



Experiences with real-time embedded CORBA in Telecom

Shahzad Aslam-Mir
Principal software engineer

Vertel



Agenda

- Who is Vertel, what do we do ?
- What are we doing with CORBA in Telecom
- Where have we applied embedded CORBA
- What we have learned as a consequence
- Industry trends in real-time embedded CORBA
- Impact on core designs of our 3rd generation CORBA solutions
- Conclusions

Who is Vertel

- Embedded Network management solutions TMN
- Pioneers in bleeding edge CORBA solutions - Wash U.
- Leaders in producing absolute smallest and fastest CORBA - Palm, WinCE, Symbian, VxWorks, Compaq NSK - 'quantum CORBA'
- Typical customer base is Telecom carriers, switch and internet infrastructure makers
- Telecom solutions providers - including CORBA
- Core competencies - Wireless, Mediation, TMN, ASP and several others

quantum  CORBA by **VERTEL**®

CORBA in Telecom

- Telecom - hierarchical and complicated object models
- No 1:1 mapping, so point solution optimized mappings need to be deployed - GDMO, JIDM are top heavy
- Telecom uses a different blend of real-time and embedded solutions than say other industry sectors
- CORBA for NMS has been most popular
- Early adopters have not so much adopted real-time spec based solutions as they have the notion of embedded CORBA embedded inside their boxes.
- There are interesting ideas in Telecom on CORBA - inside, outside, and over the box

quantum  CORBA by VERTEL®

Popular questions asked by a Telecom



- **Maximum number of objects hostable by a CORBA server**
- **Maximum message rate on a blocking vs non-blocking transport**
- **Some measurand of performance vs. footprint**
- **QoS support possibilities**

Embedded RT CORBA

- **OMG has helped tremendously through Telecom vertical DTF, RT SIG, and 2 specifications**
- **Real-time CORBA specification**
- **Minimal CORBA specification BUT**
- **There is NO embedded CORBA Spec - ?**
- **Embedded has issues in addition to what is in RT Specification**

Industry trends

- Ultra high performance
- Reliability, fault-tolerance - longevity of operation
- Traceability of execution
- Hot-swap, dynamic swap and upgrades
- Accommodate multiple protocol stacks 'seamlessly'
- QoS enforcement from layer 7 - layer 3
- Deterministic resource control - memory, OS
- Card-OS-HW-BSP tuple neutrality
- Java-C++ hybrid structures (not JNI based)

Industry trends

- Embedded has a different profile - slightly different
- Often s/w waits on h/w - developer productivity
- Prototype on some other platform
- Often then there are point solutions - not extensible, reusable
- Object solutions are relatively new
- Portability has hitherto been difficult - tied to processor-board-OS-bsp tuple
- Distributed debugging has been difficult - limited tool sets

Industry trends

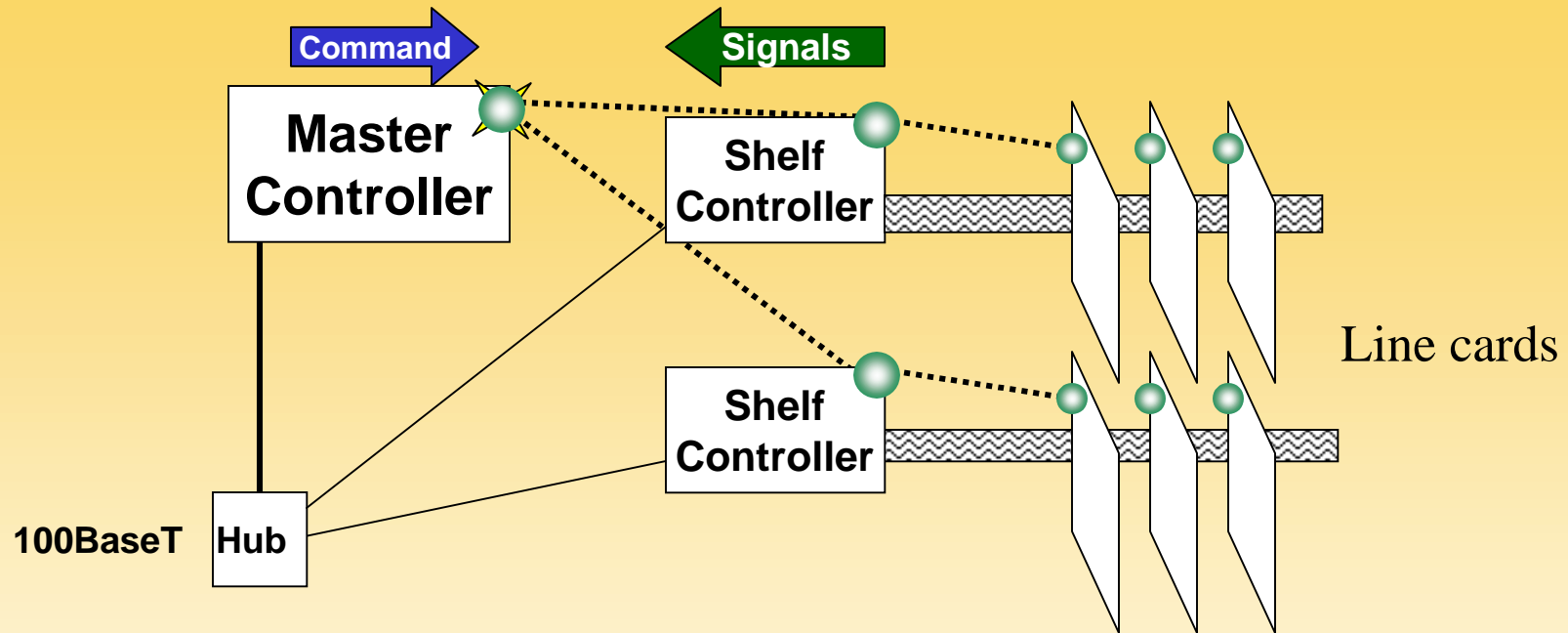
- To a large extent, embedded industry is still a hand rolled tcp/udp, C and worry about interrupts culture
- Where object solutions are deployed classical CORBA ideas have tremendous attraction - e.g. location transparency, platform, language, OS neutrality
- Telecom network h/w design has a unique fingerprint -

