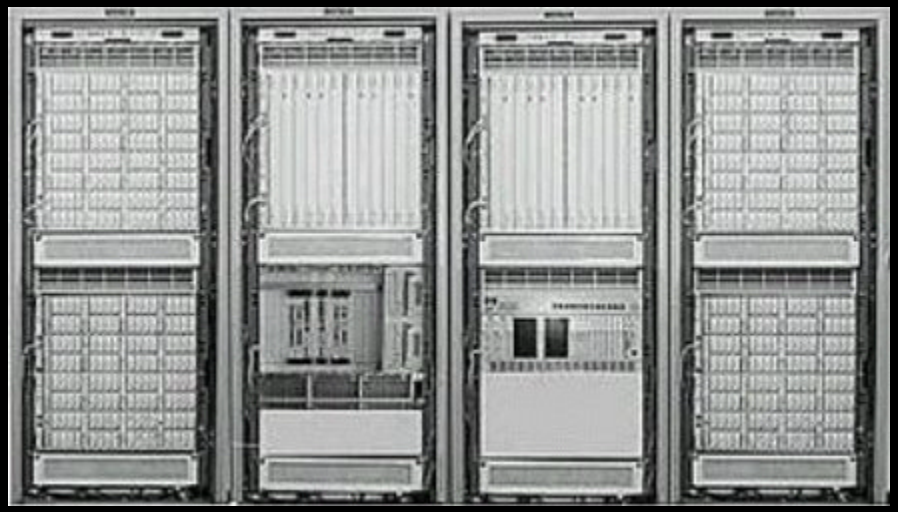
Message format (GIOP, ESIOP)

Comms transport (TCP/IP, U/RUDP)

- Custom transports let CORBA send gobs of messages over narrow-bandwidth medium
  - Reliable and unreliable UDP
  - Tiny footprint for protocol in set-top box
- Custom codec types
  - ESIOPs (environment-specific inter-ORB protocols) can compress the data carried by GIOP

vertel

Modern Optical Switches



Lots of cards (1000+) in the box, fast communication between cards

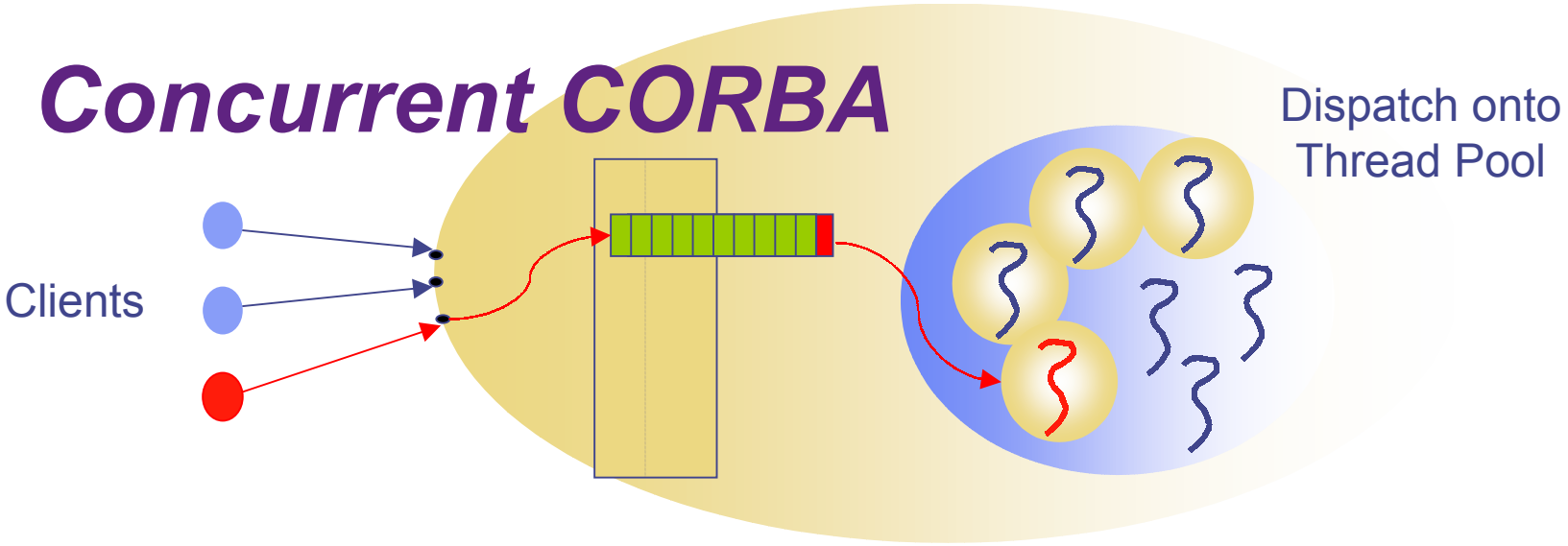
Complex hierarchy of thousands of objects

