JavaOne℠

Sun's 2002 Worldwide Java Developer Conference

Beyond the Stock Quote: BTP, the Long and Short IT (Internet Transactions)

Mark Potts, CTO, Talking Blocks
Mark Little, Distinguished Engineer, HP

Objectives

- Why: Discuss transactions within the world of Web Services, why they are required, the types of transactions need to be supported (the long and short lived versions) and what requirements and restrictions have to be considered specific to Web Services
- How: The presentation will cover the Business Transaction Protocol (BTP) as an OASIS Committee Specification, and its treatment of all aspects of transactions, specific to supporting the transactional Web

Goals

- Understand why traditional transaction semantics/technologies don’t fit in a Web Services world
- Identify what requirements must be met to support internet transactions
- Understand how the OASIS BTP addresses these needs, its place in the Web Services standards stack, benefits, and required future work

Speakers

- Mark Potts is the Chief Technology Officer at Talking Blocks
- Mark Little is a Distinguished Engineer, Transactions Architect, HP Arjuna Labs
- Both are
  - OASIS BTP technical committee members

Presentation Agenda or Key Topic Areas

- Traditional transactions
- Transactions and web services
- OASIS Business Transaction Protocol (BTP)
- Technology preview and demonstration
- Java™ specification request 156 (JSR-156) and future work

Why the Web Needs Transactionality

“Web Services will not become pervasive without QoS definition”

“Transactions help define QoS and establish trust”

“Traditional Transaction models will not support transactional Web Services!”

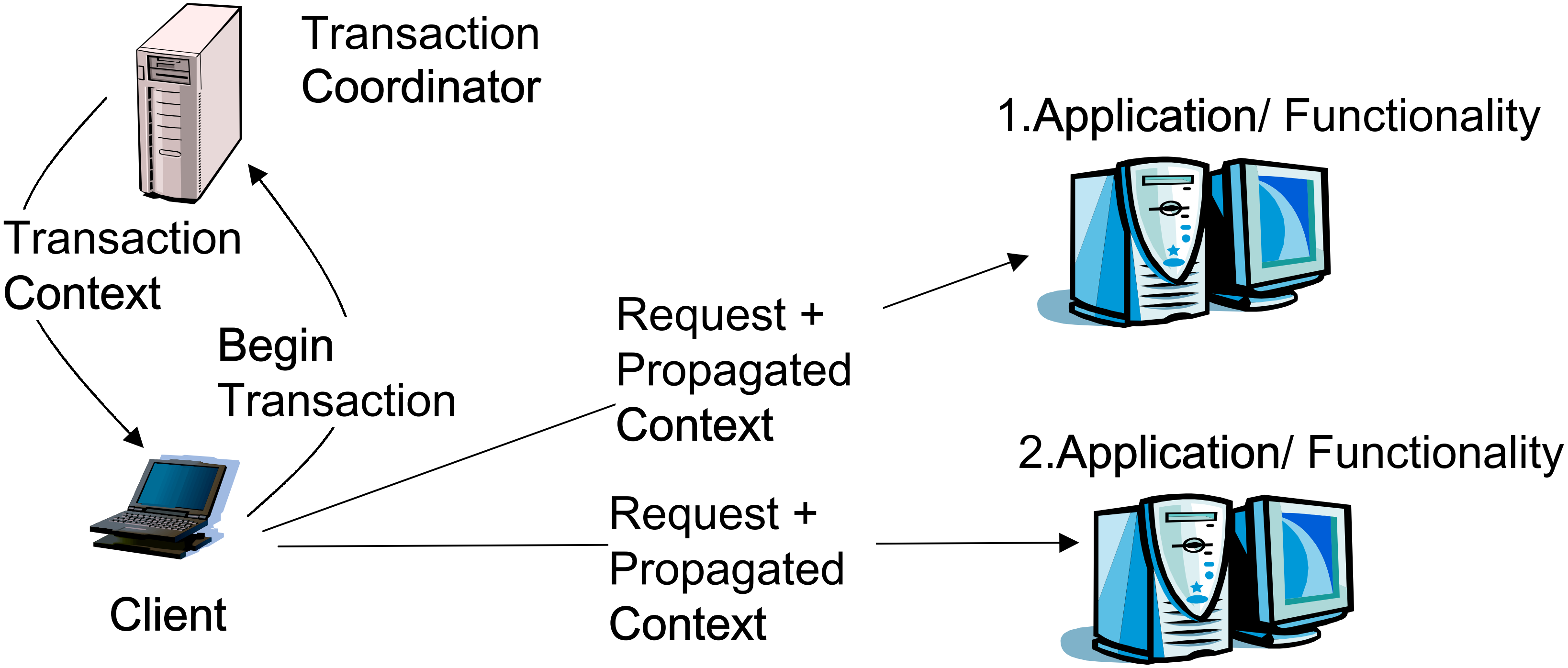
Traditional Transaction Models and ACID Properties