Telecom Network H/W Design

- Hierarchical - master, shelf, line card
- Duplicated controls, high availability
- N+1 protected line card architectures
- Powerful processors, e.g PowerPC on each line card
- Sophisticated, highly optimized IPC over shelf back planes, and Multiple back planes for HA
- Growing use of COTS middleware
- Use of 3rd party RTOS and s/w components
- Some use of LAN technologies for inter shelf comms
- Severe design cycle compression

quantum  CORBA by VERTEL[®]

Hardware architecture for Telecom NEs



COTS middleware
and h/w technologies

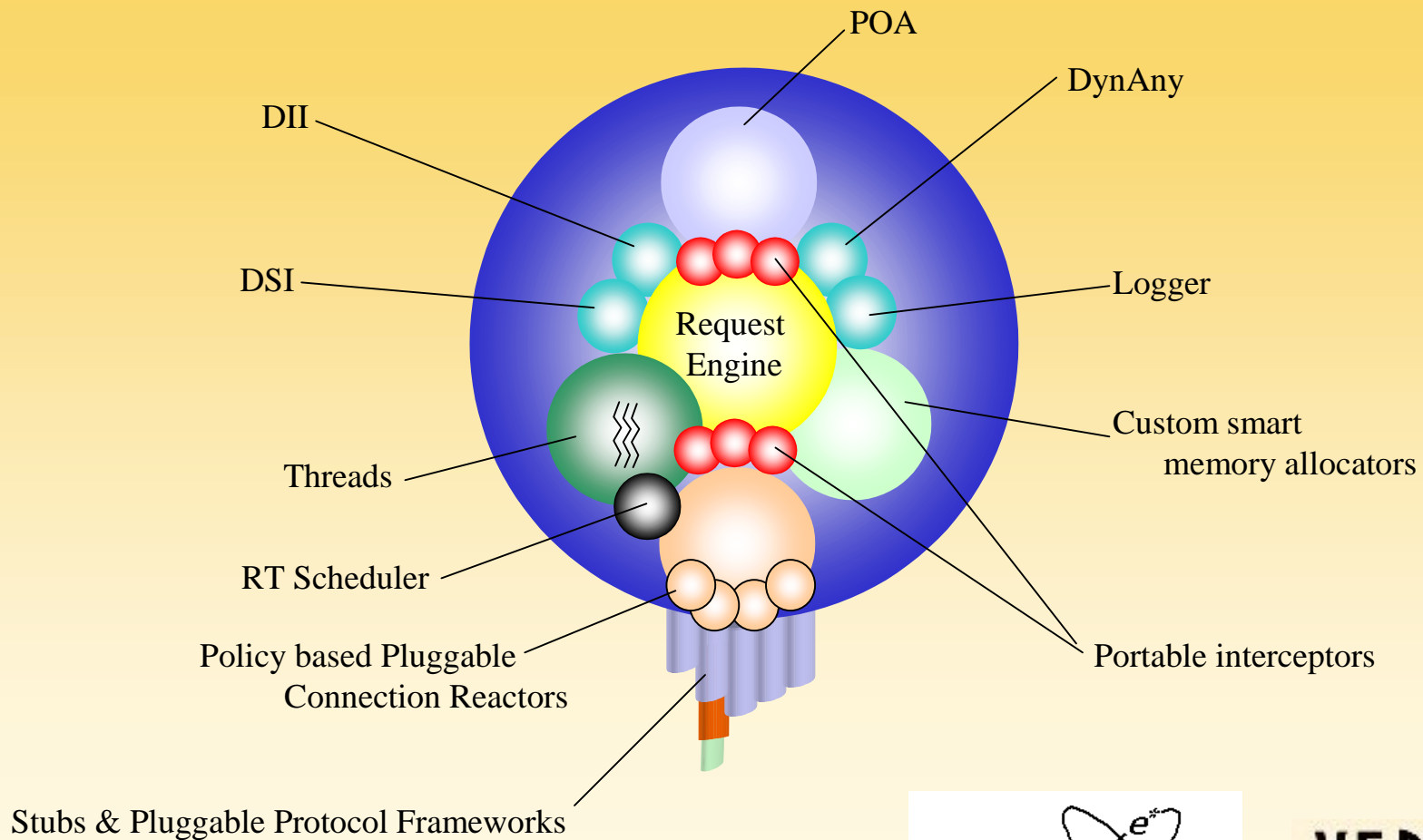
quantum ^{e⁻}CORBA by VERTEL®



**Lessons learned
developing a 3rd Generation
embeddable real-time
CORBA ORB**

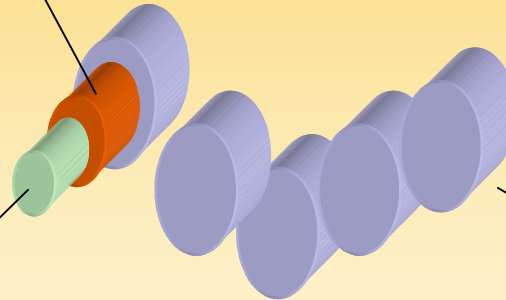
quantum  *CORBA* by **VERTEL**®

Simplified anatomy of an ORB