Alarm storms

24x7 operation, hot swap

Dynamic configuration

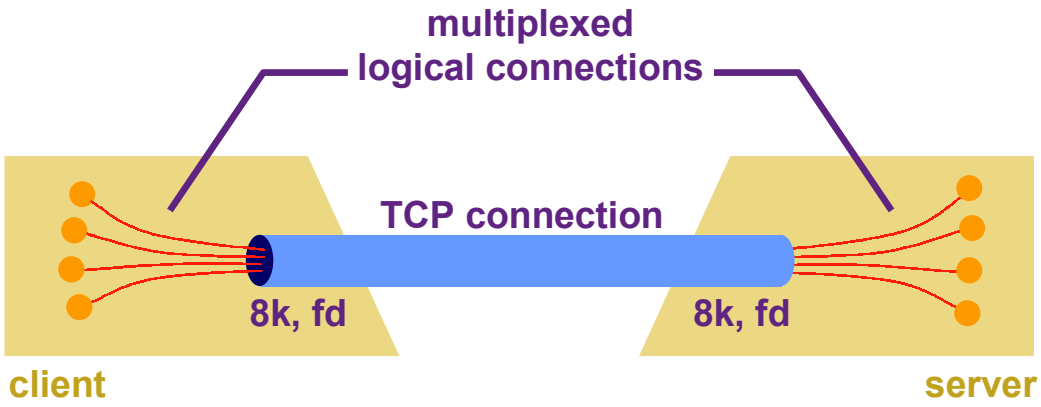
Concurrent CORBA



- A threading model is a rule that maps incoming requests to servant threads
  - Increased performance, because server can process multiple requests simultaneously
  - Managed threads by preallocating pools
- CORBA does not define threading models; many ORBs offer a variety of threading models

vertel

Shared Connections



- You can have more connections than fd's
  - Connections can remain open, no heartbeat
  - Eliminate per-request connection and thread
- Shared connections are transparent to code
- Most ORBs support shared connections

vertel

Connection Aging

- When server is at maximum number of connections, each new connection request causes the oldest to close automatically
- Bound total number of connections
  - Protection from runaway clients
- Server can retire connections automatically
  - The client recreates connection automatically when needed

