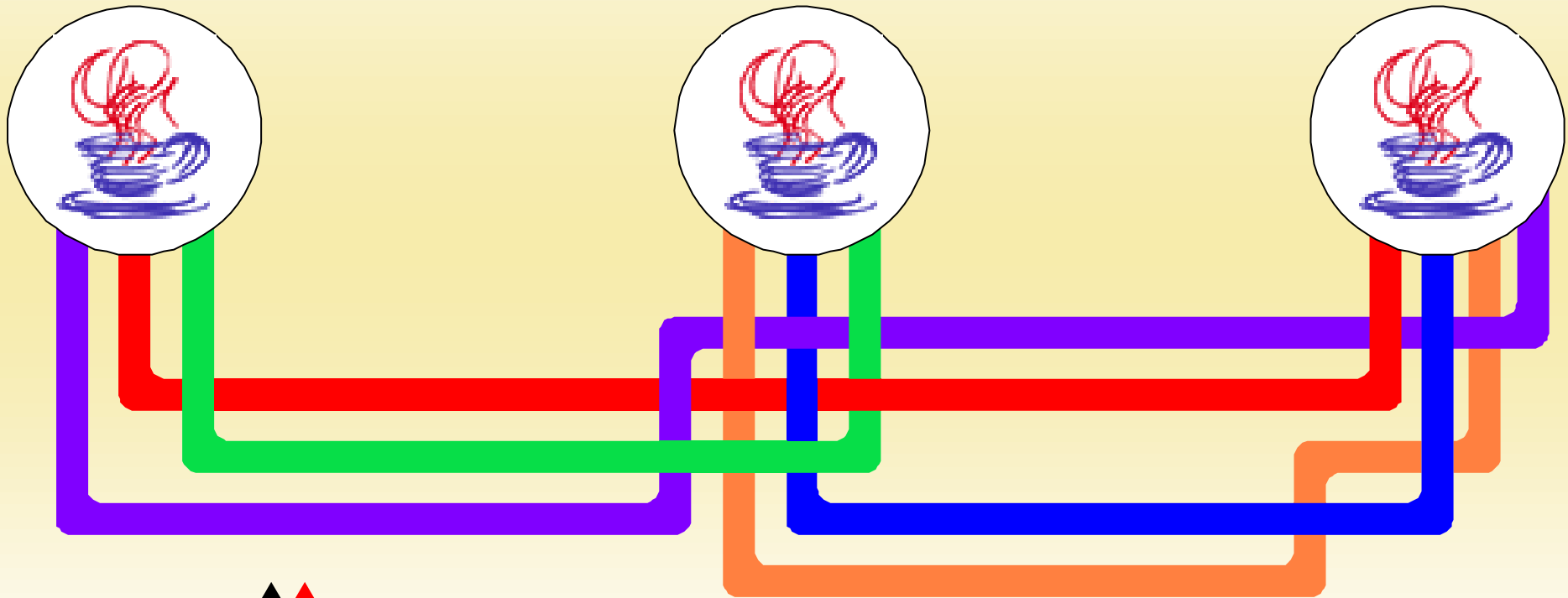


FlexiNet Open ORB Framework



14 Oct 97

Richard Hayton

APM.2077

© 1997 ANSA Consortium

Overview

- Core FlexiNet framework
 - future work will be built on this
- Provides transparent binding
 - For local or remote interconnection
- Open binding architecture
 - To allow general reflection or communication
- Minimalist API
 - External control via reflection



Topics

- Generic Communications and Reflection
- Naming and Binding
- Resource Management
- API
- Performance & Summary



Topics

- **Generic Communications and Reflection**
- Naming and Binding
- Resource Management
- API
- Performance & Summary