Pluggable Protocols

Open framework for
pluggable Message Format

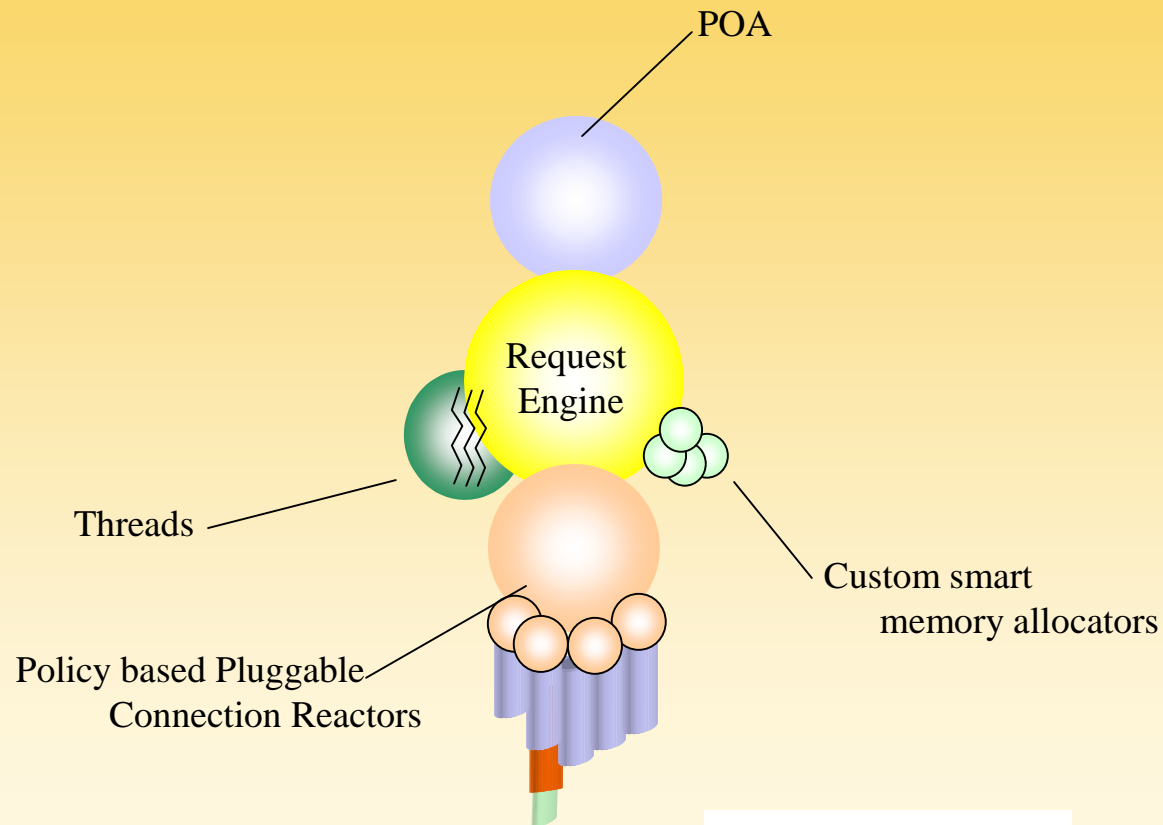


Open framework for
pluggable transport protocols

Compact Code
minimal static and
runtime footprint

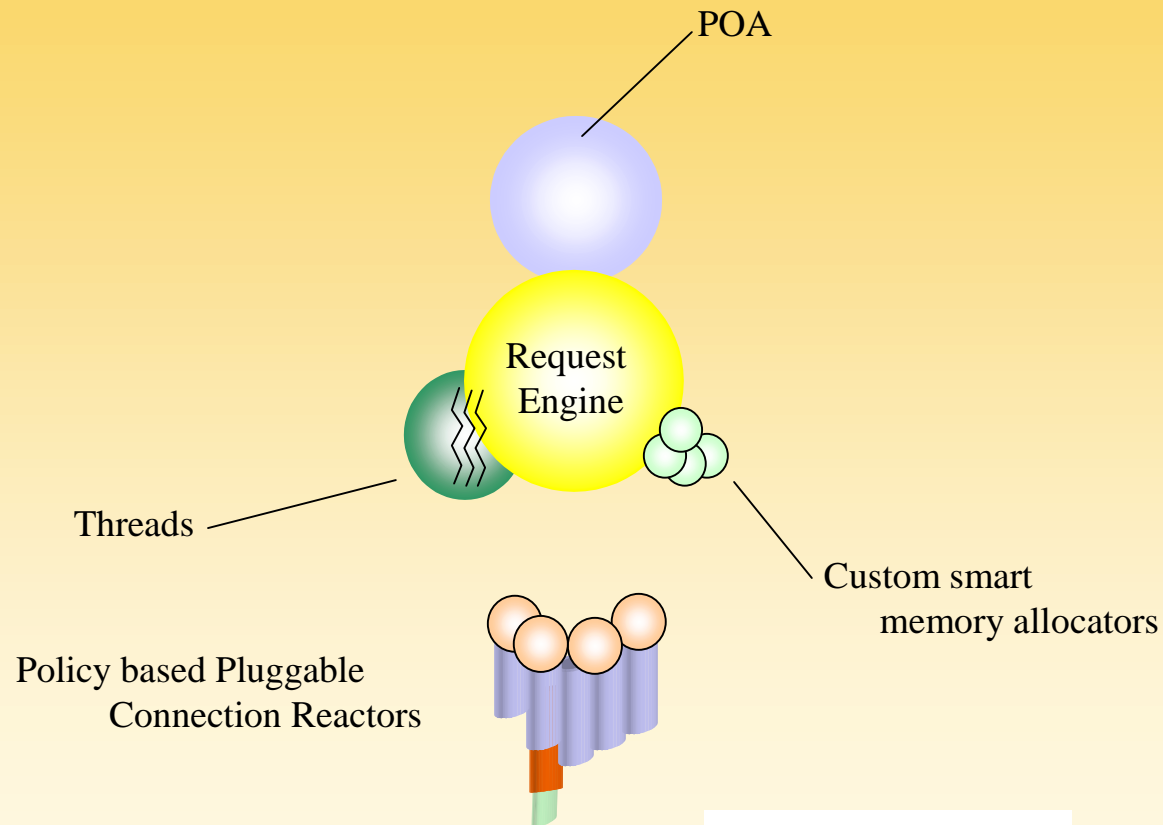
quantum  CORBA by VERTEL®

Basic internals that should be open

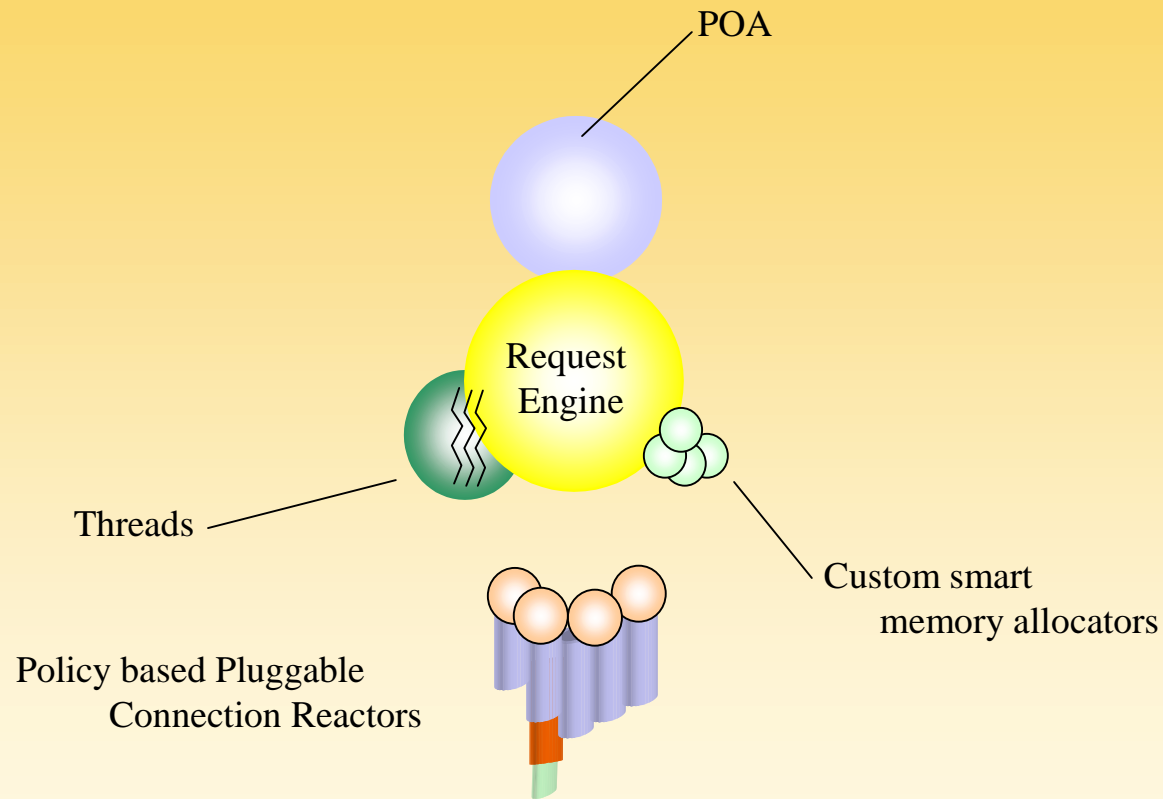


quantum e^+ CORBA by **VERTEL**®

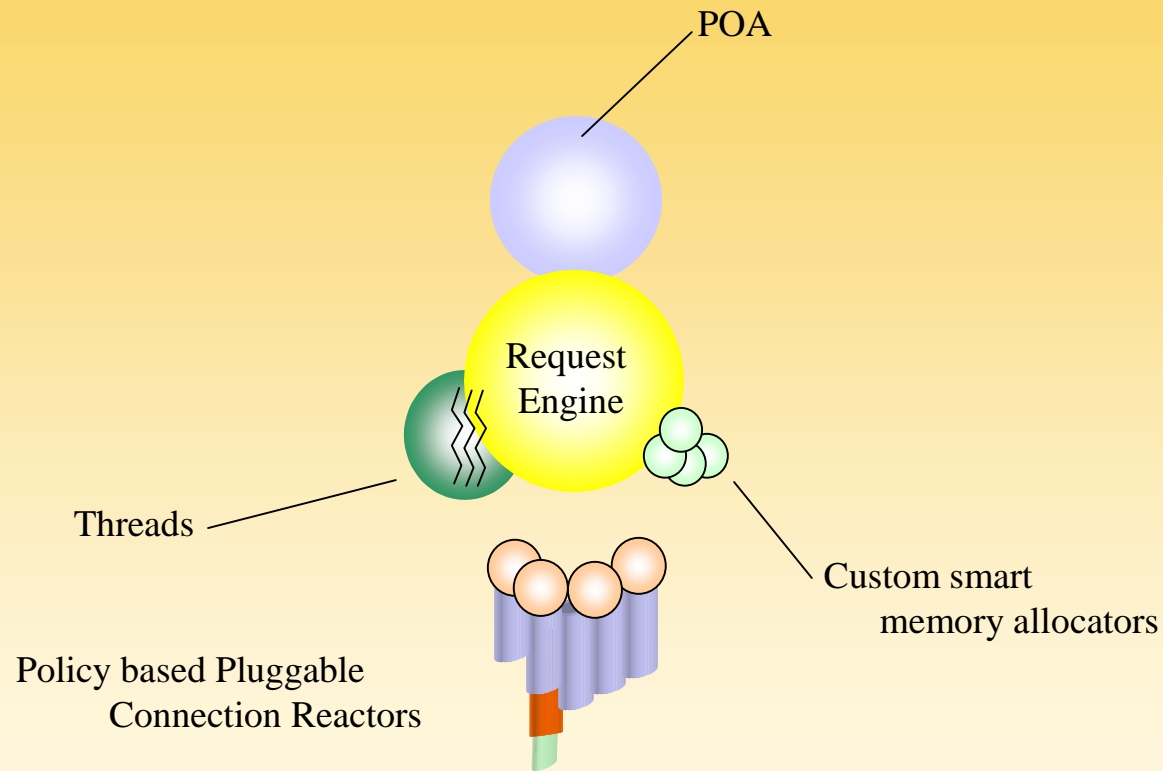
Basic internals that should be open



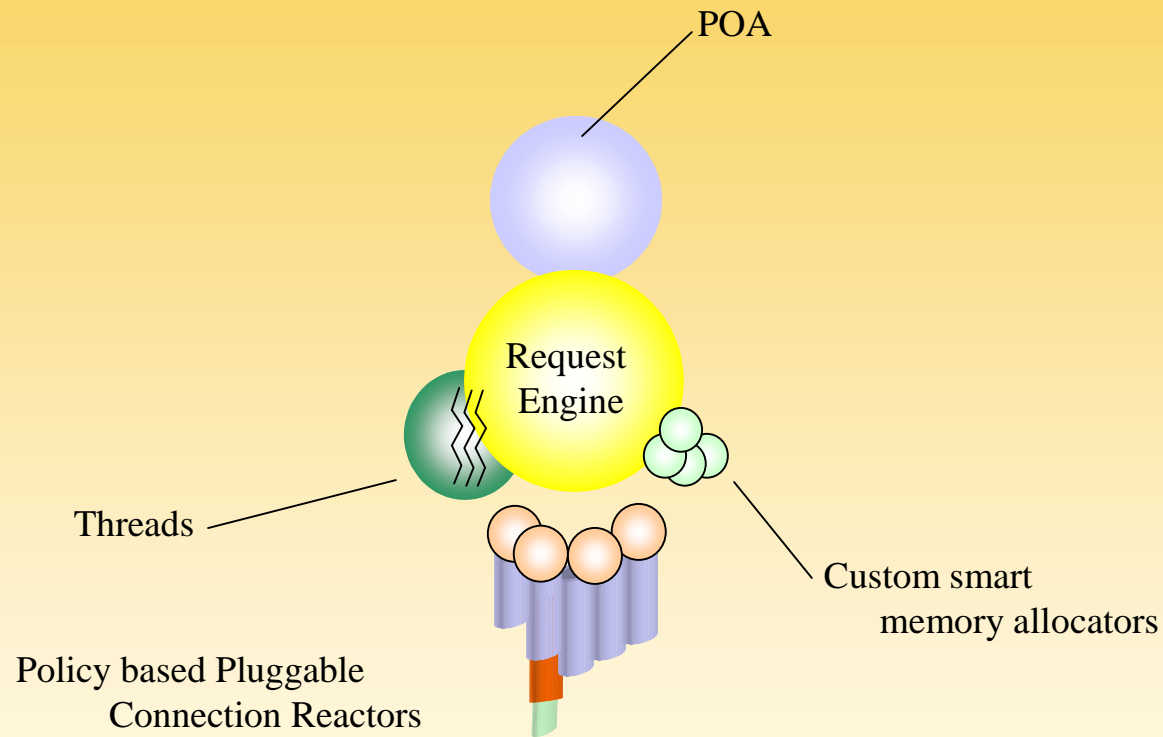
Basic internals that should be open



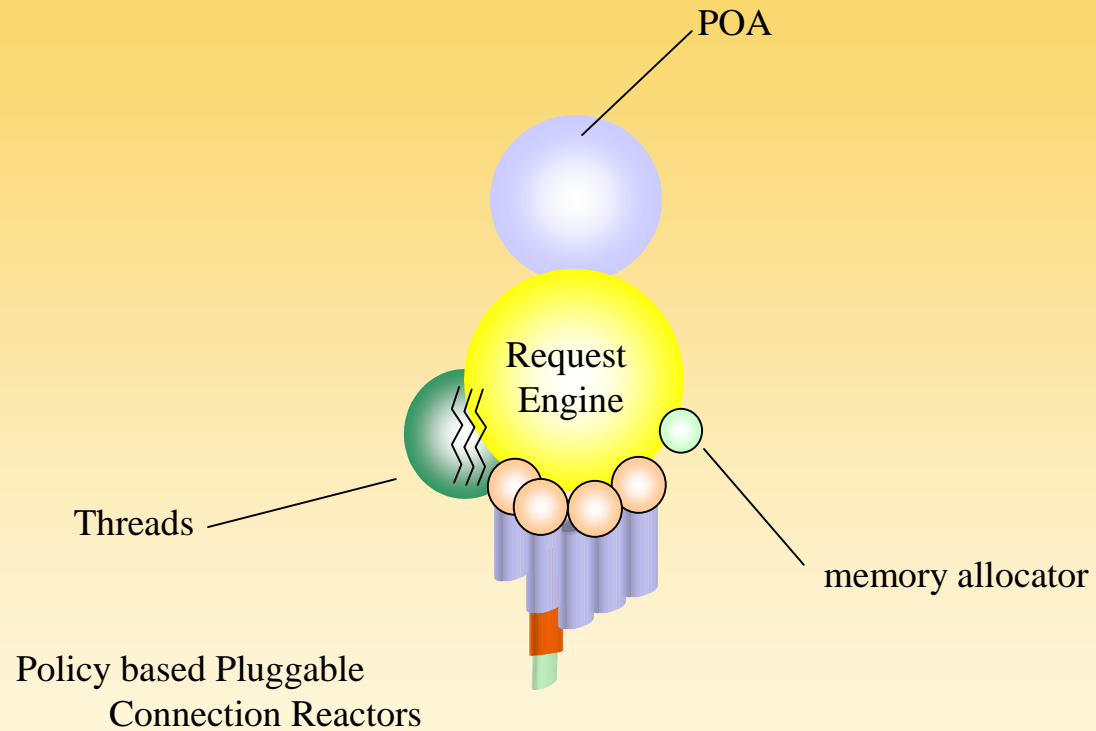
Basic internals that should be open



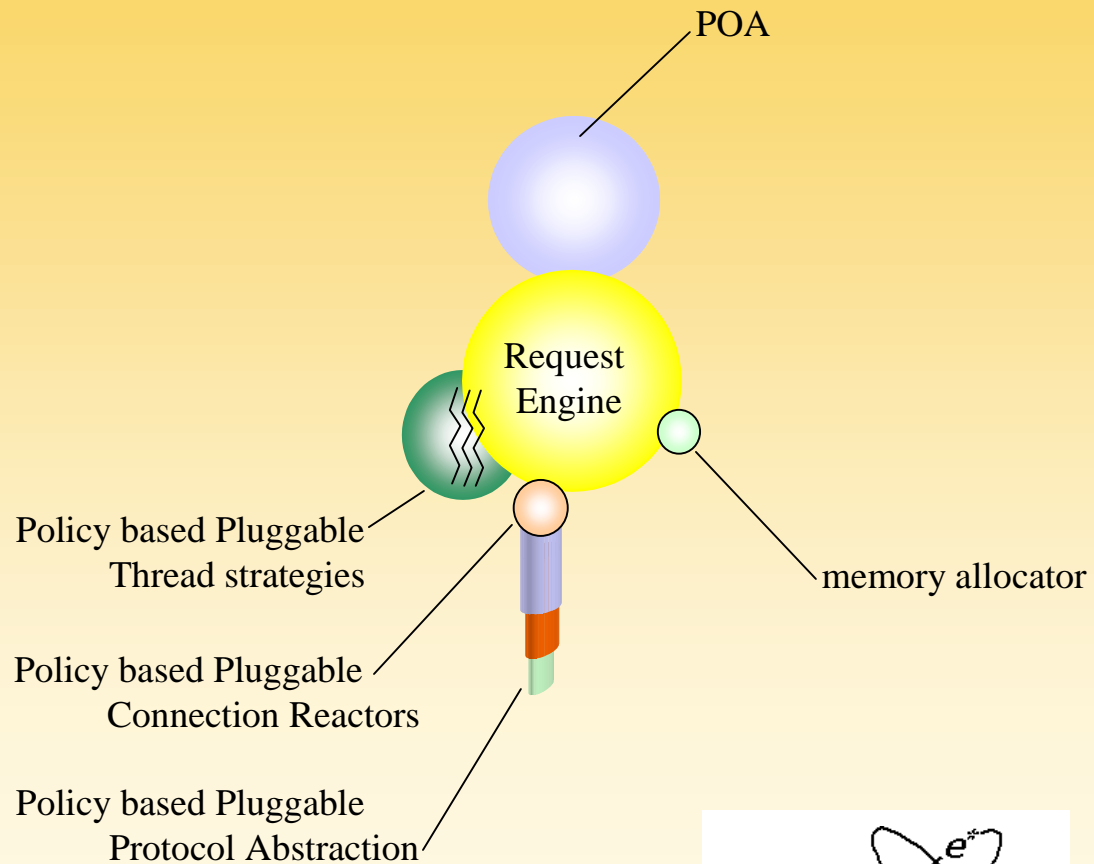
Basic internals that should be open



Basic internals that should be open

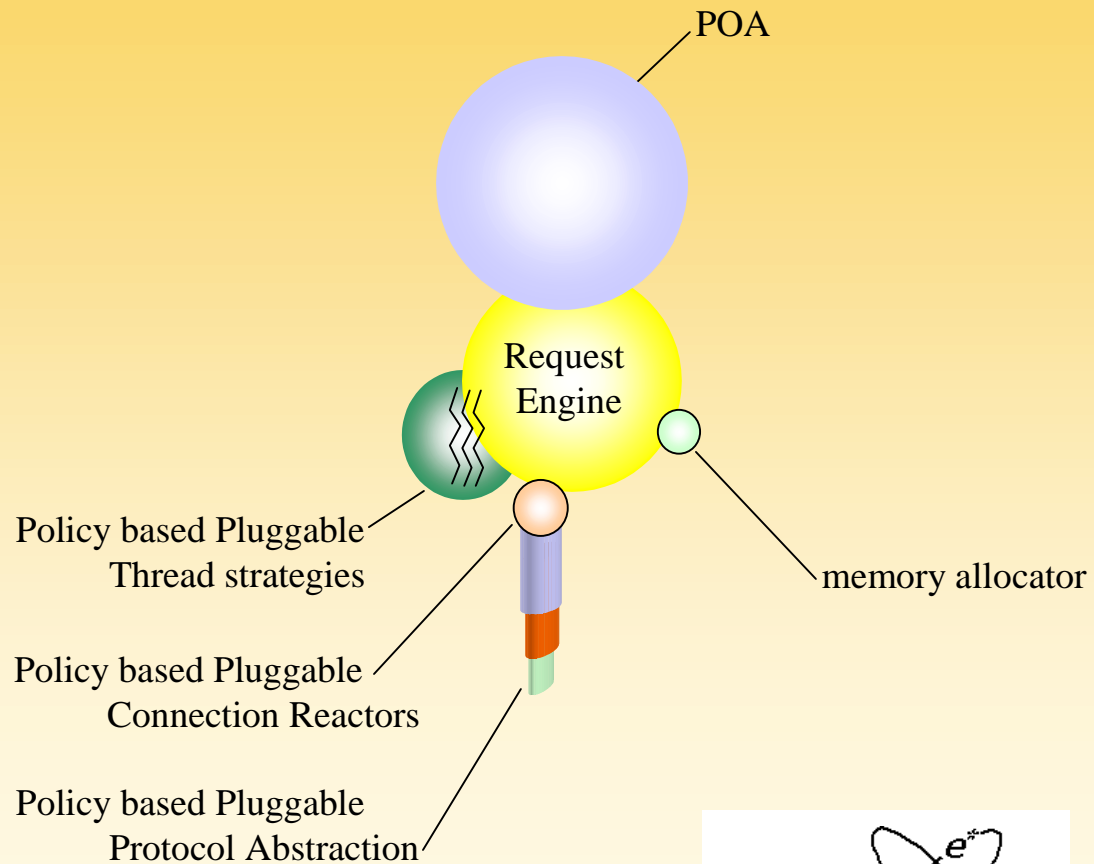


Basic internals that should be open



