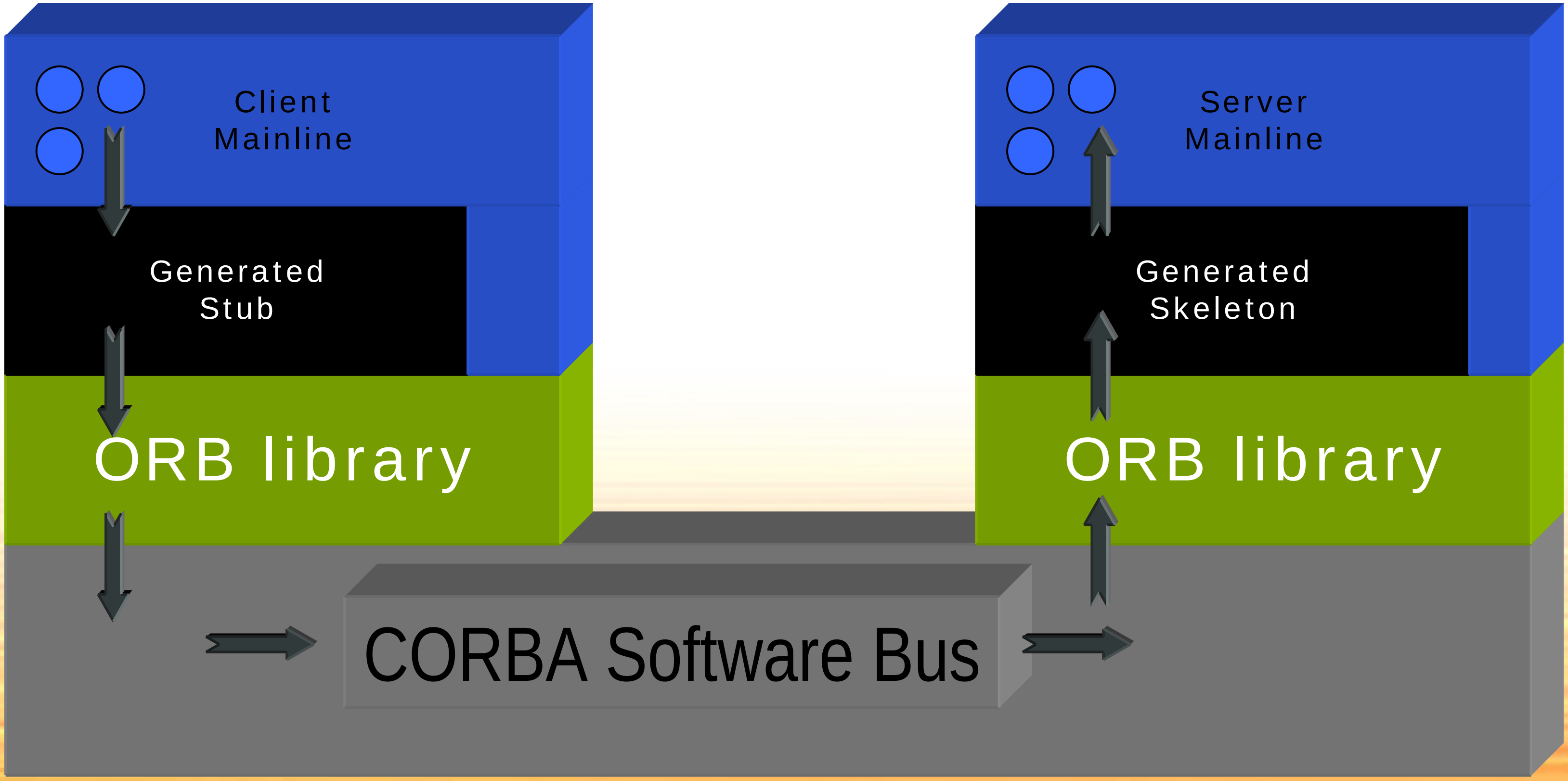
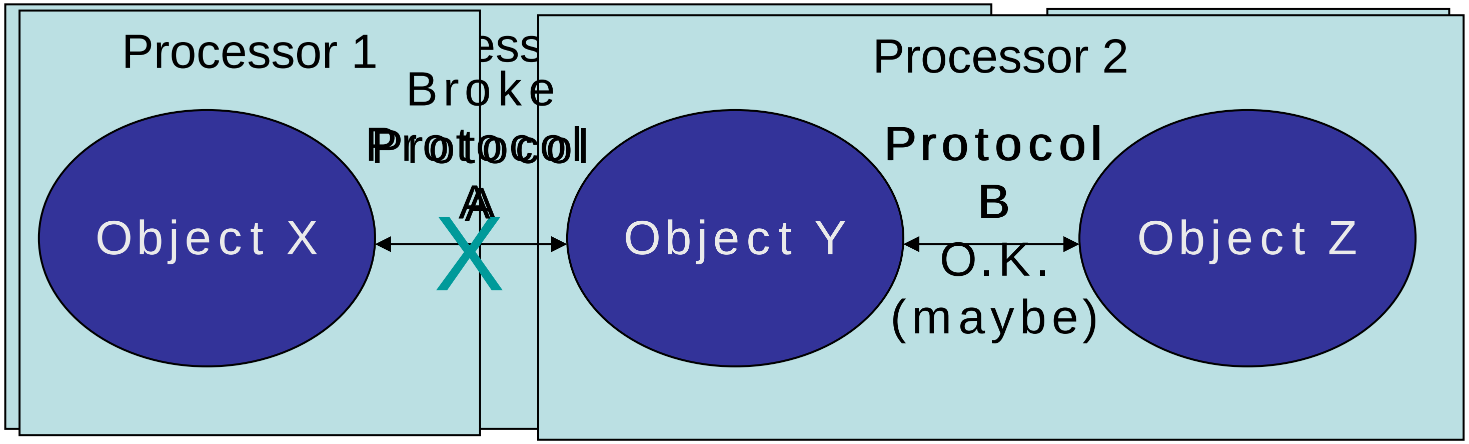
CORBA Across Embedded Devices

