

Synchronous & Asynchronous Computations & Communication

From the Synchronous End of the Spectrum

Raymond K. Clark, Ph.D.

rkc@mitre.org

Revised 29 January 2001

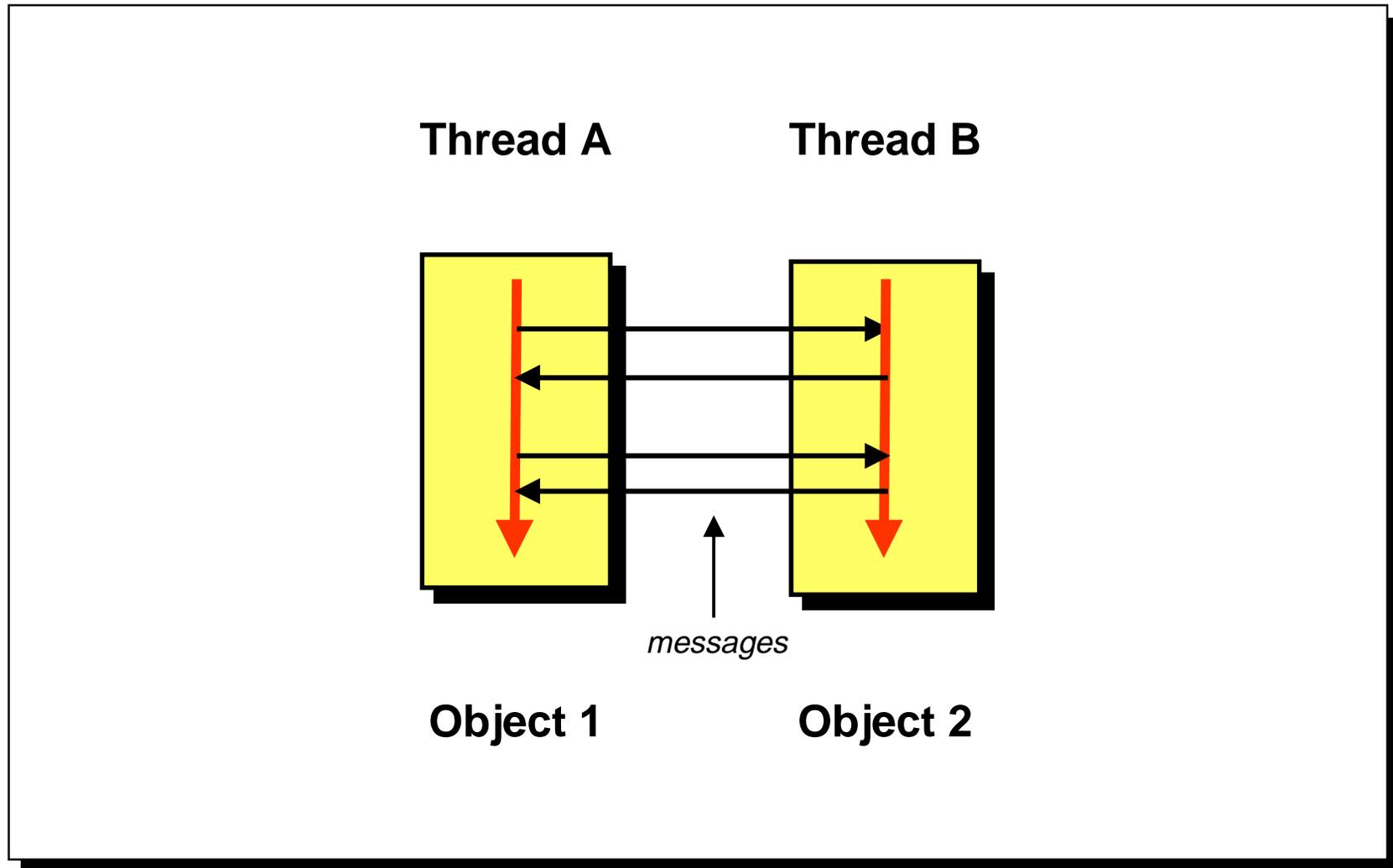
MITRE

Outline

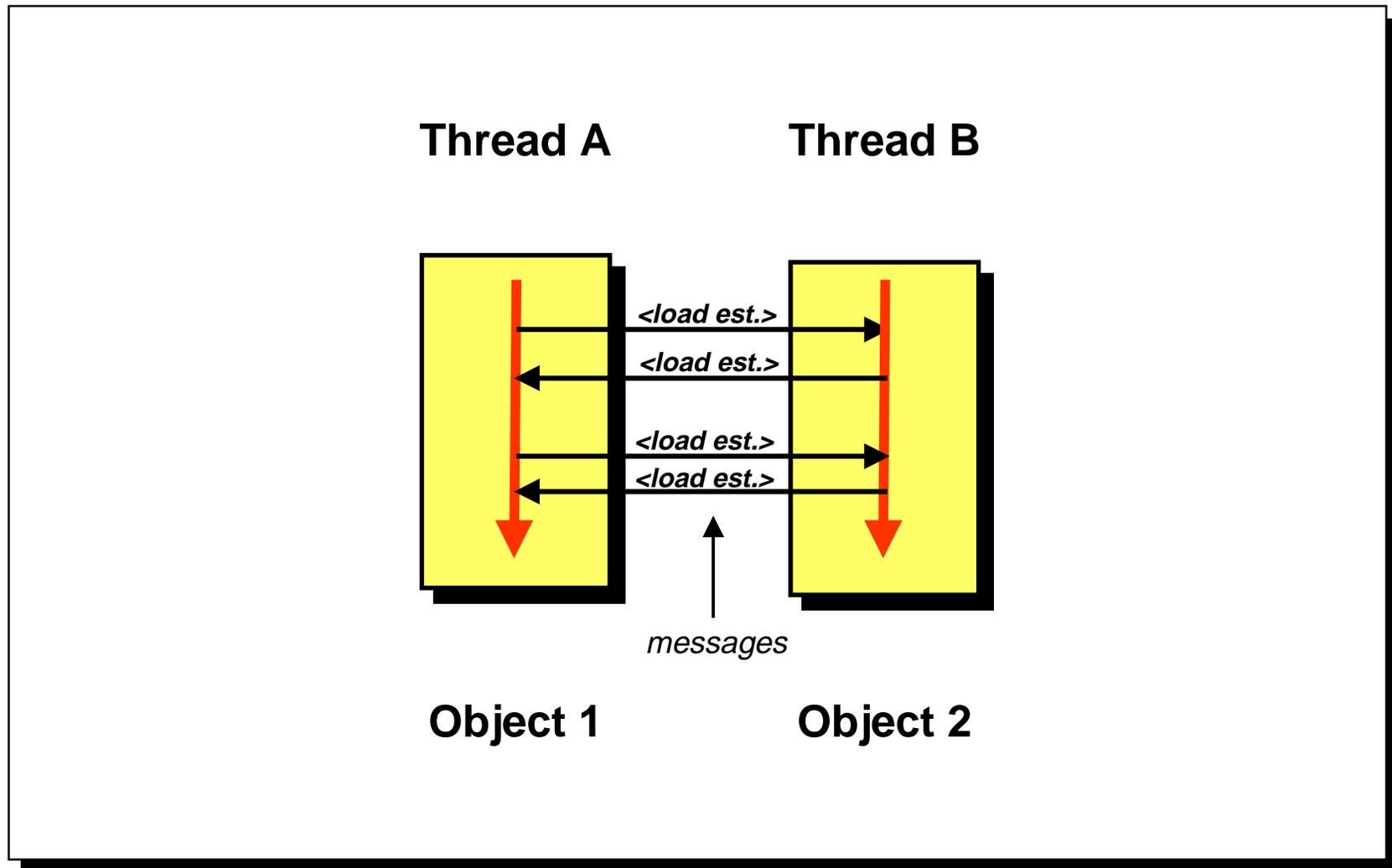
- **Communicating threads**
 - asynchronous interactions
 - synchronous interactions
- **Support for synchronous computations (distributed threads)**
 - simple properties and rationale
 - implications on scheduling (resource management)
- **Distributed threads and asynchrony**