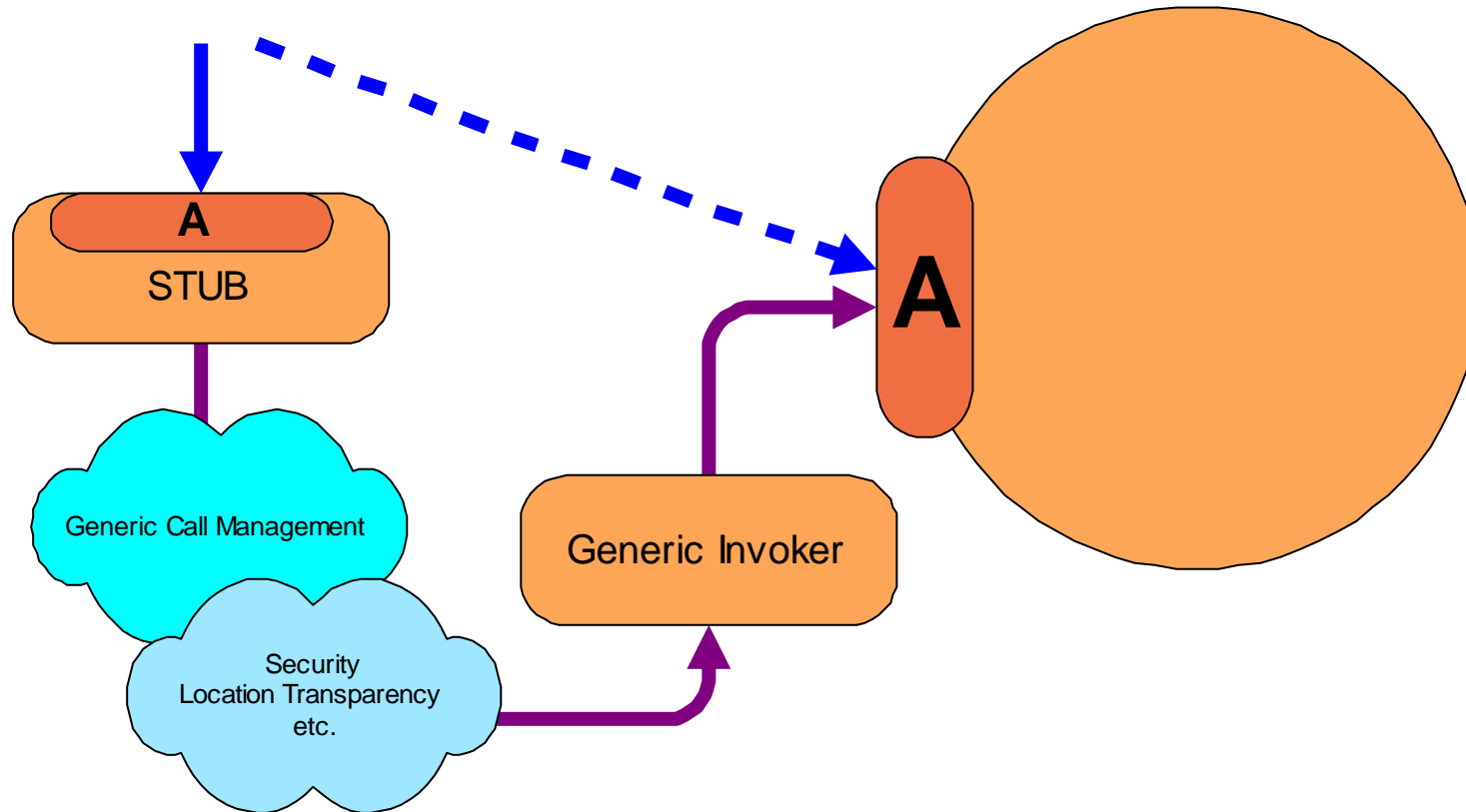


Generic Communications

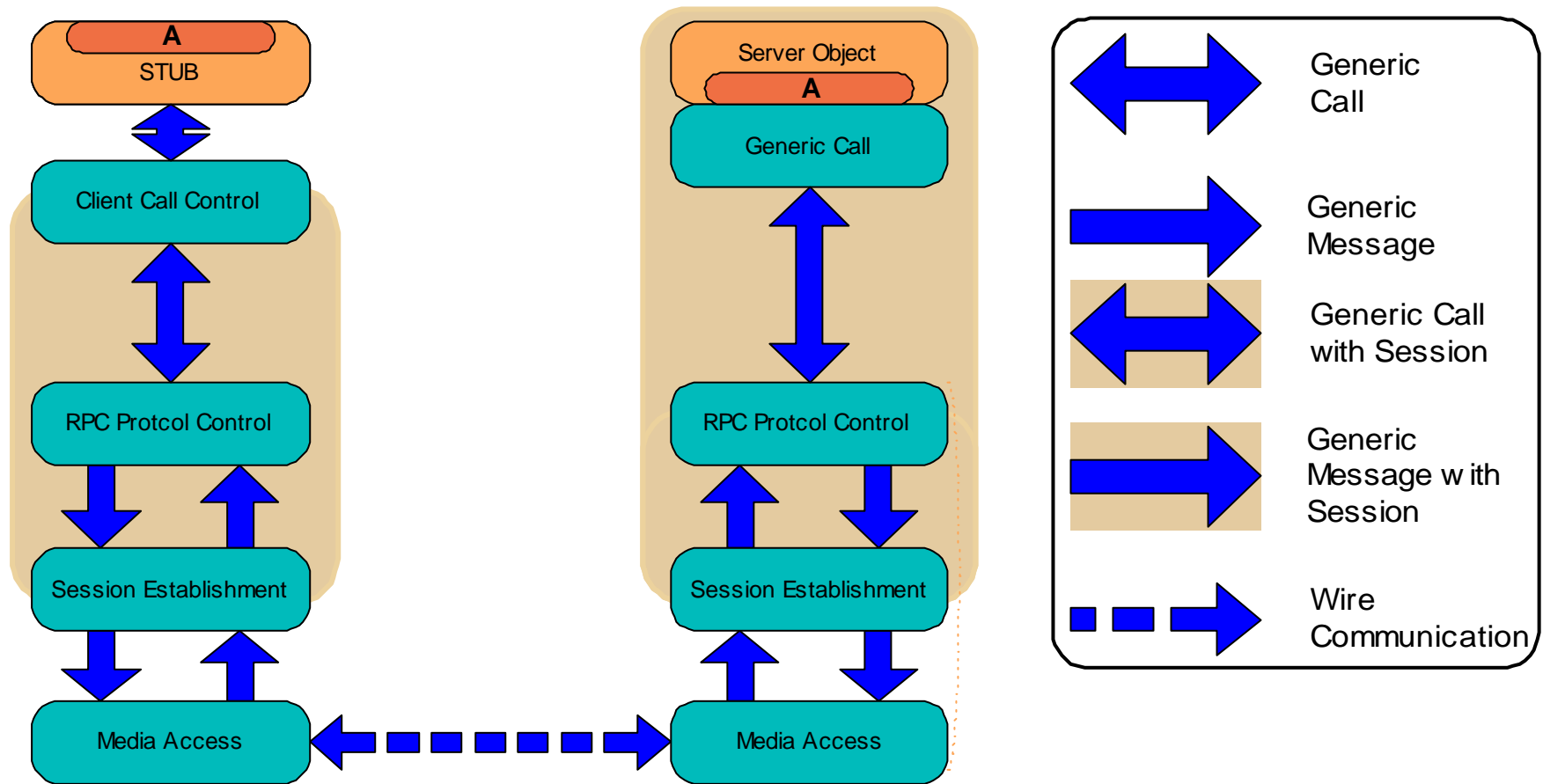
- Transparent but Powerful
 - No special compiler - stub generated on the fly
- Utilise Java Core Reflection
 - Standard representation of a method call
 - Use Java generic method invocation
- Flexible Reflection
 - Can apply any transformations or restrictions on call
 - Stub is not dependant on interconnect mechanism
- Evolvable: engineering can change under the feet of API