- Traditional transaction systems offer ACID guarantees
  - Atomic
  - Consistent
  - Isolated
  - Durable
- Implicit contract that exists between
  - Transaction coordinator
  - Participants
    - e.g., XAResource

Transactional Roles—Abstract

- Coordinator
  - Does the hard work of ensuring atomicity (including failures)
- Application/Functionality
  - Does the actual business logic, e.g., talks to a back-end database
- Participant
  - Controls the fate of the work done by the transactional object
- Context
  - Flows between end-points and contains information about the transaction such that participants can enroll in the transaction

Transaction Propagation and Enrollment



Transaction Coordinator

Transaction Context

Begin Transaction

Client

Request + Propagated Context

1.Application/ Functionality

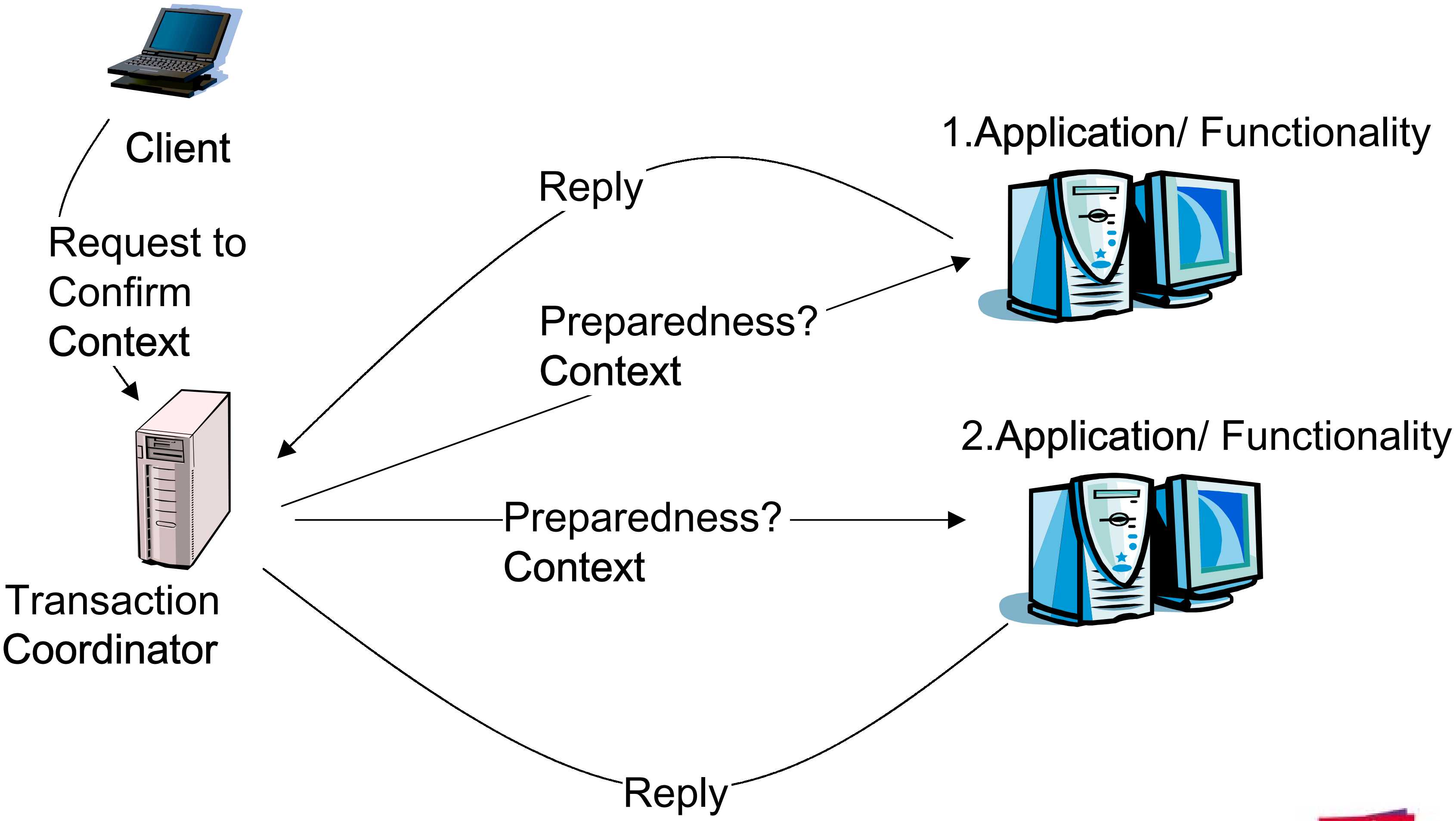
Request + Propagated Context

2.Application/ Functionality

Termination Protocols

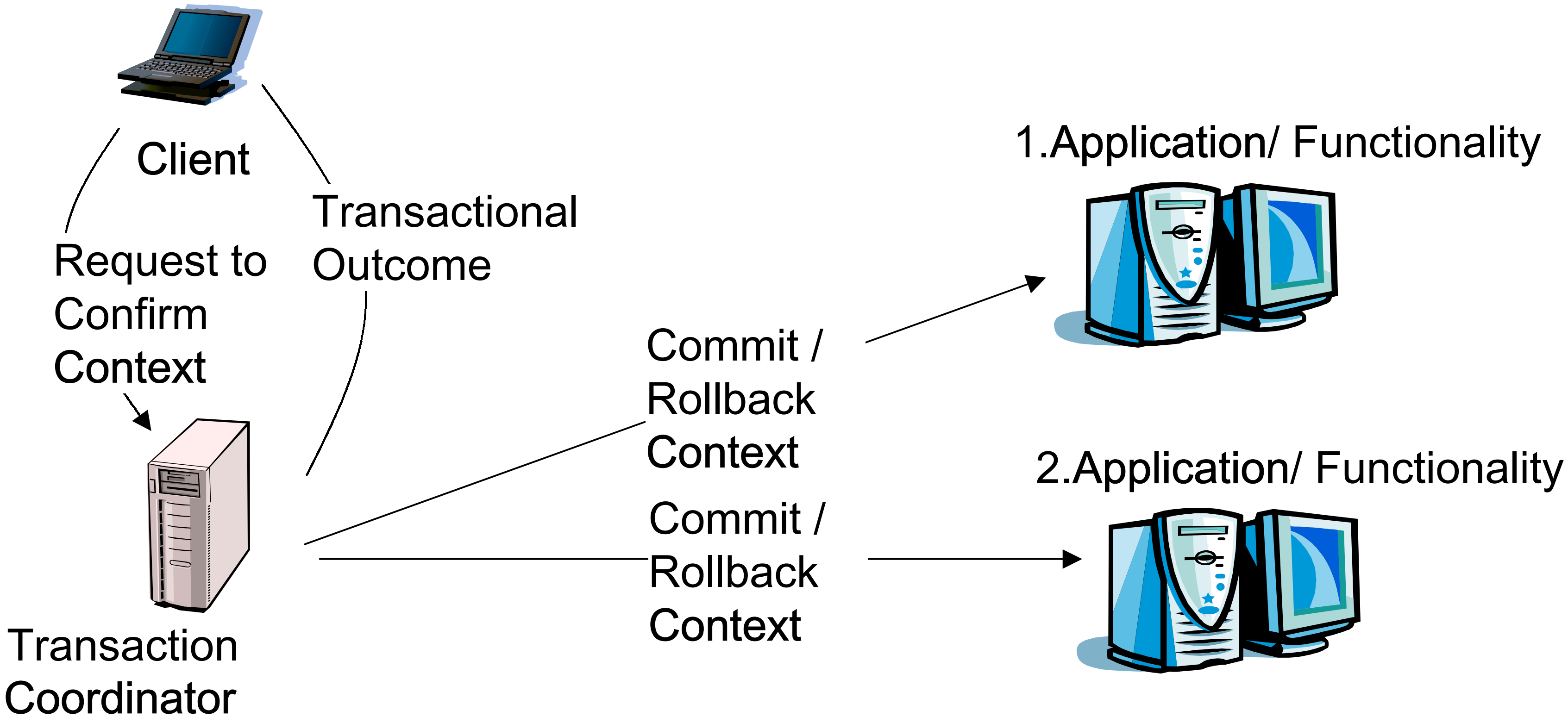
- Typically use a two-phase commit protocol
  - Three-phase protocol also exists
- During commit, the coordinator determines whether or not each enlisted participant can commit the changes
  - Prepare phase
    - Participants that can commit are required to record sufficient information to allow completion if failure
- If all say yes, then proceed to commit the transaction
  - Commit phase
    - Coordinator records sufficient information to complete in case of failure
- If any say they cannot, then proceed to rollback the transaction
  - Rollback phase