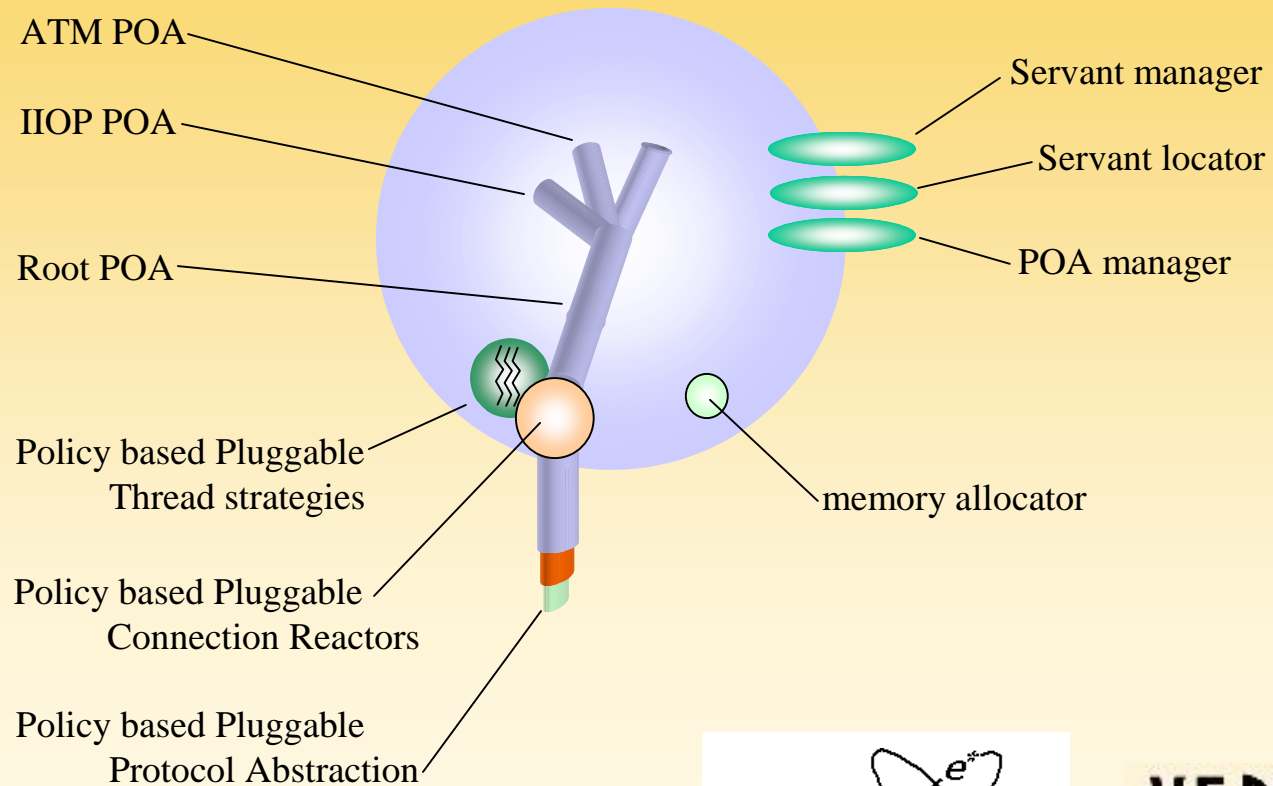
quantum CORBA by **VERTEL**®

Basic internals that should be open



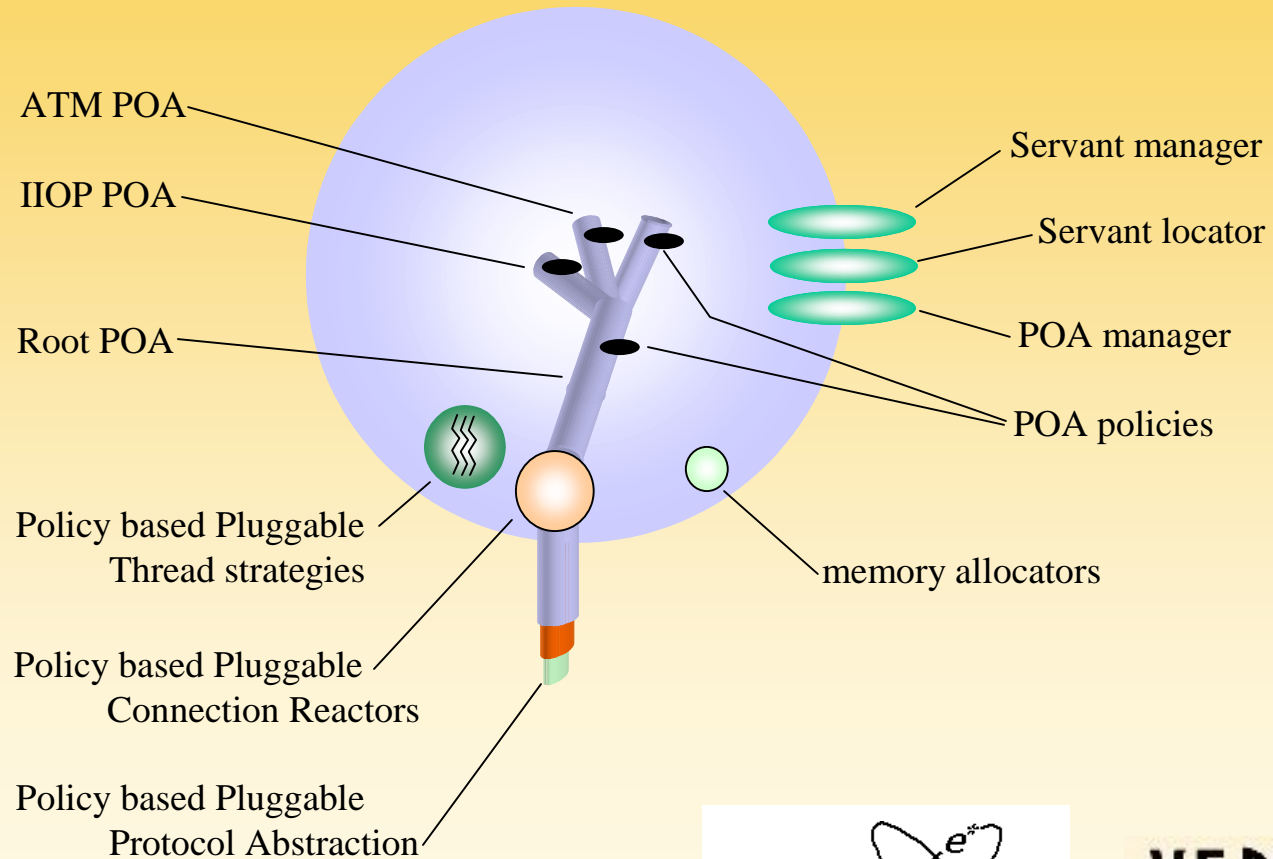
quantum CORBA by **VERTEL**®

Basic internals that should be open



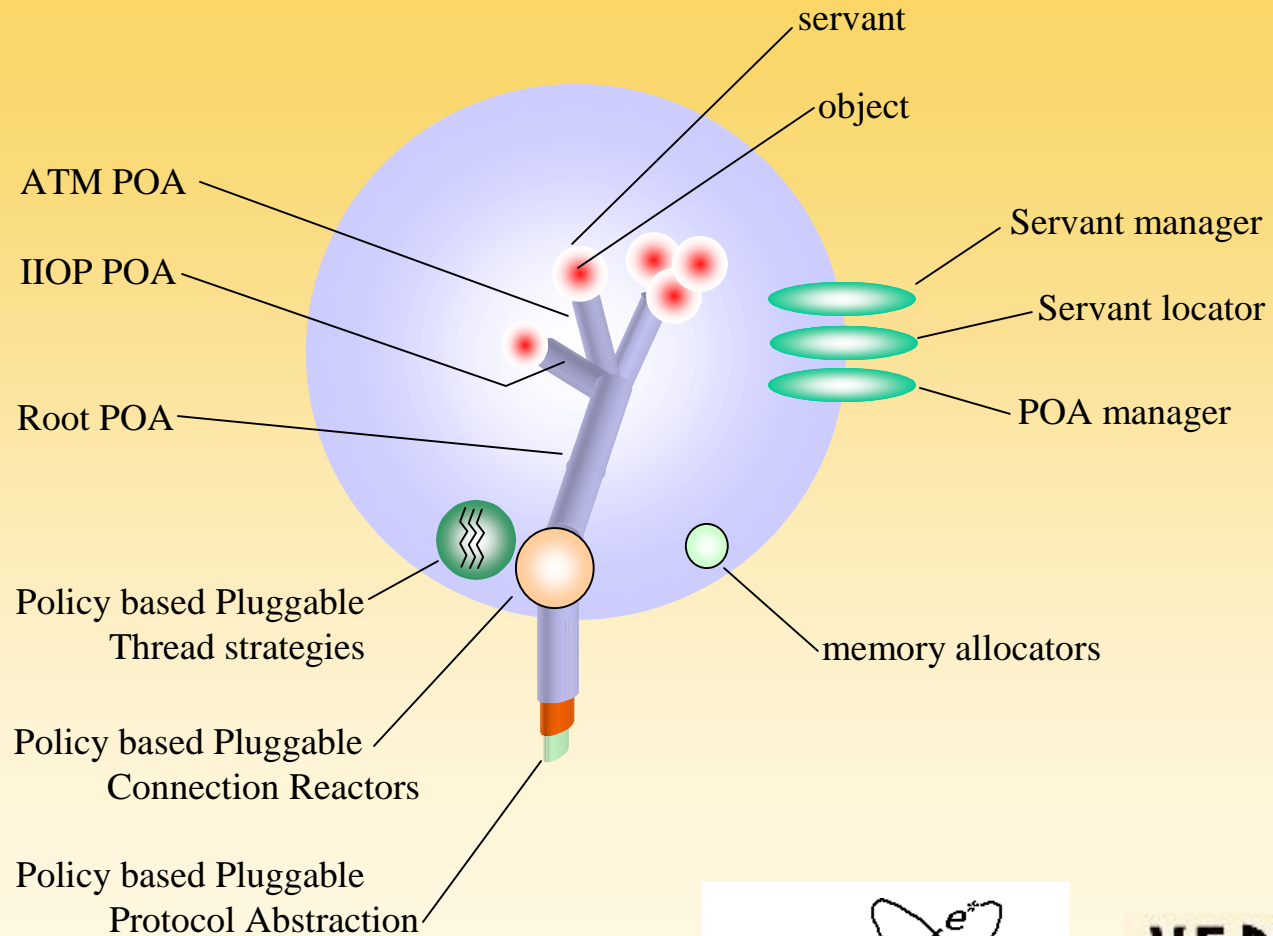
quantum ^{e+}CORBA by VERTEL®

Basic internals that should be open



quantum CORBA by VERTEL®

Basic internals that should be open



What we learned as a consequence

- Solutions customer driven, e.g Sprint, Tellium - want to shorten development cycle
- 3-G ORBs need an open architecture - high/low level middleware
- Internal architecture 'must' be exposed to make extensibility intuitive - reactor, acceptor etc.
- Peripherals 'must' be open source documented through well understood and written about patterns -
- Complex optimizations must be exposed cleanly for extensibility and reuse