Victor Giddings

Objective Interface Systems

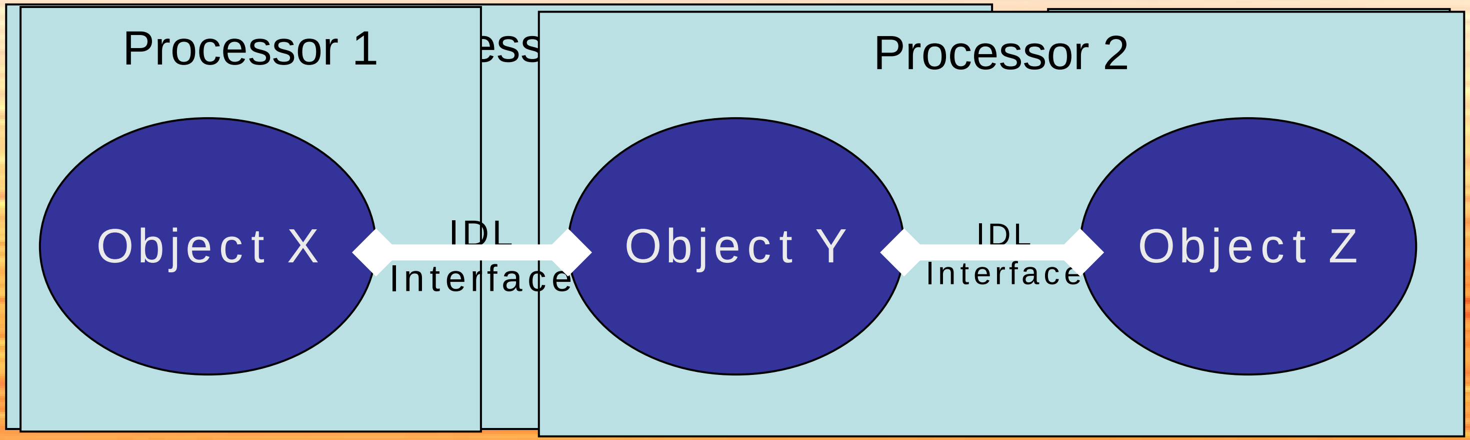

Client Mainline
Generated Stub
ORB library
Server Mainline
Generated Skeleton
ORB library
CORBA Software Bus