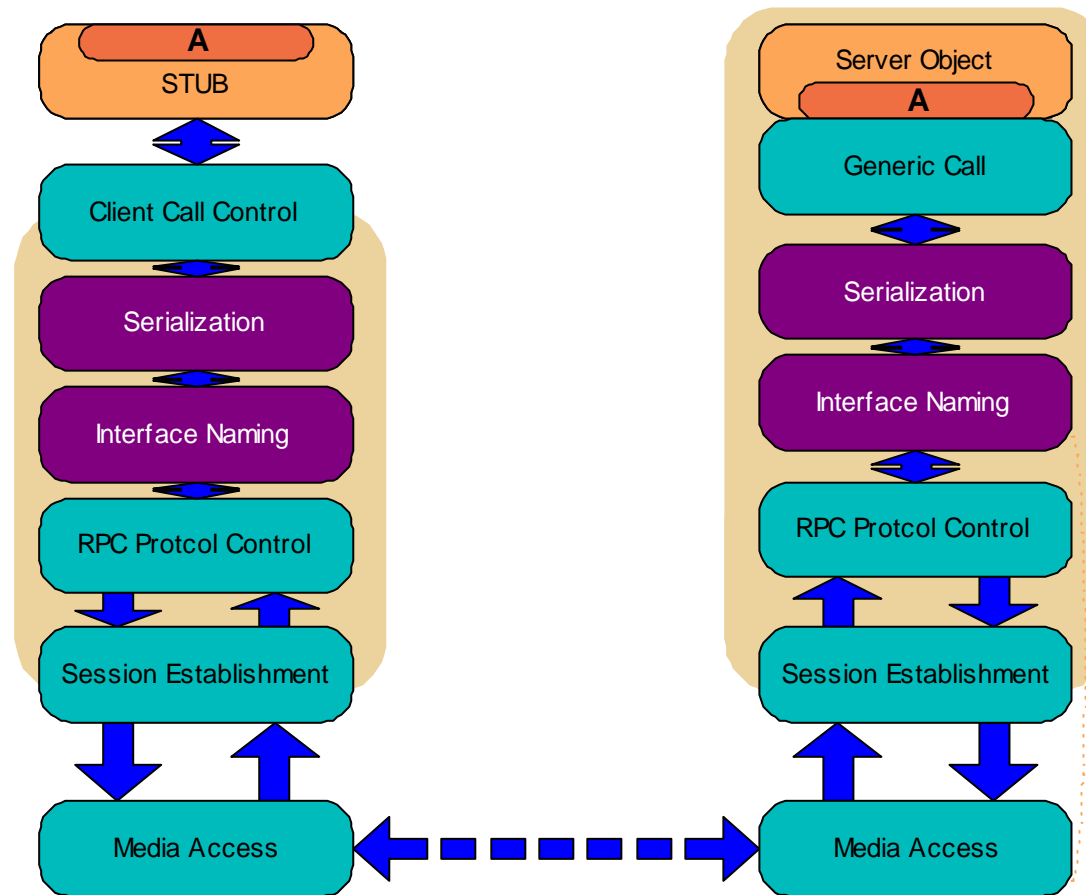
Generic Communication



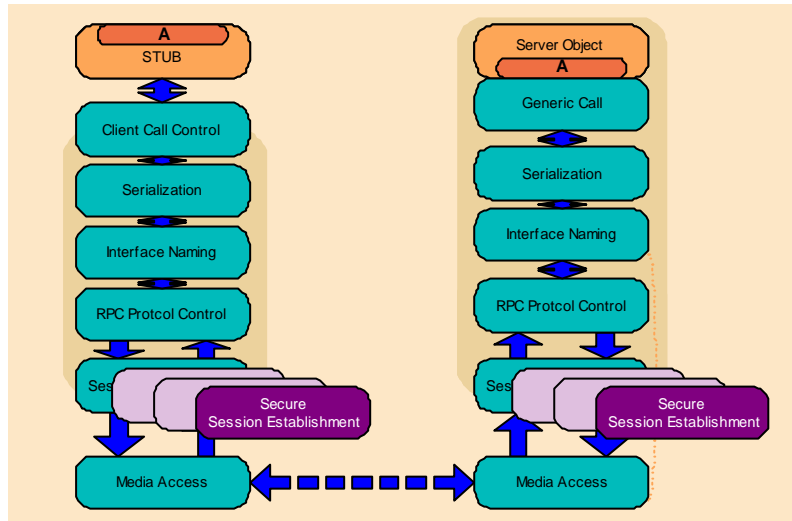
Generic Communications Stack



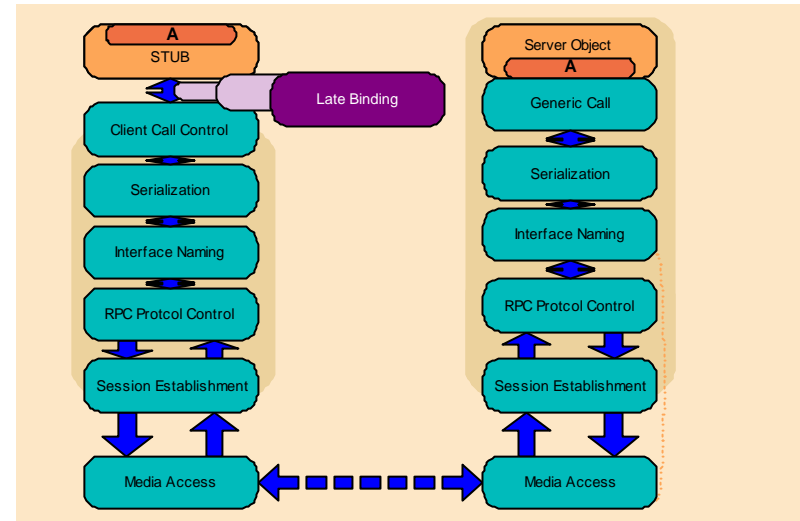
Simple Remote Invocation Stack