- Extensibility - 'pluggable'



Impact on core design

- Performance vs. Footprint tradeoff -
- Many features have to be insertable/pluggable
- POA, IFR, IIR, interceptors - whatever
- POA's come in many guises -
- architecture must support end users GIOP/IIOP, ESIOPs extensions easily
- Open accesible architecture - source level
- Stub generation - custom (exotic platforms)
- Roll your own protocols - proper understandable abstractions - get out of TCP thinking box

Impact on ORB architecture



Evolved ORB based on empirical benchmarks

Extensively - measure latencies (o/w, t/w) and priority inversion in core - 'flood-gate', and 'thread storm' tests

Extensive automated white-black box tests

Isolate ORB and transport overheads - no silver bullets

Manage and guarantee memory determinism - no copying

Necessary evil - custom memory allocations are harder to track and debug and cost 'time'

AVOID at all costs complex concurrent designs in core

POA-BOA duality

quantum  *CORBA* by **VERTEL**®

Impact on ORB architecture

Custom protocols will have significant impact on performance - Wash-U studies

Clean separation between protocol, transport and connection abstractions is vital

Minimization of dynamic connections and memory

Mux-demux should be kept in different threads based on priorities

Possibility of assymmetric connections

Expose control to 'request pending back pressure'

Minimize code size repeatedly by iterative rebuilds and - test retest

Impact on ORB architecture



Continuously optimize stub code

Aim to integrate ORB-OS-I/O as much as possible yet try to maintain a decoupled architecture