- **Possible topics along the way:**
 - details on material presented
 - possible real-time scheduling policies
 - examples of computational relationships

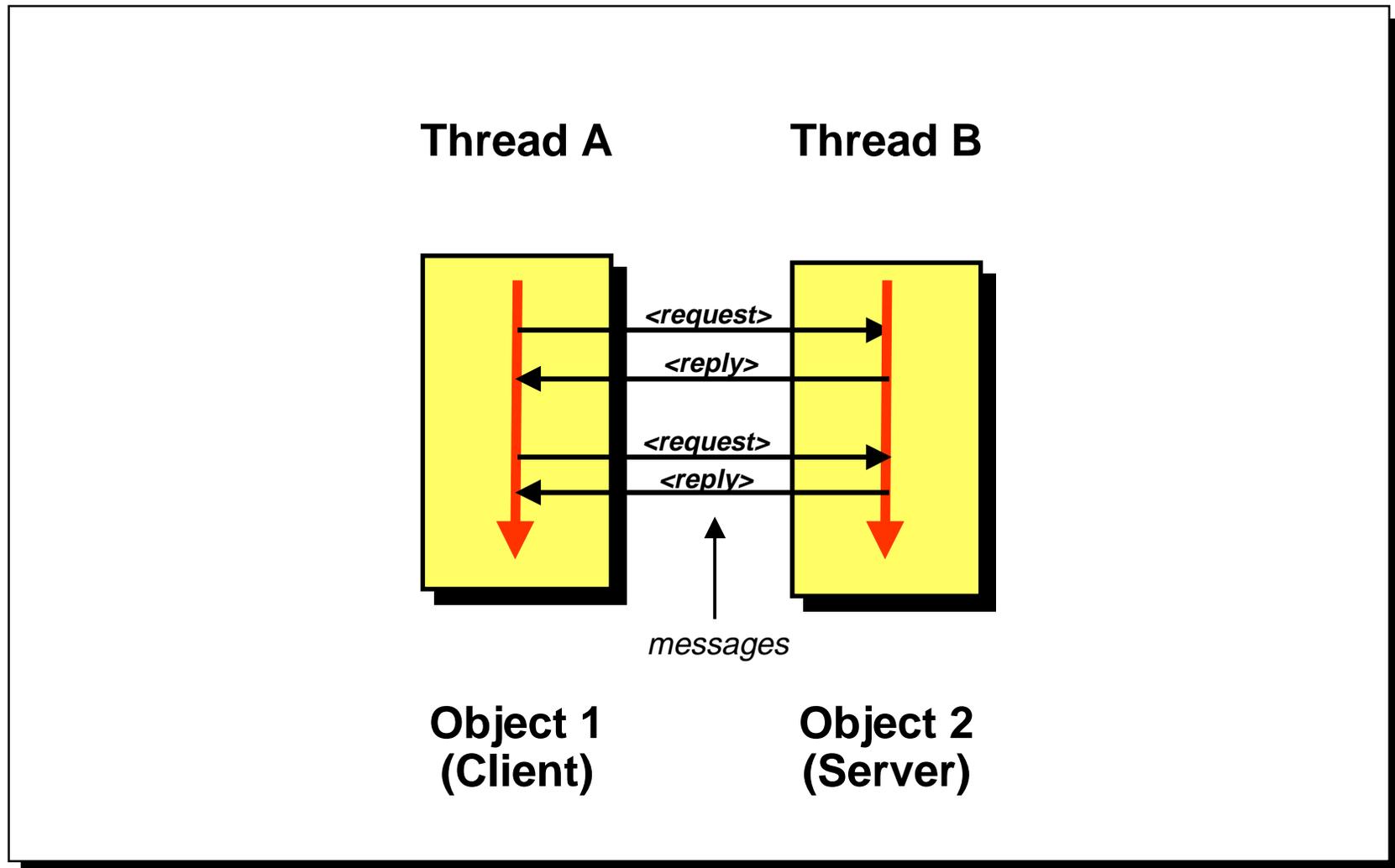
Communicating threads in a message-passing system