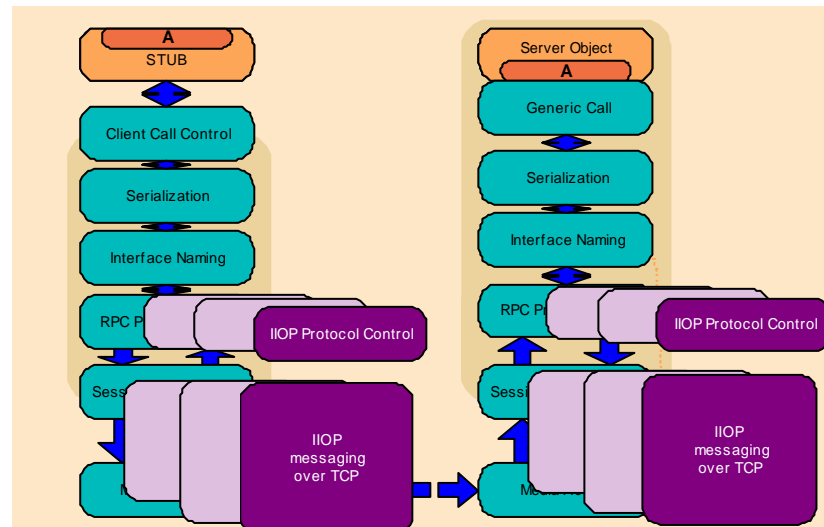
Other Remote Invocation Stacks



SECURE SESSIONS



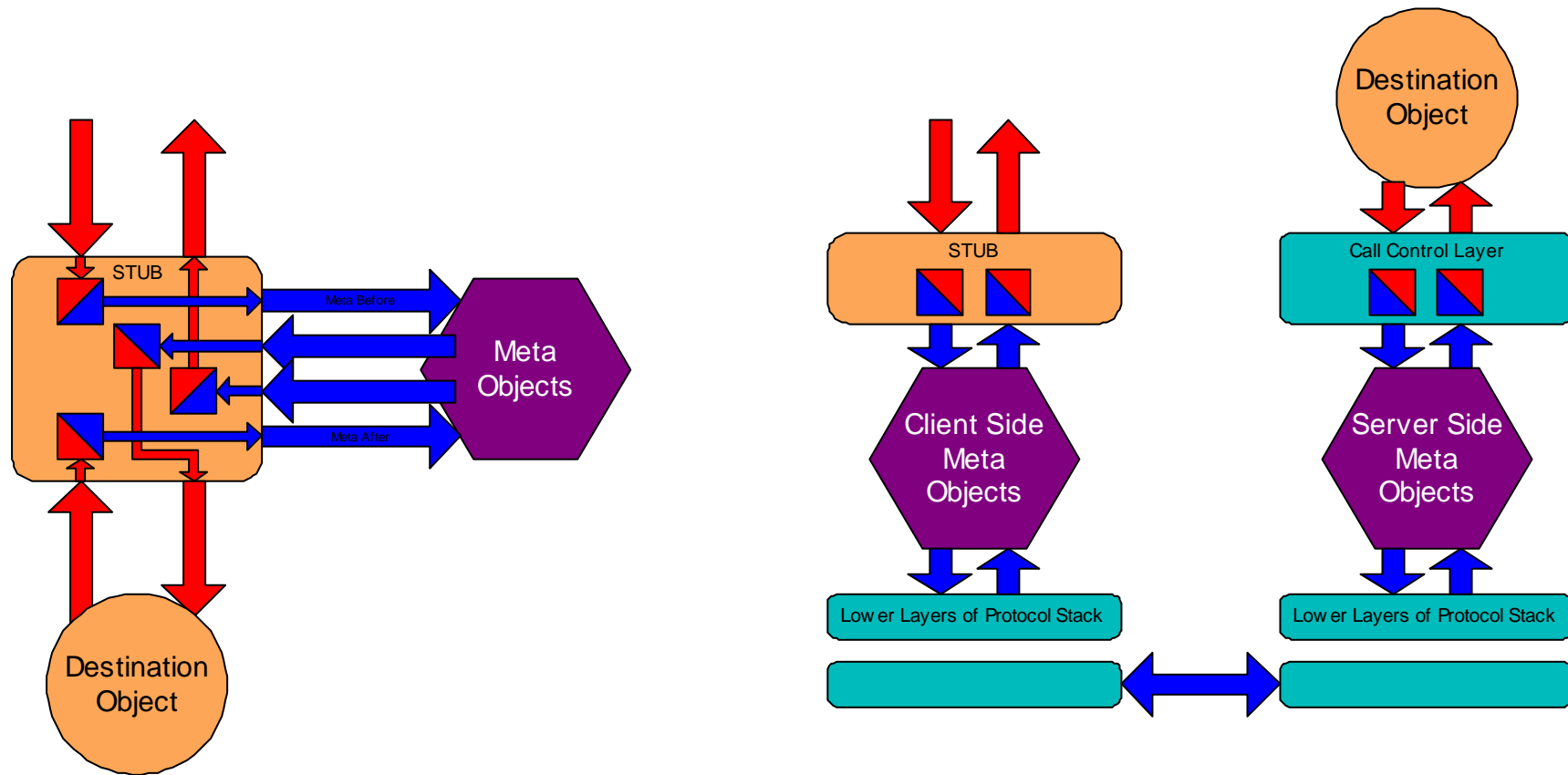
LATE BINDING



*IIOP
TRANSPORT*



FlexiNet Reflection