Provide a clear understanding of peripheral but necessary services - naming, trader, notification, txns, logging

Evolve ORB in a service ready mode - portable interceptors

Extensible transports, or pluggable transports ? Mcast, udp - need clean ways to incorporate these

Possibilities for ORB to support reduced forms of GIOP

ORB must insulate the end users from Compiler challenges

gnu, Greenhills, *quantum* CORBA by VERTEL.

Impact in other areas

- Apart from ORB internals having to be tailored to embedded habitat there's one more essential element
- CORBA or CORBA-like minimal embeddable services, - self contained *in the box* -
- Interoperable naming, Notification/event, trader, logging and others e.g. scheduling
- Newer faster backplane and protocol technologies pose a new set of issues -
- Fast responsive support

Problems

- Still lack of real QoS-enabled, enforceable middleware
- lack of units of functionality - component model like solution
- Unsolved mysteries - Bootstrap & lifecycle
- There is still relatively small uptake of CORBA in the embedded community by enlarge - old myths about CORBA still around
- For those that have adopted embedded CORBA they sit between 2 specs.
- What effect will Infiniband and Rapid IO have ?

Concluding remarks

- This is still an evolving space
- CORBA is beginning to be used to Command and Control
- The challenge is to maintain performance to footprint ratio
- There needs evolve an embedded CORBA spec
- There need to be minimal-embeddable real-time capable CORBA services
- QoS based components should be the next *quantum step* ;-)

quantum  CORBA by VERTEL®



Thank you

quantum  *CORBA* by **VERTEL**®