Messages might be asynchronous in nature...



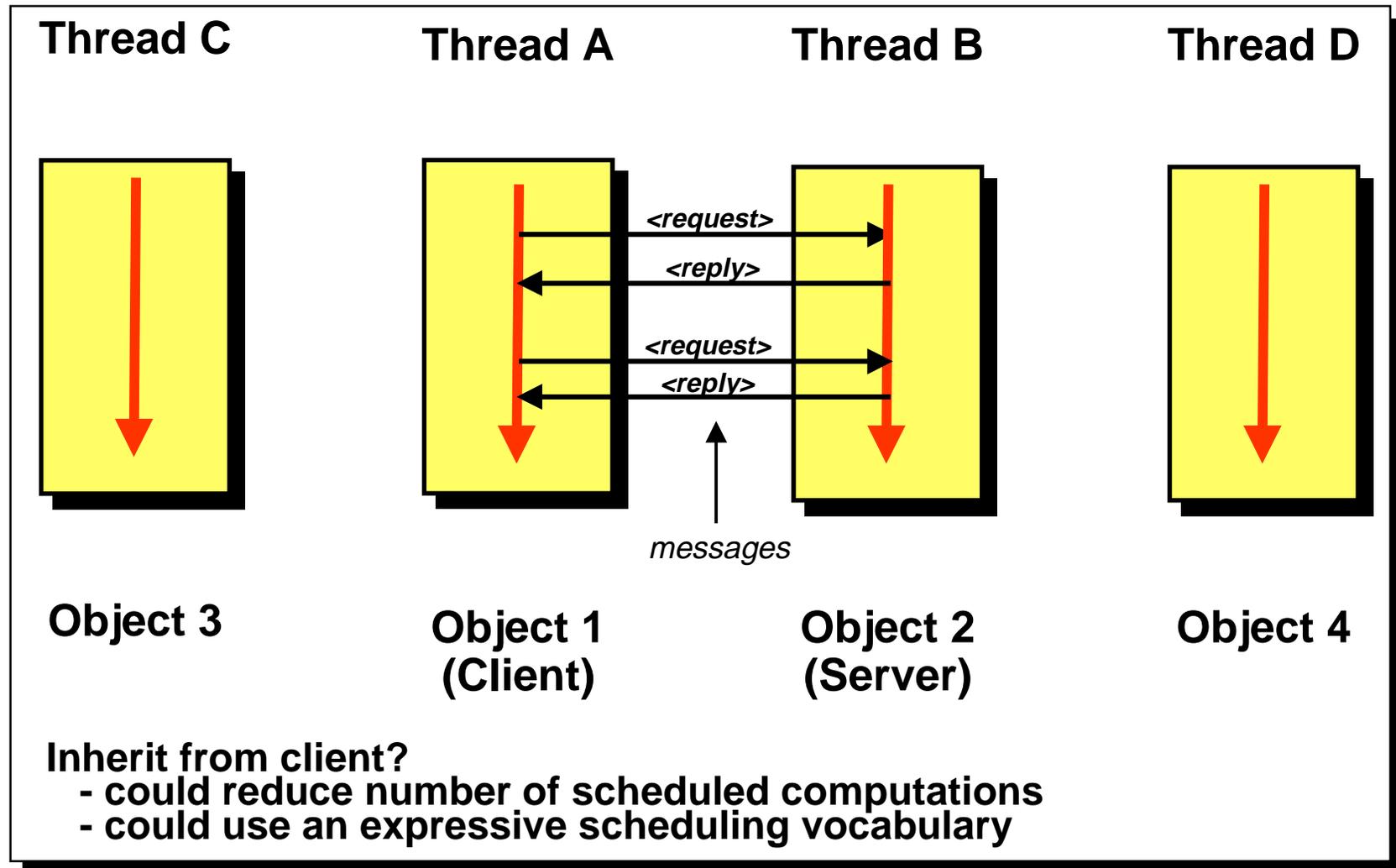
Messages might be synchronous in nature...