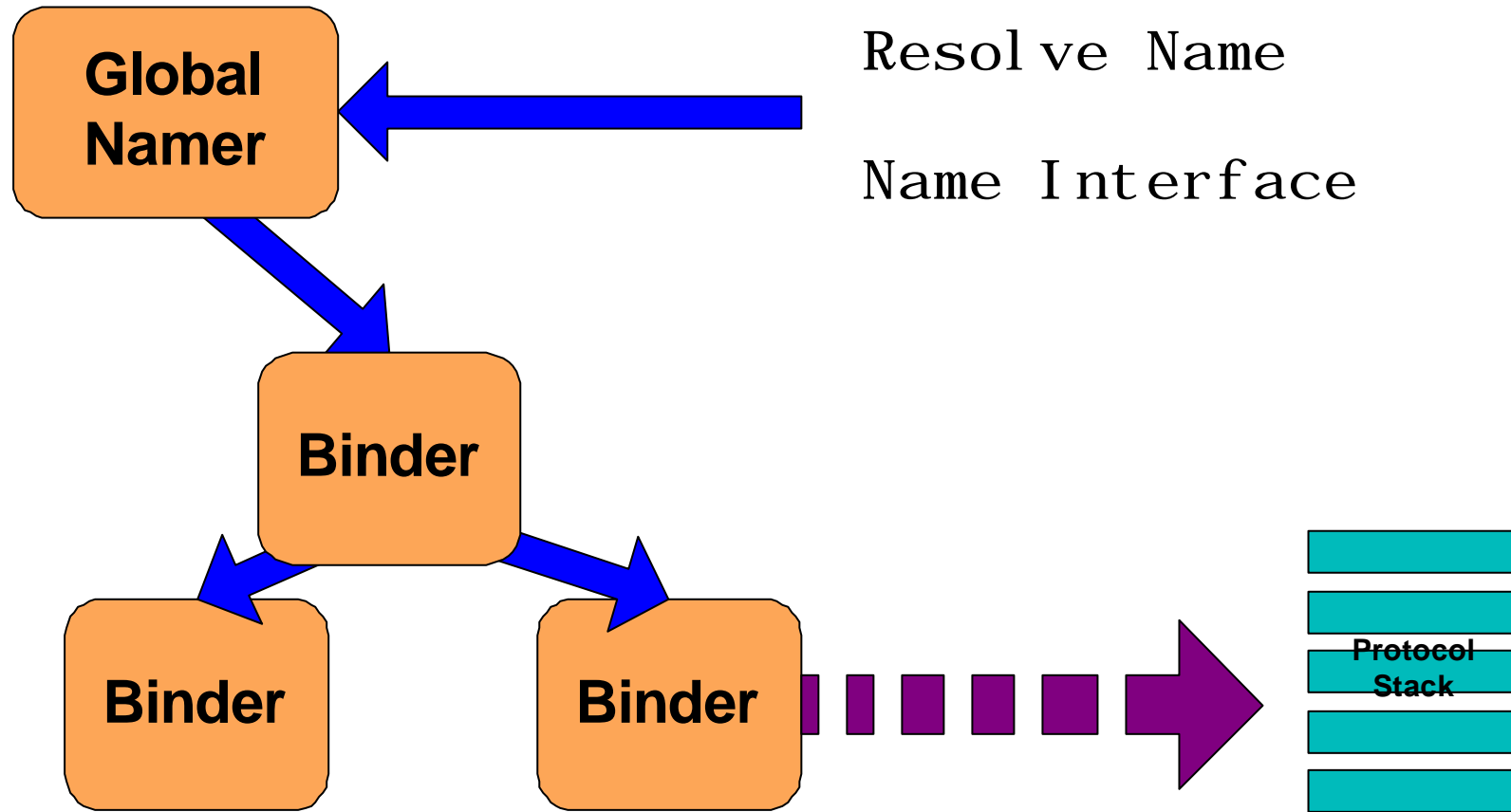


Topics

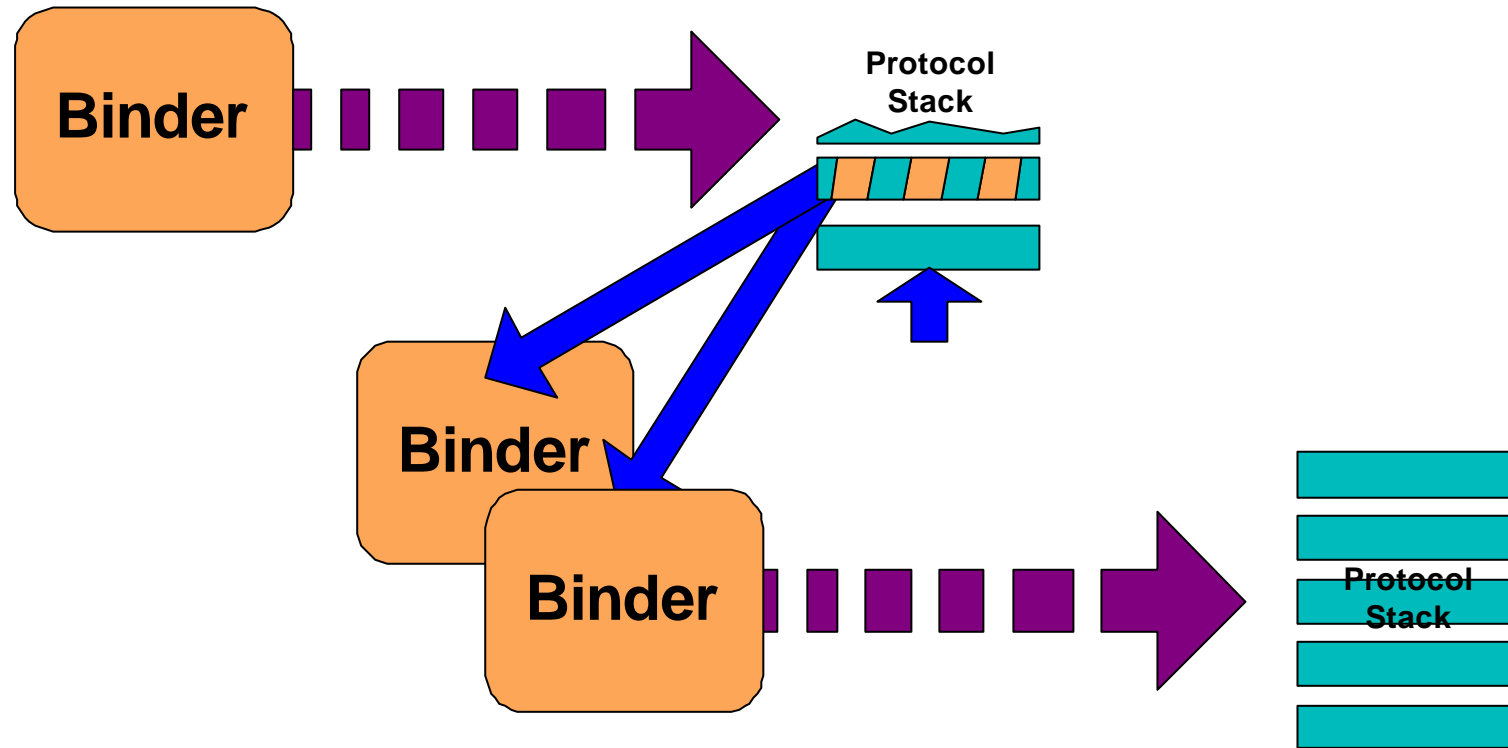
- Generic Communications and Reflection
- **Naming and Binding**
- Resource Management
- API
- Performance & Summary