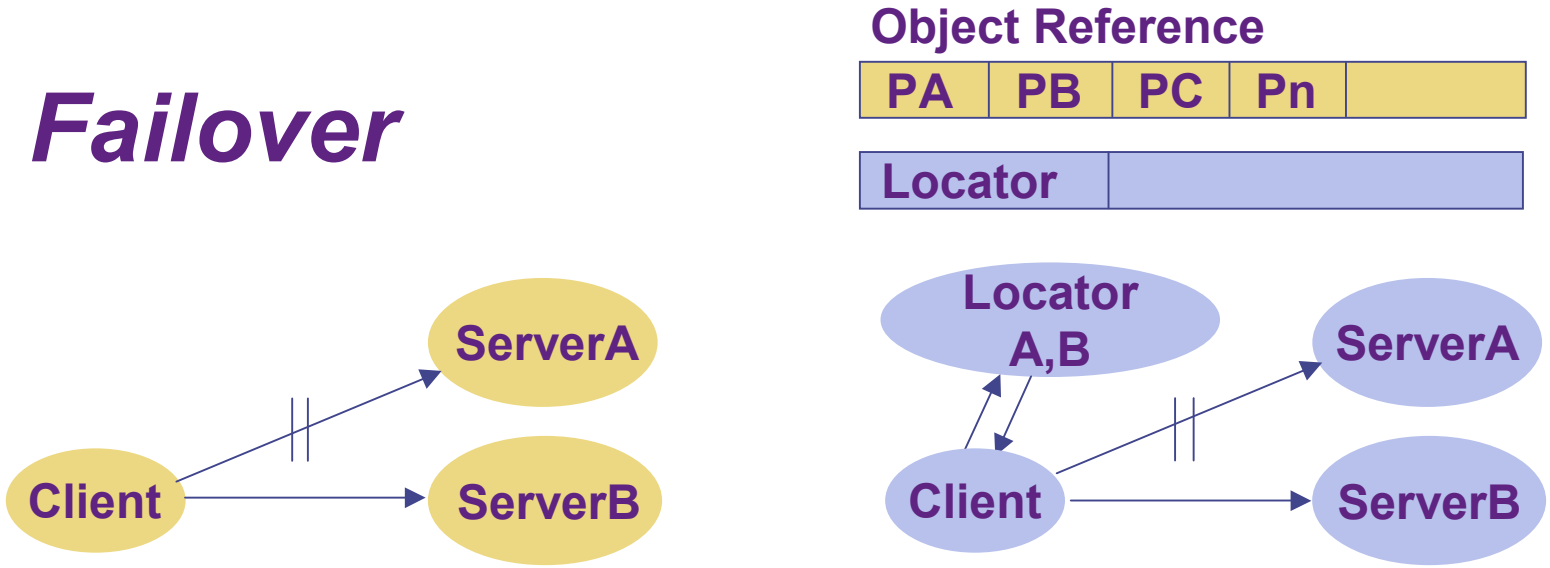
vertel

Timeouts

- Give up on request and ignore reply if received after time *t*
- Timeouts bound the amount of time to wait for distributed call to complete
- Prevents system lock up:
  - Busy server, Busy network, Buggy server

vertel

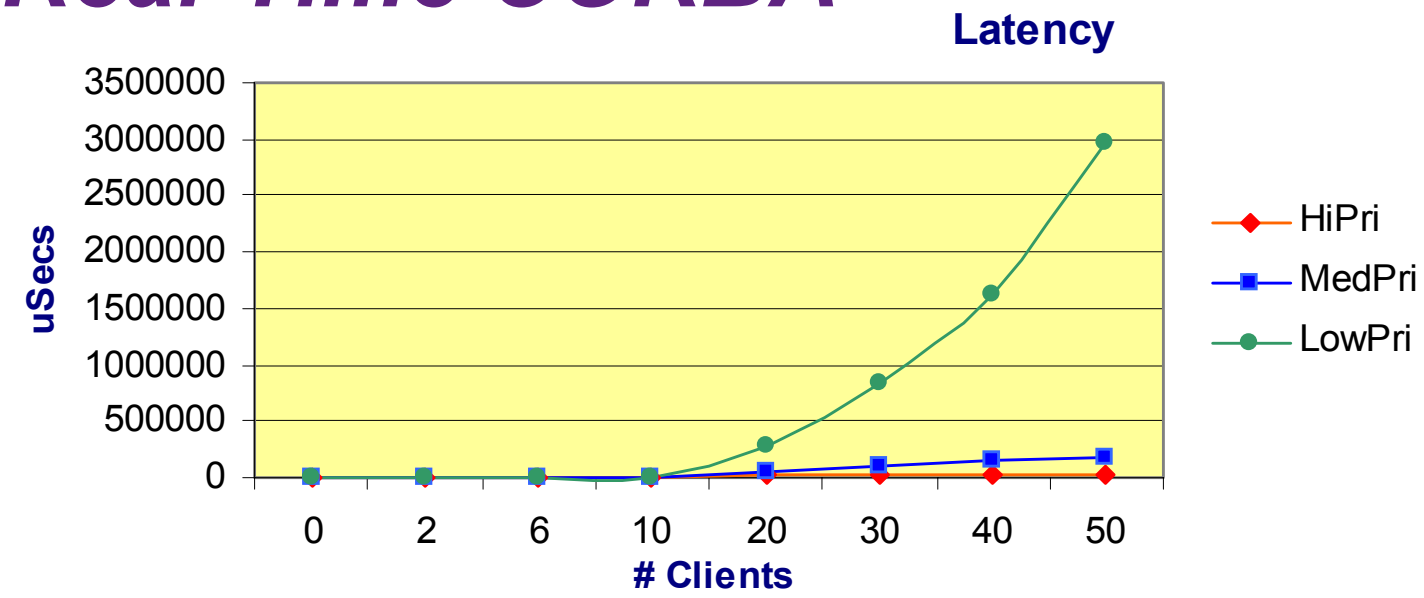
Failover



- Pull a card, stick a new one in, and the client never notices
  - 24x7
  - Upgrade without bringing down the system
- Failover is built into GIOP
  - Supported by a number of ORBs
  - Transparent to client code, but servers must share state