Without Location Transparency



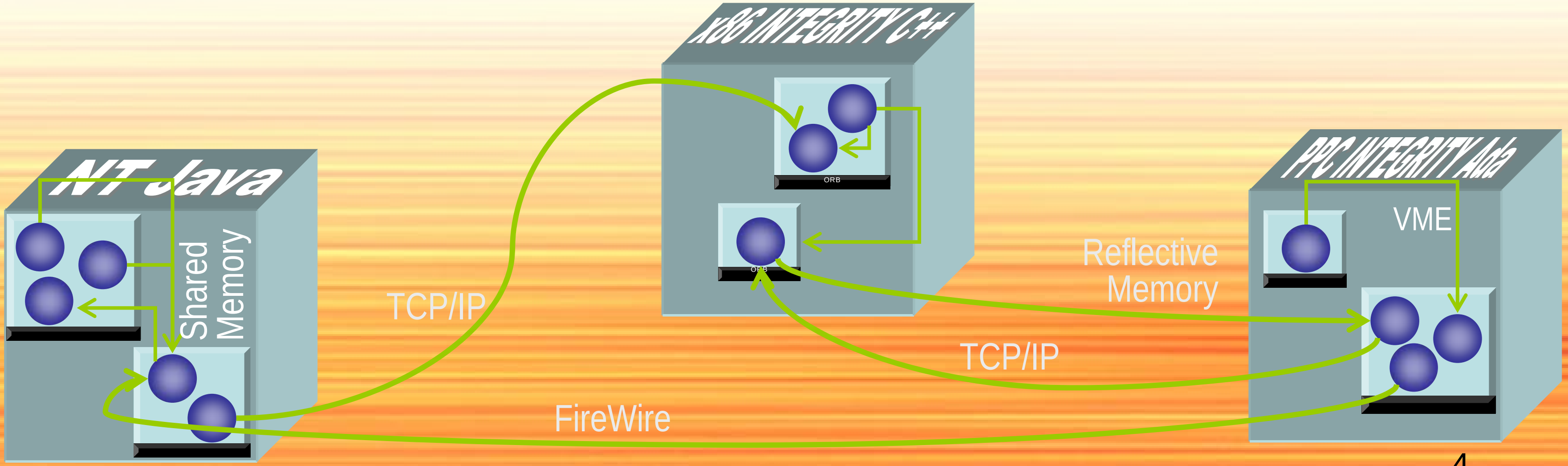
Oops …

With Location Transparency



CORBA Across Embedded Devices

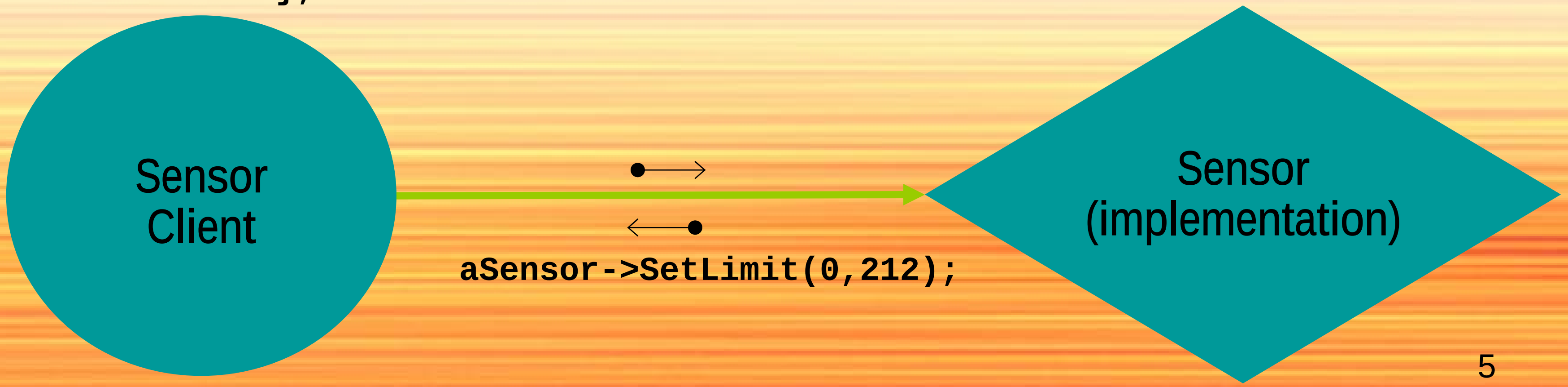
- *Transparent* Distributed Communication
  - Location Transparency
    - Distributed application written the same as non-distributed
    - Deployment Flexibility
  - Transport/Media Transparency
- Heterogeneous Communication
  - Processor,
  - RTOS,
  - Language



Interface Definition Language (IDL) is the Key