Naming and Binding



Late / Negotiated Binding



Ongoing Naming/Binding Work

- Negotiation for choice of Binder
 - negotiation based on declarative specifications
- Parametric Binder
 - slot in transparencies
- Integration with transactions
 - layer to maintain transactional context
- Integration with mobility
 - naming layer for interfaces that move



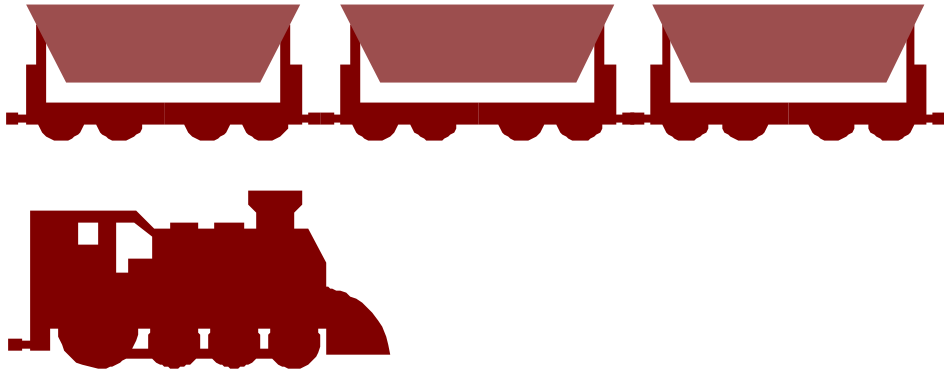
Topics

- Generic Communications and Reflection
- Naming and Binding
- **Resource Management**
- API
- Performance & Summary