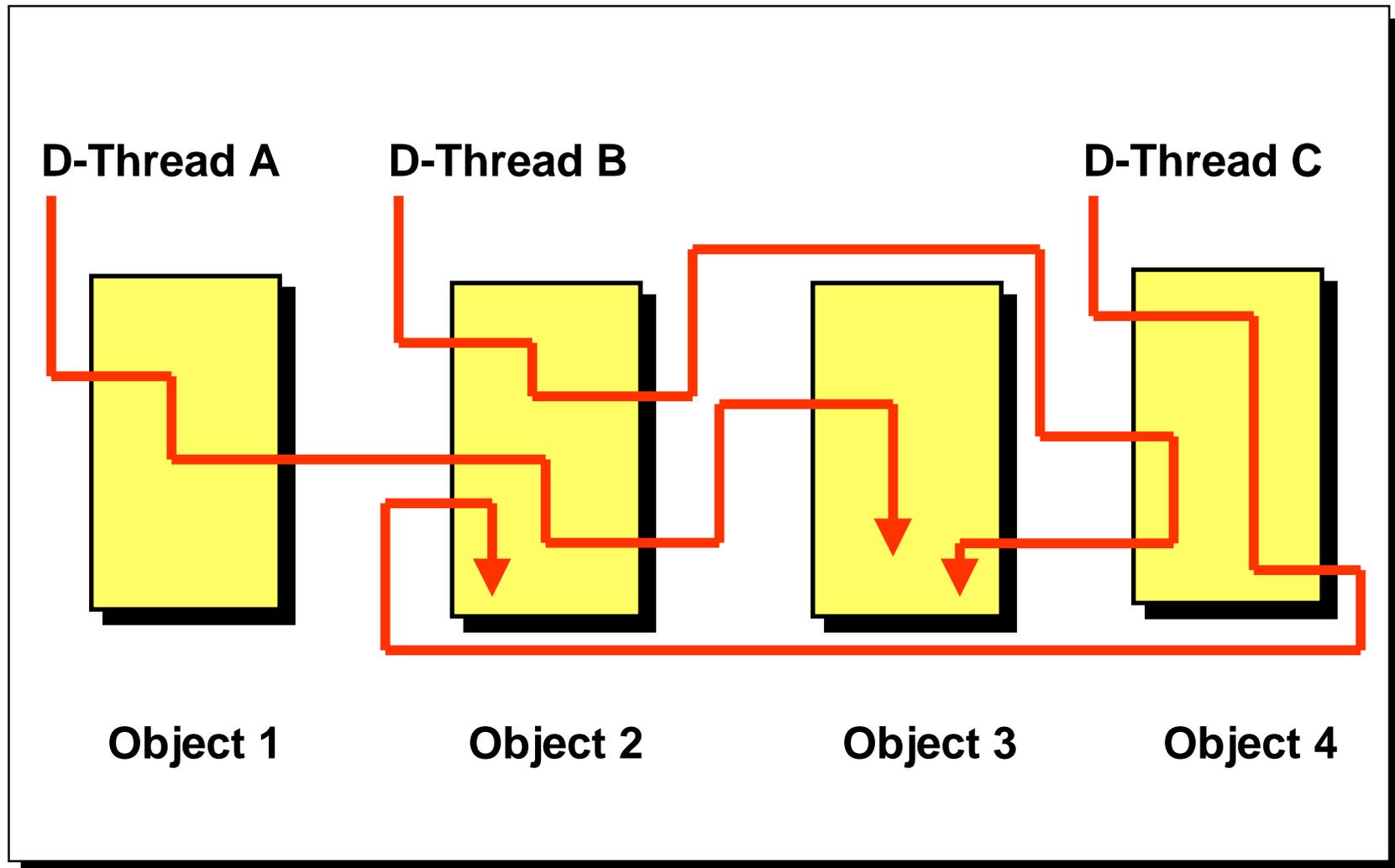
An Important Special Case

MITRE

Synchronous case: How to schedule the server's work?