Two-Phase Commit Protocol—Phase 1

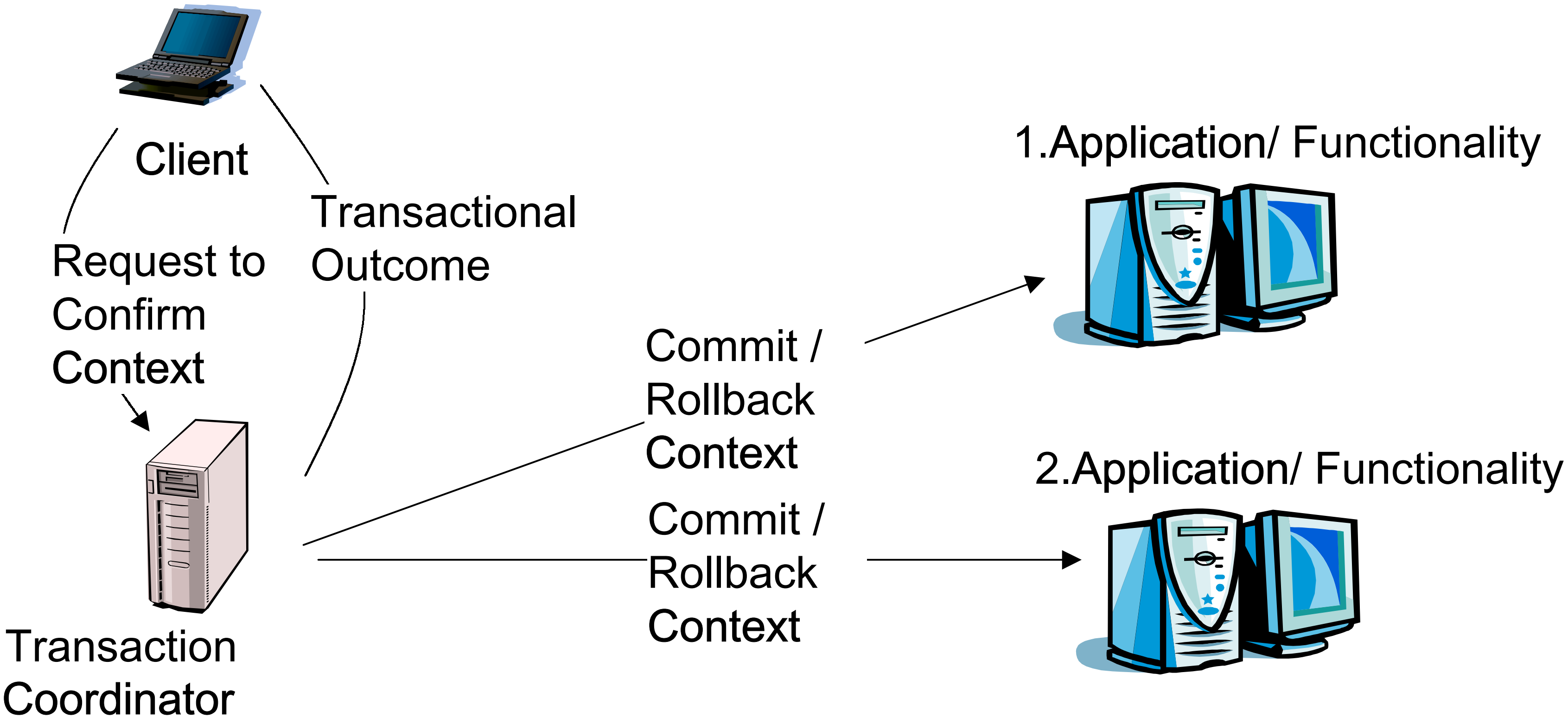


Client
Request to Confirm Context
Transaction Coordinator
Reply
Preparedness? Context
1.Application/ Functionality
2.Application/ Functionality
Preparedness? Context
Reply

Two-Phase Commit Protocol—Phase 2


Client
Request to Confirm Context
Transactional Outcome
Transaction Coordinator
Commit / Rollback Context
Commit / Rollback Context
1.Application/ Functionality
2.Application/ Functionality

Two-Phase Commit Protocol—Phase 2


Client
Transactional Outcome
Request to Confirm Context
1.Application/ Functionality
Commit / Rollback Context
2.Application/ Functionality
Commit / Rollback Context
Transaction Coordinator

2PC is a protocol and does not define transaction qualities - i.e. ACID or isolation levels i.e. two phase locking.

However…

- ACID transactions are good for “short”-duration activities
  - Seconds, minutes, …
- Resources must remain “locked” for the duration of the transaction
  - Early release of resources may cause cascade-rollback
- Coordinator failure may leave resources locked for extended durations
  - Reduces the concurrency in the system

Environmental Impact

- ACID transactions implicitly assume
  - Closely coupled environment
    - All entities involved in a transaction span a LAN, for example
  - Short-duration activities
    - Must be able to cope with resources being locked for periods
- Therefore, do not work well in
  - Loosely coupled environments!
  - Long-duration activities!

The Wonderful World of Web Services!

- Business-to-business interactions may be complex
  - Involving many parties
  - Spanning many different organizations
  - Potentially lasting for hours or days