Resource Management

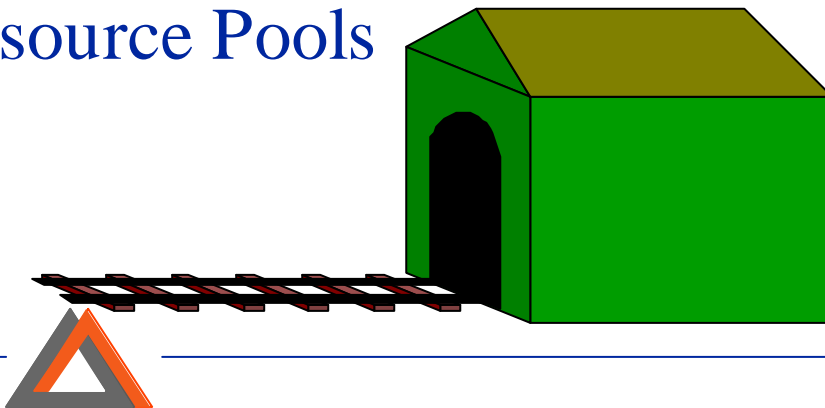
Resources



Management Policies



Resource Pools

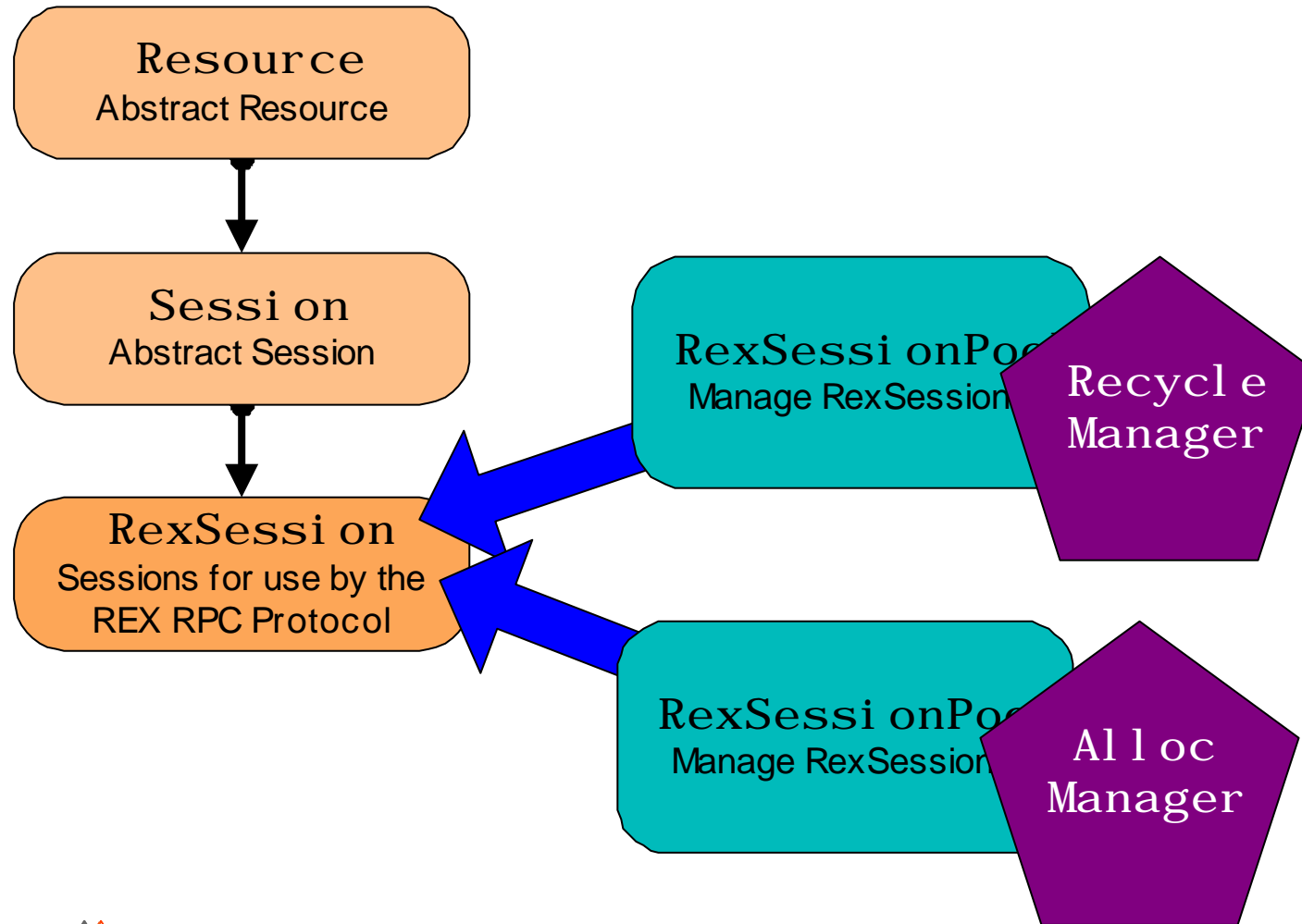


Resource Management Issues

- Abstract resource management
 - one layer creates resource, another layer uses it
 - separation functions of related layers
- Decouple resource and management classes
 - don't redesign management for every class
 - simplify sub-classing of resources
 - enable reuse
- Dynamic Configuration
 - allow run time choice of management strategies



Resources, Pools, Pool Managers

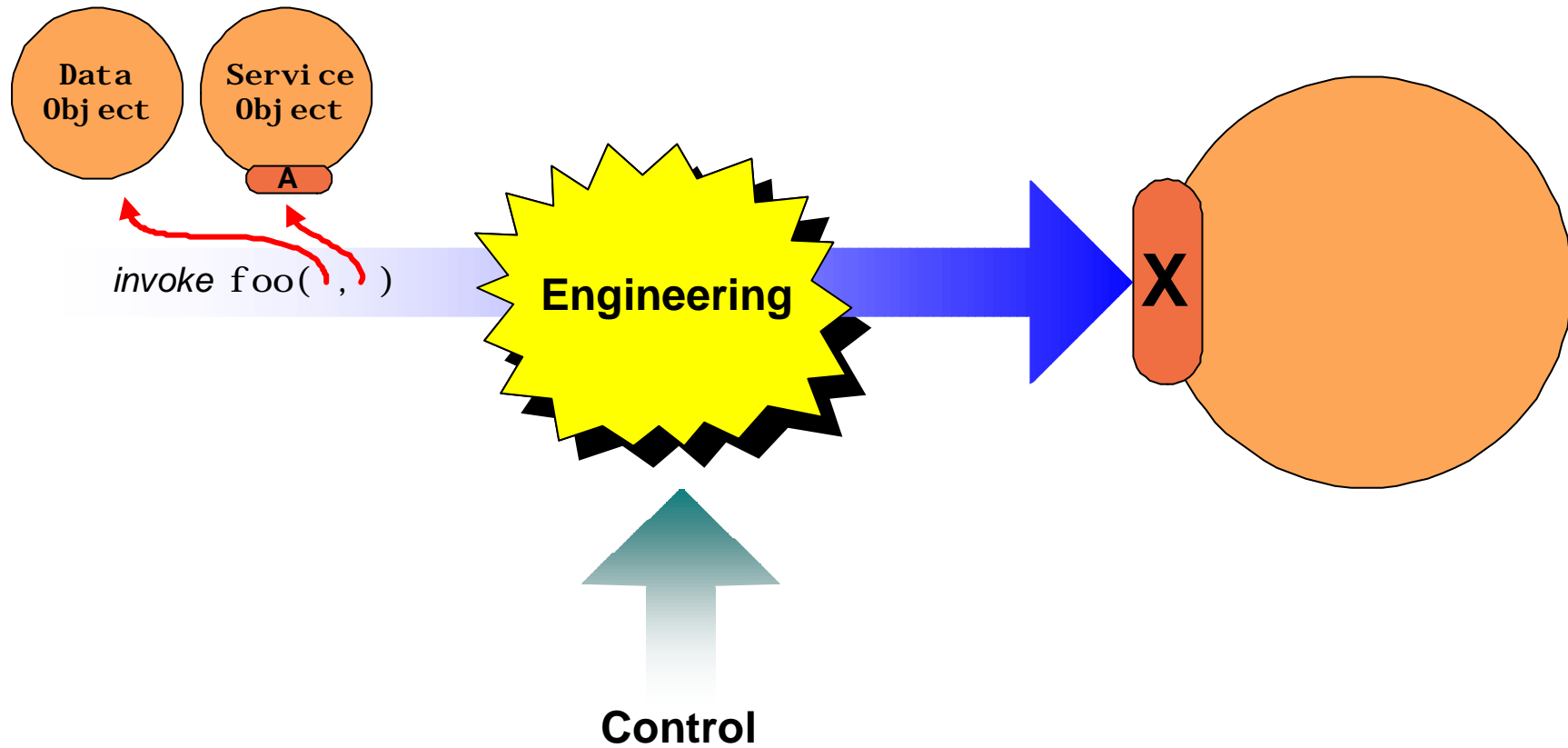


Topics

- Generic Communications and Reflection
- Naming and Binding
- Resource Management
- **API**
- Performance & Summary