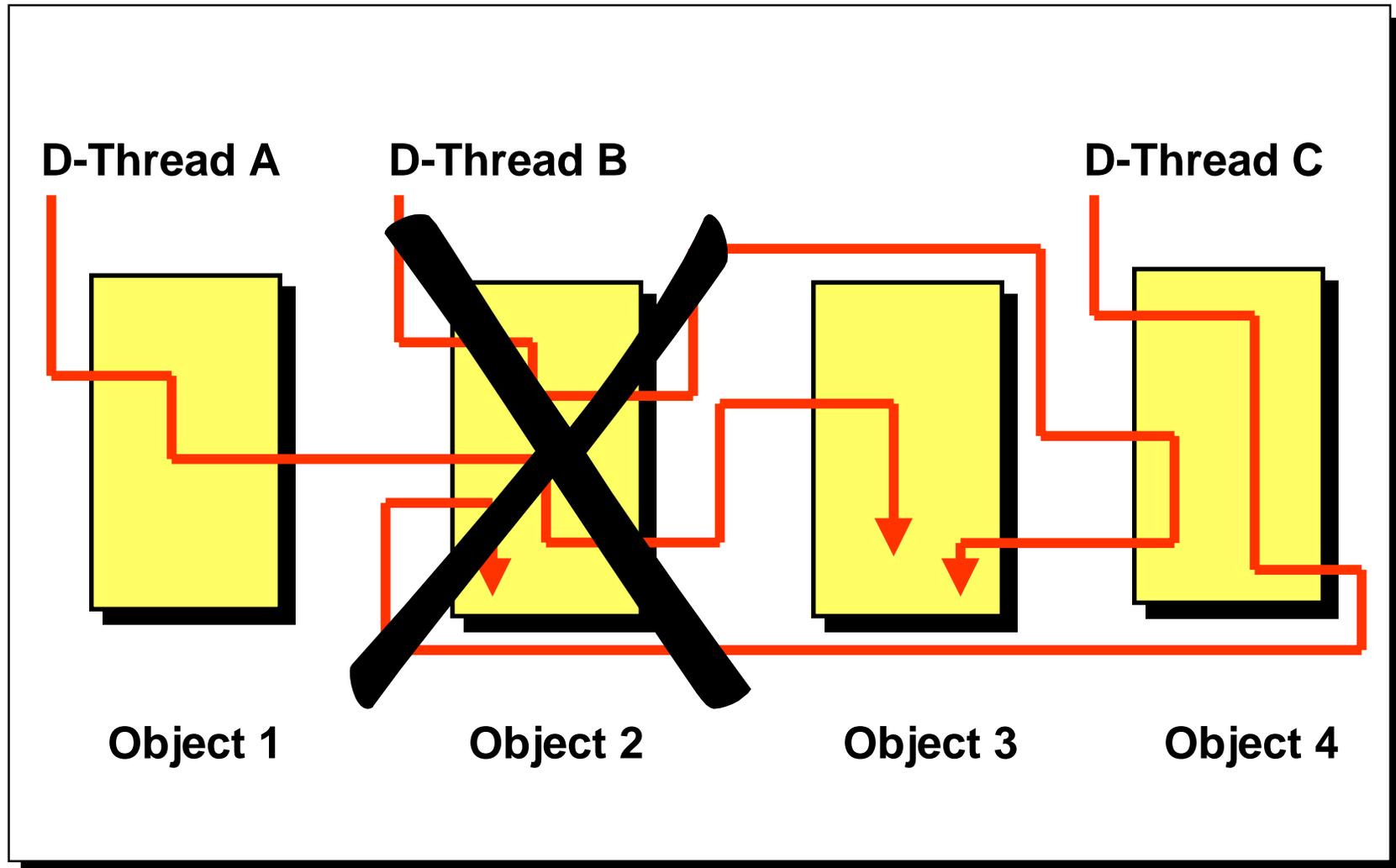
Distributed (trans-node) threads: synchronous interactions (e.g., RPCs)