    - e.g., the process of ordering and delivering parts for a computer which may involve different suppliers, and may only be considered to have completed once the parts are delivered to their final destination
- B2B participants cannot afford to lock resources exclusively on behalf of an individual indefinitely
  - Rules out the use of atomic transactions

BTP Beginnings

- World is composed of closely coupled environments glued together by loosely coupled infrastructure
  - We already have the closely coupled world tied up
    - Java™ 2 Platform, Enterprise Edition, OTS…
- Even if closely coupled solutions could be tailored for Web Services they would have problems
  - Firewalls!
  - One company’s protocol may not interoperate with another’s

BTP and Web Services

- Transaction information must leverage the existing WS standards and initiatives
- ACIDity, specifically isolation needs to be relaxed such that parties can negotiate the transactional commitments at runtime
  - Should also support ACID, consensus between participants, as illustrated in an atomic transaction, is extremely useful
- Use a two-phase protocol since it is easier to understand
  - Again 2PC does not infer ACIDity, or locking!

Relaxing Isolation

- Loose coupled or long running activities need negotiated commitments
    - Consider Hotel bookings (optimistic)
    - Flight bookings (pessimistic)
    - Autonomous decisions are key!
- Internal isolation or resources should be a decision for the service provider
    - Commit early and define compensation activities
- Compensation can be whatever is required
    - Before and after image
    - Other business processes
    - E-mail the DBA