- Specifies interfaces to remote objects
- Part of the CORBA specification
- Fully Object-Oriented
- Mapped to modern programming languages – Ada 95, C, C++, Java, Smalltalk, OO Cobol ...

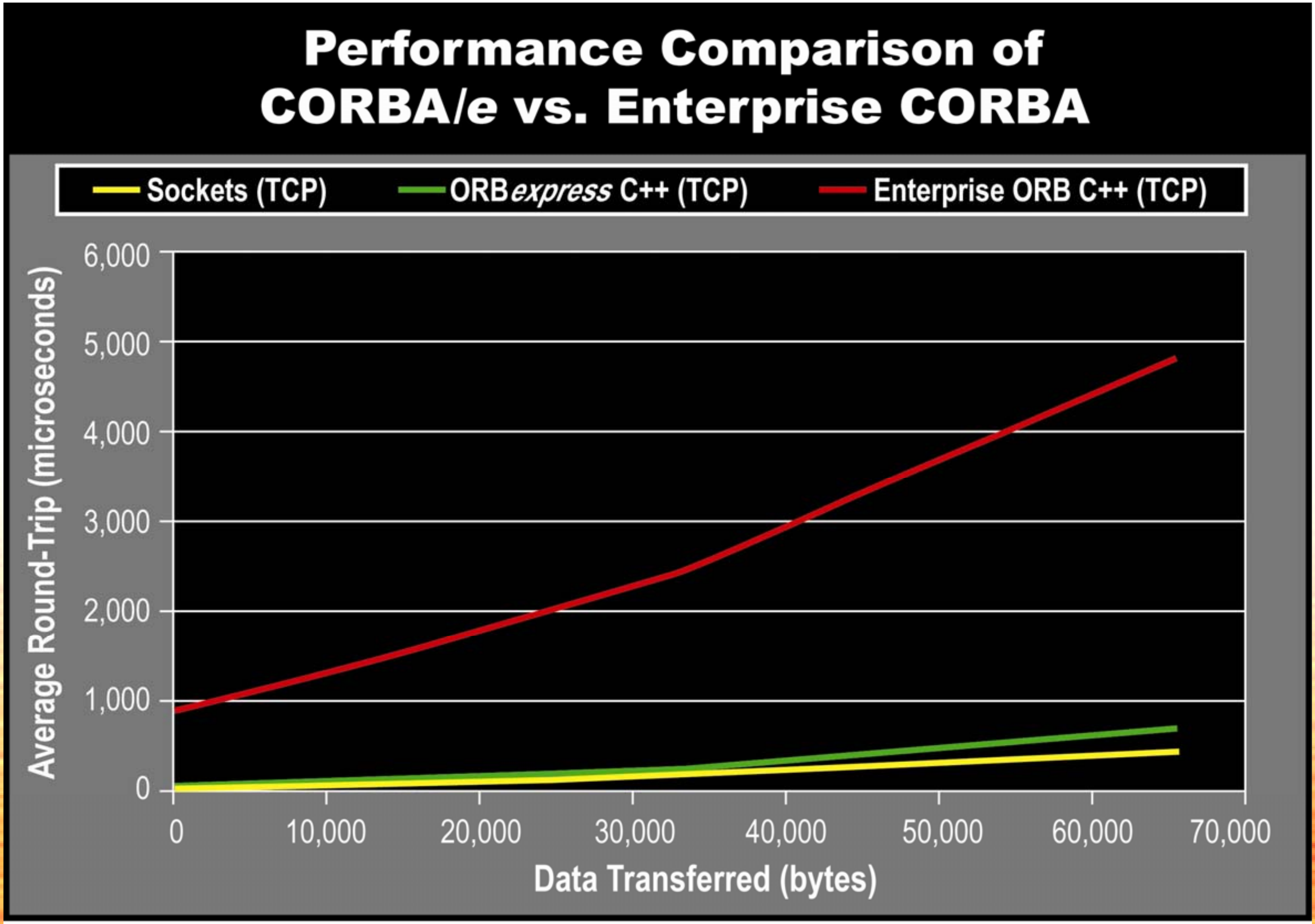
```
interface Sensor {
   exception InvalidLimit{};
   void SetLimit (in long Lower_Limit,
                         in long Upper_Limit)
               raises (InvalidLimit);
   boolean Valid ();  //true if within limits
   readonly attribute long CurrentValue;
};
```



```
aSensor->SetLimit(0,212);
```