vertel

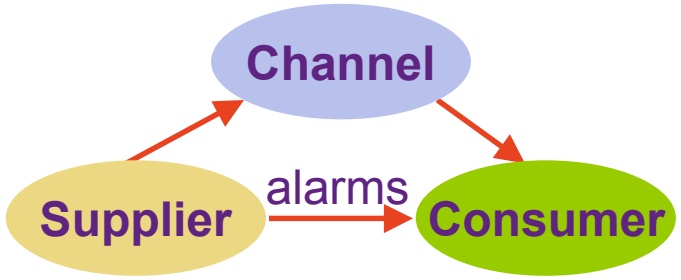
Real-Time CORBA

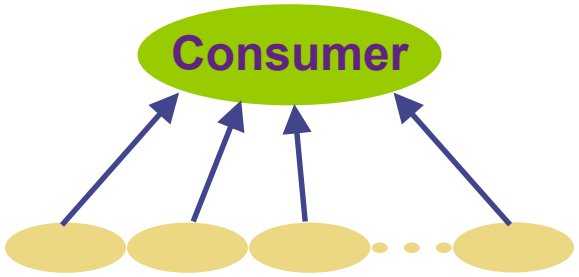


- Prevents priority inversions:
  - ORB dispatches high-priority messages before low-priority messages
  - If transport permits, high-priority messages travel ahead of low-priority messages
- Bound request latencies

vertel

Notification Service

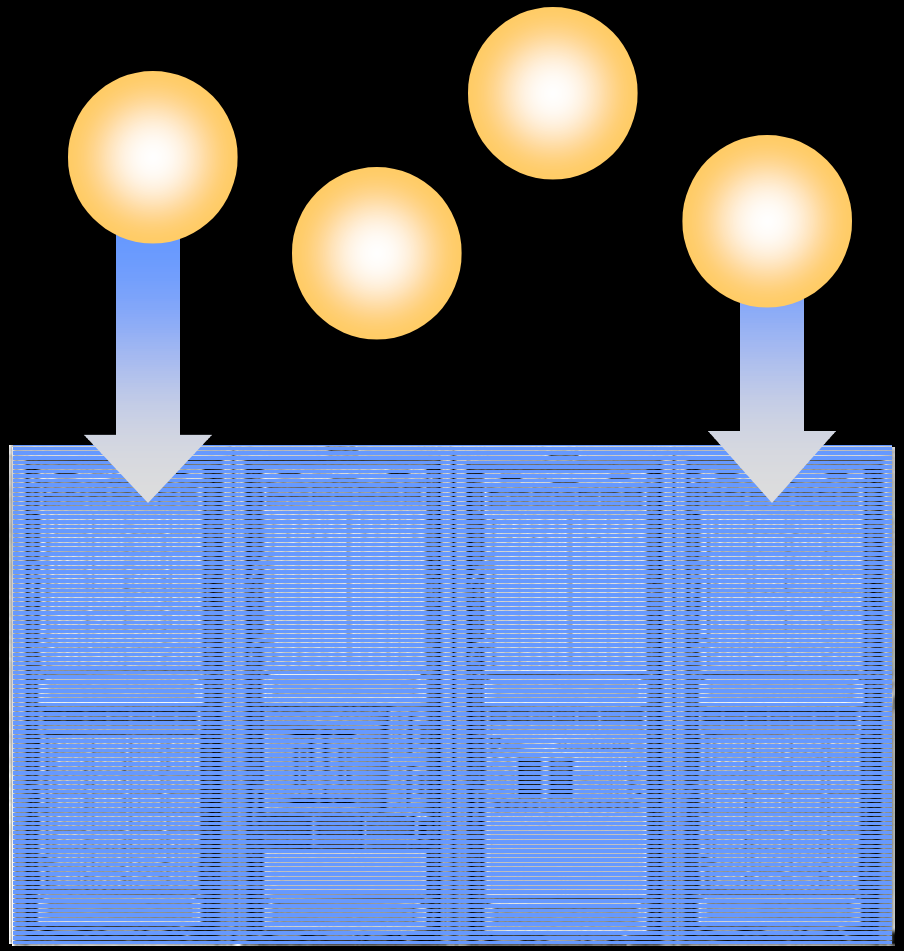




- Notification service filters and broadcasts events from “suppliers” to any “consumer” that requests to be put on the list
  - Suppliers and consumers don’t have to know about each other
- Direct notifications obtain transport QoS on one hop delivery
  - More efficient for alarm storms
- Event handlers can change on the fly (24x7)