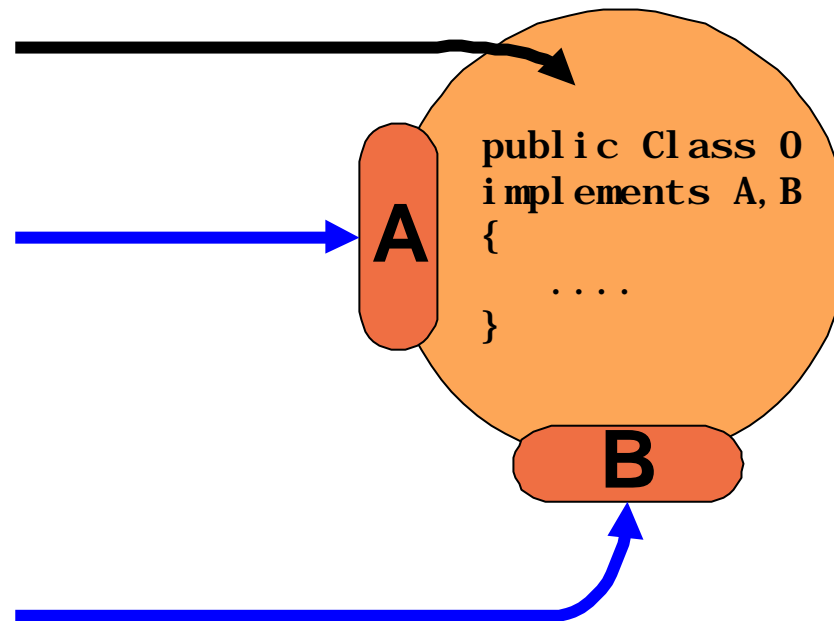
API: Transparency Issues



When to pass by reference?

Pointer to an object, pass the state of the object.

Pointer to an interface, pass a reference to the interface.



*Solution: examine the class of the **reference***

class 0

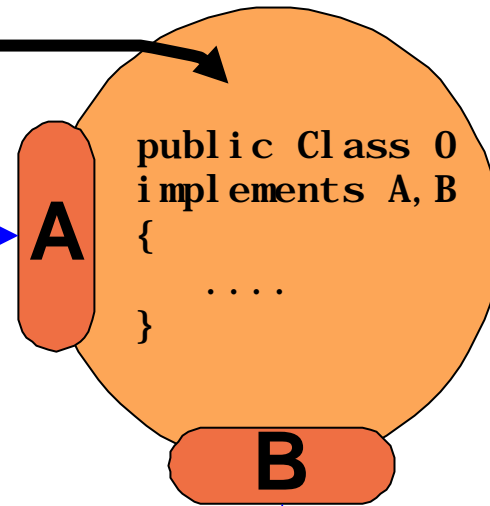
0 foo =

interface A

A foo =

interface B

B foo =

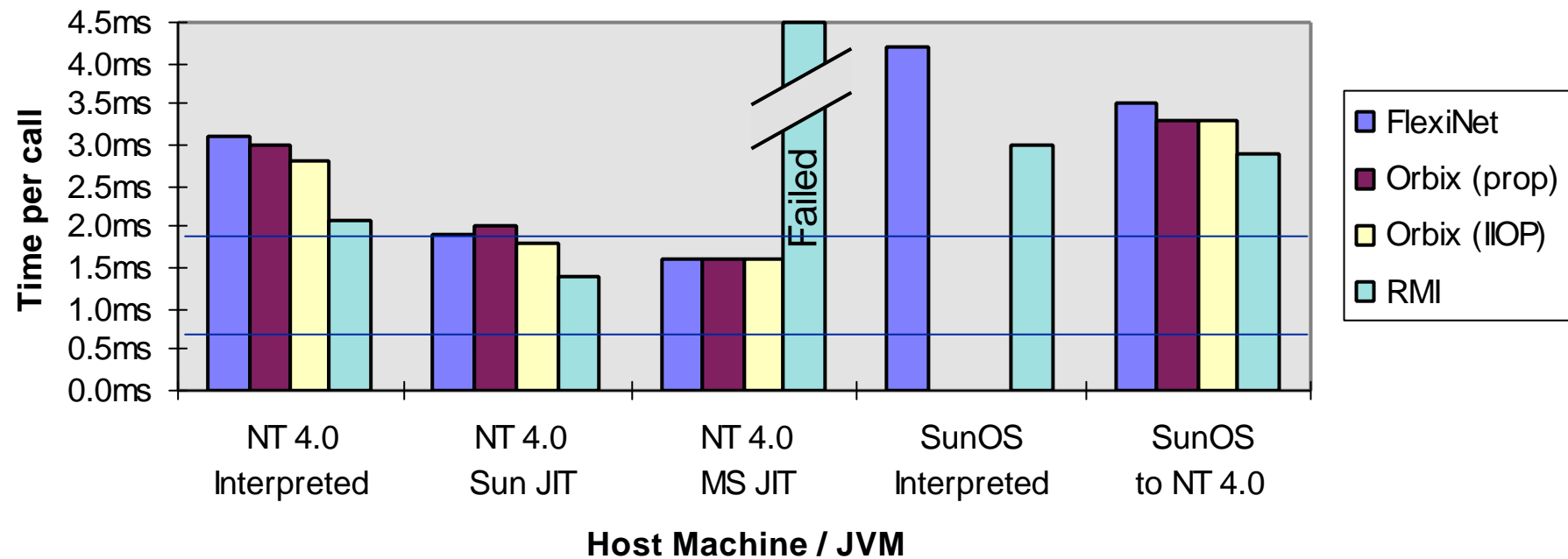


Topics

- Generic Communications and Reflection
- Naming and Binding
- Resource Management
- API
- **Performance & Summary**



Comparative Performance for Null-Operation



Java Lessons Learnt

- Core reflection is really nice
 - But: too much left to JVM implementation
- Pure Java?
 - Suffer from 'next release syndrome'
 - Different interpretations of the spec.
 - But: better than the C++ story!
- Speed
 - Can be made to go fast
 - Slow in unexpected places - e.g. object creation



Summary

- FlexiNet ORB Framework
 - built, working and tested
- Available for use
 - released to sponsors
 - works in applications and applets
- Pure Java
 - some clever (but legal) techniques
- Future research built on this testbench
 - transactions, mobility, security