Why CORBA?

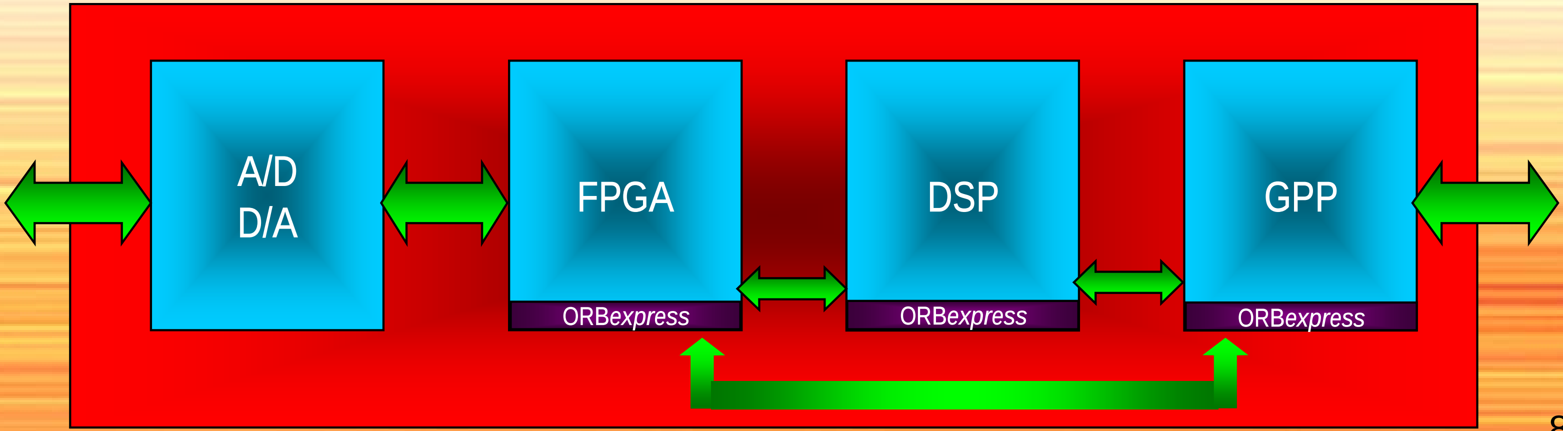
- CORBA removes the need for
  - Low-level Bit Programming
  - Defining a client/server protocol
  - Defining a message format
  - Demarshalling data upon return from call
  - Marshalling of data
- Preserve investment in software artifacts while changing
  - Processor
  - Operating system
  - Programming language