Bind scheduling information to distributed threads

MITRE

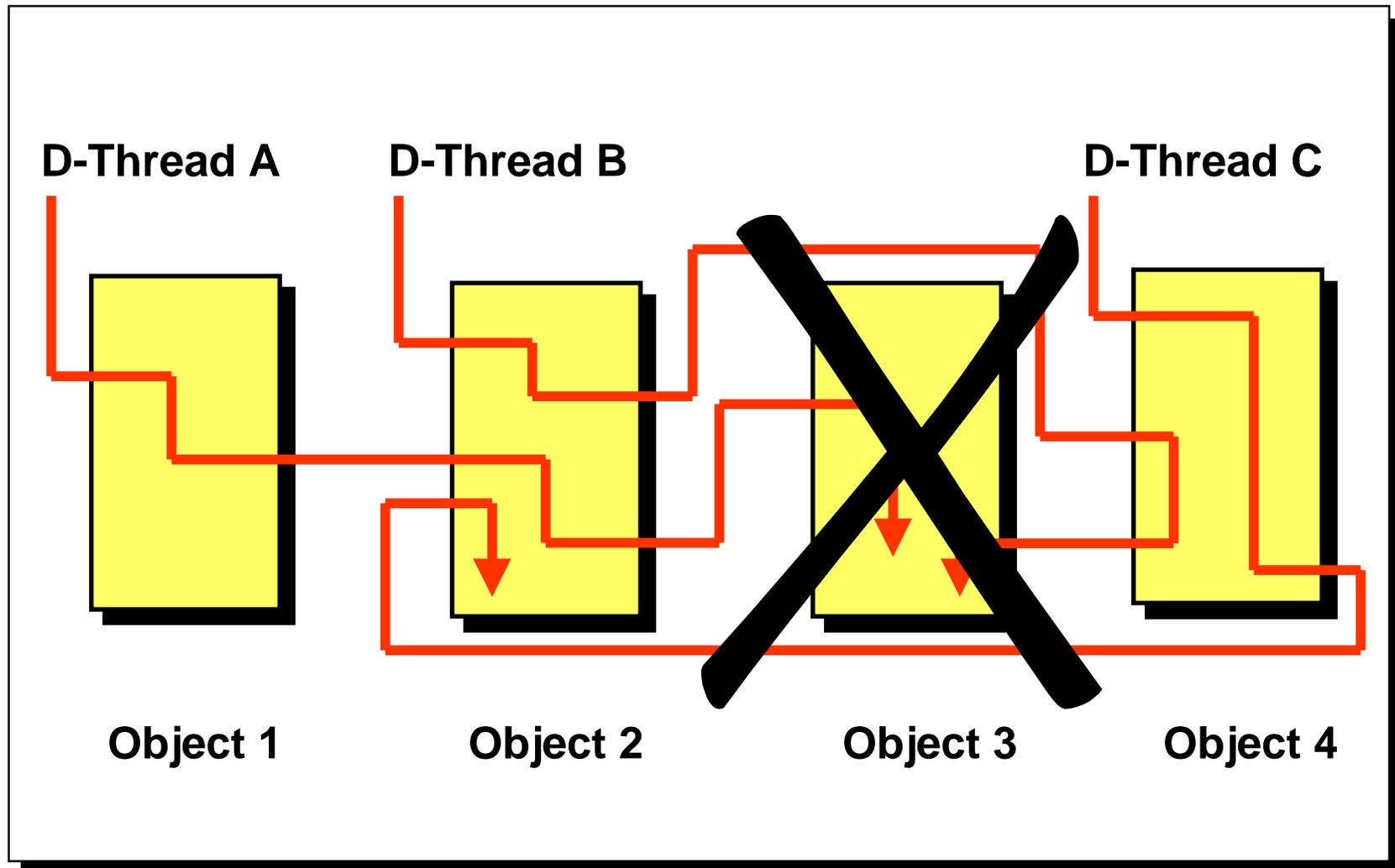
Distributed threads: partial failures



Orphans, thread repair, etc.

MITRE

Distributed threads: support for atomic (trans)actions

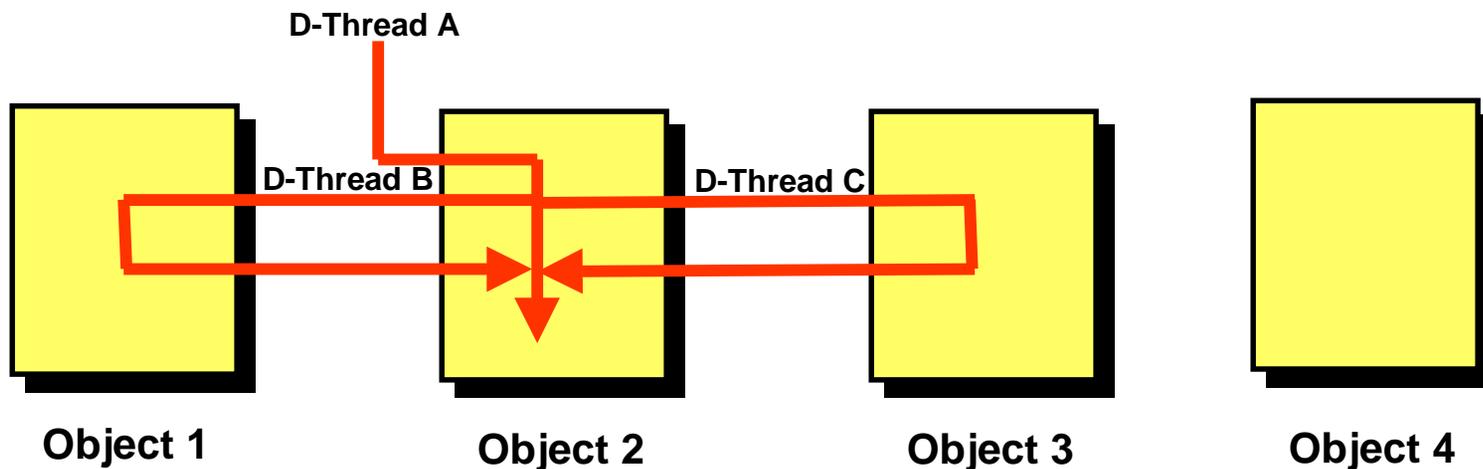


Thread integrity extends to all visited objects

MITRE

What about asynchronous computations?

- **Case I: independent computations**
 - covered by distributed thread abstraction
- **Case II: interacting computations**
 - can utilize distributed threads
 - require higher-level support (e.g., distributed thread groups)



What semantics are desired/required?

MITRE

What about asynchronous computations? *(cont.)*

- **Message passing can be implemented on a synchronous model**
 - message passing can be controlled by manipulation of scheduling parameters (e.g., declaration of message-passing time constraint)
 - there is no particular support for scheduling the receiver's computation
 - what scheduling support would be useful?
 - are there common idioms or patterns that are valuable?