vertel

CORBA-Induced Trends



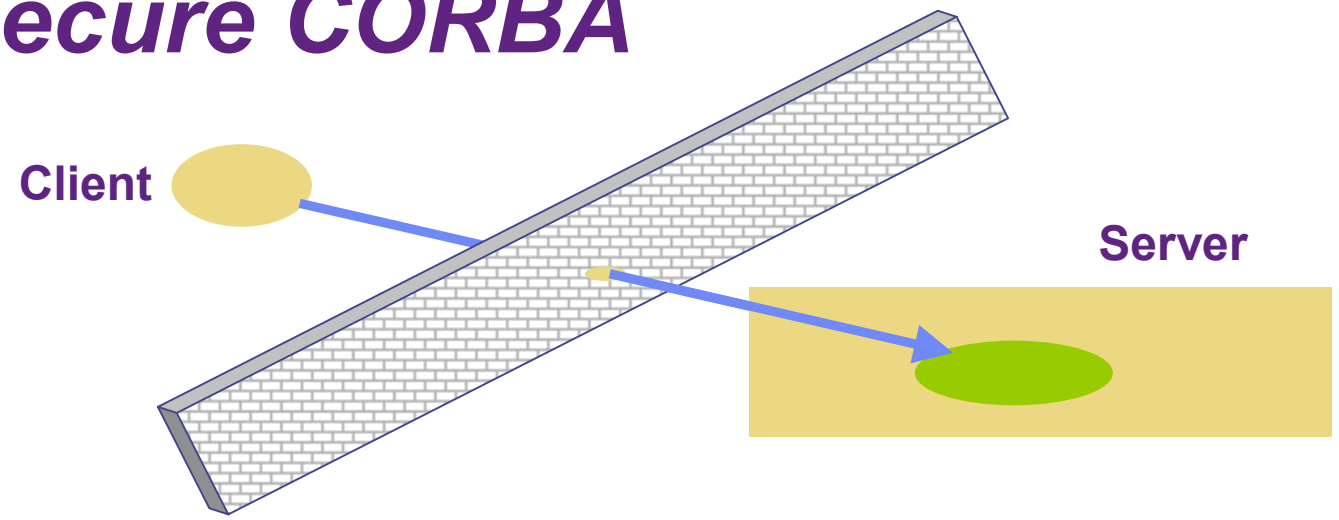
Network Elements controllable via network

EMS functionality split among different machines

Multiple agents

Finer-grained, more direct control over Network Elements

Secure CORBA



- CORBA can run transparently over SSL and firewalls
- Secure access to embedded devices over the Internet, transparent to your code
- Services can live on either side of the firewall

vertel

Why are we here?

- Explosive growth of Internet has fueled growth of new class of ultra-high bandwidth HW platforms (Broadband)
- Greater complexity, greater demands on flexibility, scalability, reliability, performance
- How do we deal with these trends *and* reduce time to market?

vertel

CORBA-Reduced Leadtime & Effort

- CORBA
  - software developed and tested in parallel with hardware
  - tested, stable, portable infrastructure
  - standard facilitates adding and training engineers
- IDL
  - eliminates message coding
  - strong type checking
    - fix interface errors on compile vs system test (1:10 hr/error)
  - interface errors resolved at HL design rather than at coding

vertel

Embedded CORBA

- Some current embedded CORBA-based technologies:

Optical Switching

Network Routers

Hubs

Telematics

Software-Defined Radio

Infomatics

Transportation

Avionics

vertel

Summary

- CORBA is being used to speed up embedded development and achieve:
  - high performance
  - small footprints
  - high reliability
  - high scalability
- CORBA-based products are flexible—they can work in a variety of QoS environments without code change

You can have your CORBA and embed it too!

vertel

Thank You

www.vertel.com