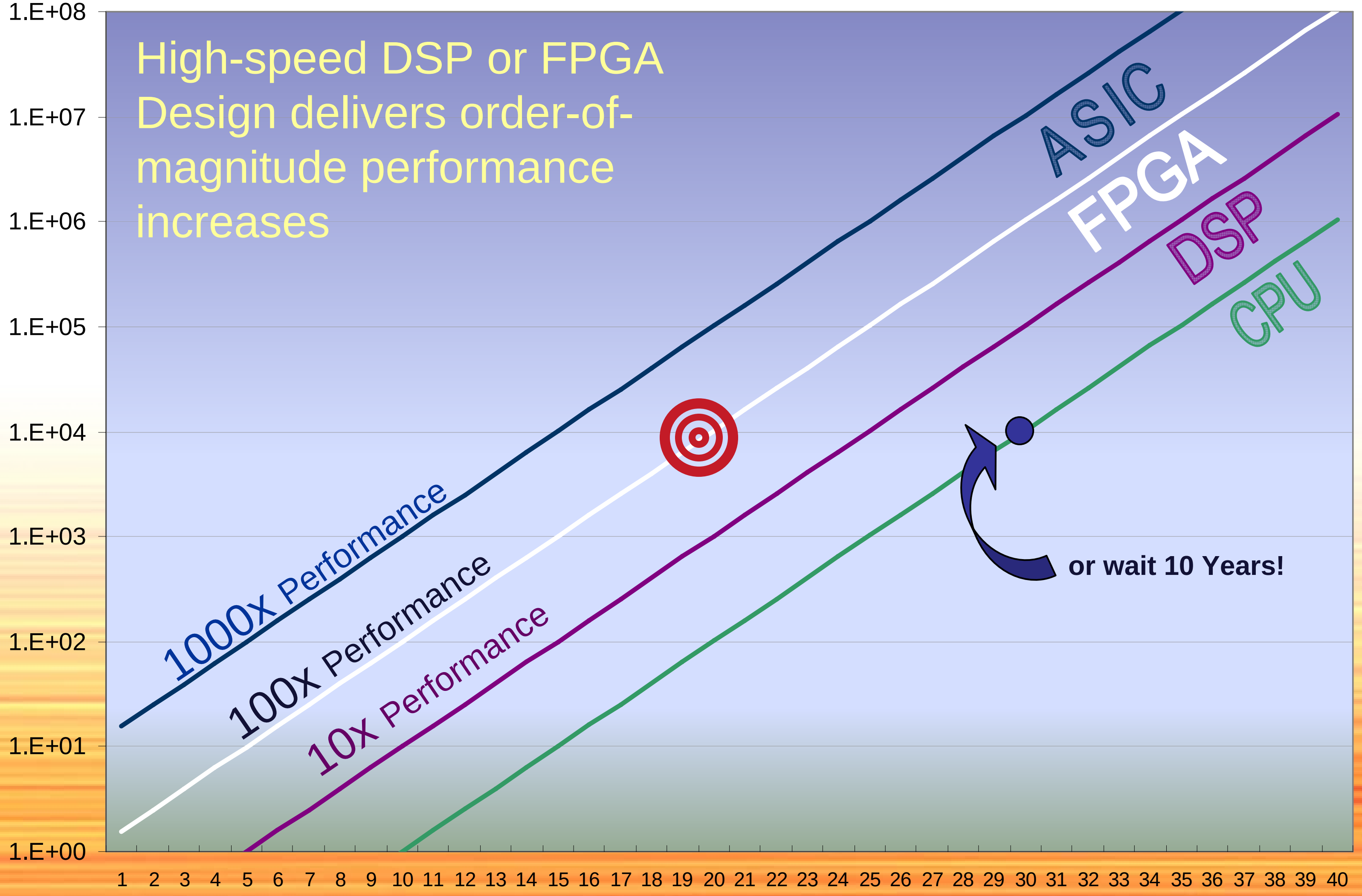
Source: Lockheed Martin's Advanced Technology Laboratories

Why CORBA for embedded?

- Location Transparency
  - Architectural flexibility because of no-cost relocation
- High-performance, low-footprint implementations
- Interoperability with server and enterprise systems
- Wide Availability
  - Microprocessors, RTOSes, transports
  - ➢ DSPs
  - ➢ FPGA



Why Use DSPs or FPGAs?



High-speed DSP or FPGA Design delivers order-of-magnitude performance increases

or wait 10 Years!

CORBA on DSPs Characteristics

- Challenges
  - Operating environment – Compilers and BIOS
  - Footprint optimization
- Implementation tactics
  - Operating environment not as challenging as expected
    - E.g., TI BIOS is as capable as many RTOSes – threads, synchronization, etc.
  - Decomposable features – footprint cost limited to features used
  - Avoid language library bloat, e.g.,
    - C++ STL
    - C++ native exception processing

CORBA on DSPs Characteristics

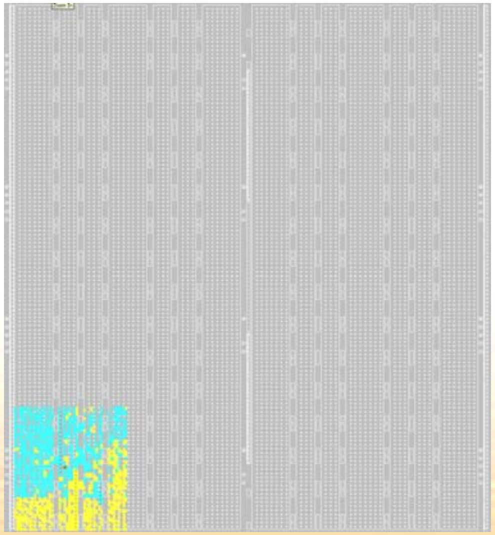
- Result – multiple implementations reporting
  - Small performance overhead
    - Single digit percentage
    - 100x speed up in throughput
  - Less than 100K footprint

CORBA on DSPs
Benefits for DSP Developers

- DSP developer uses own existing development tools
  - BIOS
  - Compiler
- Performance benefit with minimal overhead
- Easy transition of applications from GPP code to DSP code
  - Can be re-compile
  - Optimize to use native library
- Changing input and output sources does not require changes to DSP code

CORBA on FPGA Characteristics

- Challenges
  - What does a firmware ORB do?
  - Stream-oriented data optimization
  - Minimizing gate count
  - Maintaining throughput
- Results
  - Support dynamic partial reconfiguration of FPGA code
  - Less than 2% of FPGA real-estate for ORB functions
    - Support customization to eliminate unused functionality
  - Process messages at up to 1GByte per second line rate
    - 32-bit data path at 250MHz
  - Support dynamic computational chains



FPGA Designer Benefits

- FPGA designer uses own existing development tools for FPGA simulation and netlist generation
- Performance benefit with minimal overhead
- Easy conceptual transition of applications from GPP code to FPGA code
- Changing input and output sources does not require changes to FPGA code

Why CORBA for embedded? (reprise)

- Location Transparency
  - Architectural flexibility because of no-cost relocation
- High-performance, low-footprint implementations
- Interoperability with server and enterprise systems
- Wide Availability
  - Microprocessors, RTOSes, transports
  - ➢ DSPs
  - ➢ FPGA

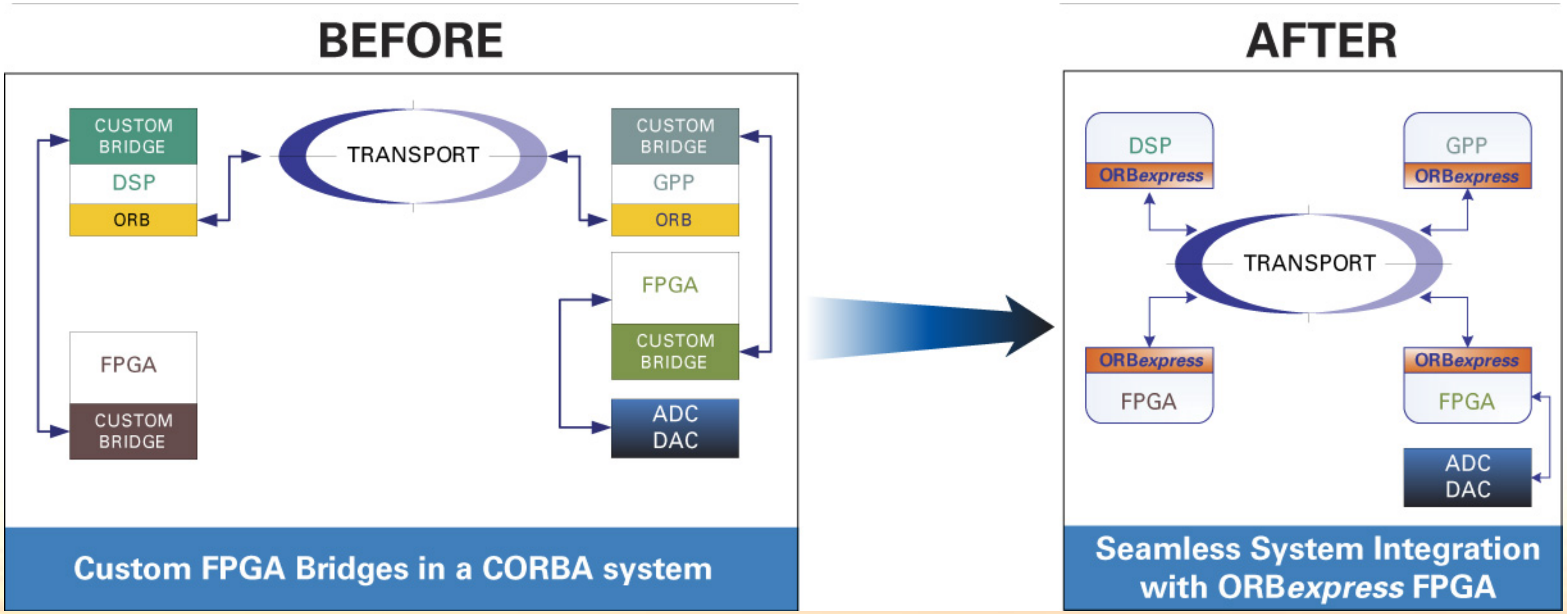
- Location transparency

  +

  Choice of
  - General Purpose Processor
  - Digital Signal Processor
  - Field Programmable Gate Array

  = **Technology Transparency**

Technology Transparency



Custom FPGA Bridges in a CORBA system

Seamless System Integration with ORB*express* FPGA

Technology Transparency System Architect Benefits

- System architect can migrate functionality among GPP, DSP and FPGA as needed
  - Assignment of functionality does not need to be decided early
  - Functionality can be tested on GPP, then migrated to FPGA as needed, without changing the main application
- System architect can focus on functionality while ORBexpress provides the necessary interface details for the (GPP, DSP, FPGA) code developer
- Full technology transparency feasible in a heterogeneous system architecture
- “Future proof”
  - To technology change
  - Operating environment migration