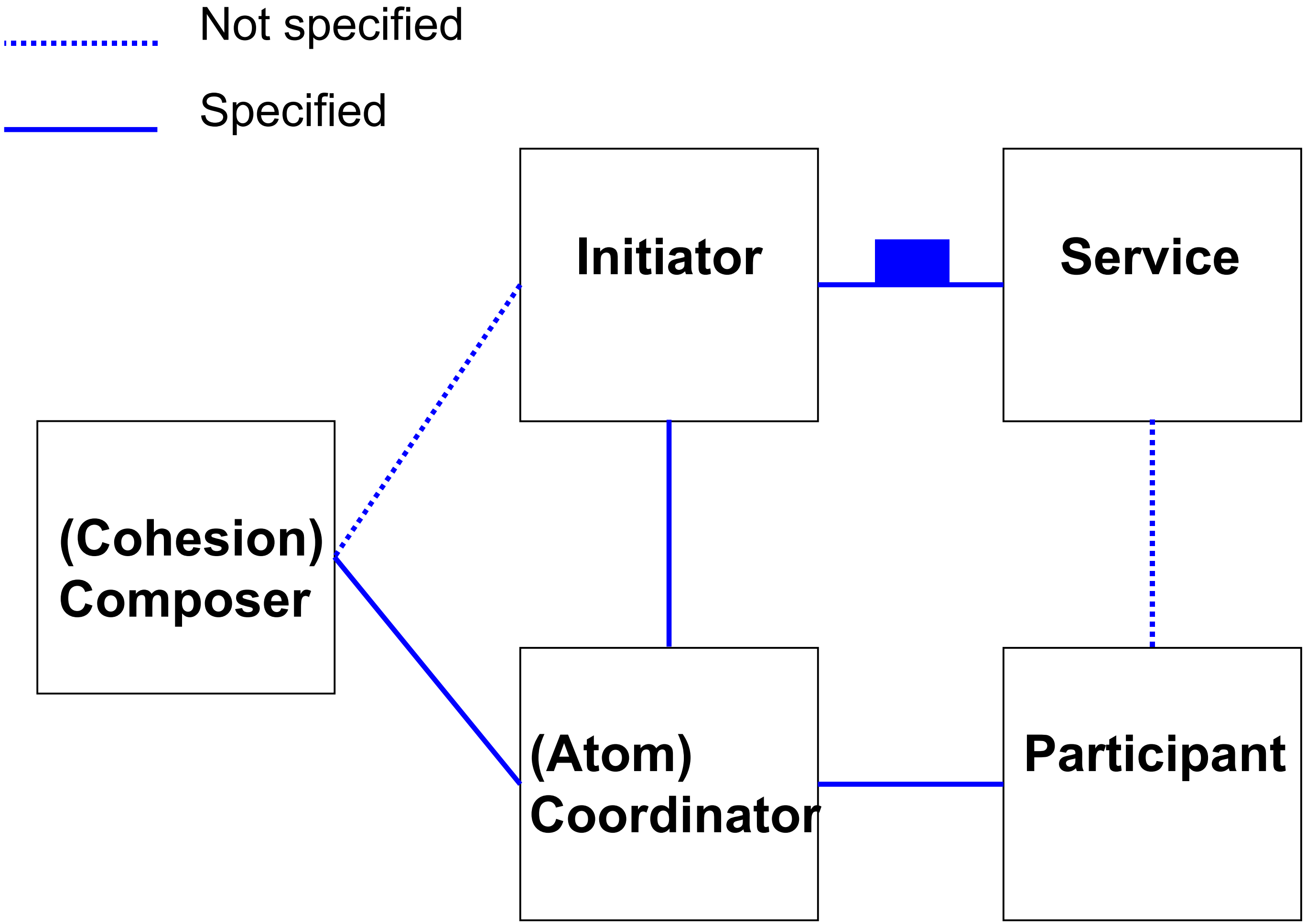
Relaxing Atomicity

- Sometimes in a loosely coupled or long running activity it may be desirable to cancel some work without affecting the remainder
- Very similar to what nested transactions give us
  - Work performed within scope of a nested transaction is provisional
  - Failure does not affect enclosing transaction
- Nested transactions require nested-aware resources
  - Not many of these around

Atoms and Cohesions

- An Atom is very similar to a traditional transaction coordinator
  - Work is done by services within the scope of an Atom
  - The Atom guarantees that all participants will see the same outcome (atomic)
- Cohesions allow the selective confirm (commit) or cancel (rollback) of participants
  - Cohesion participants are Atoms!
  - Not atomic

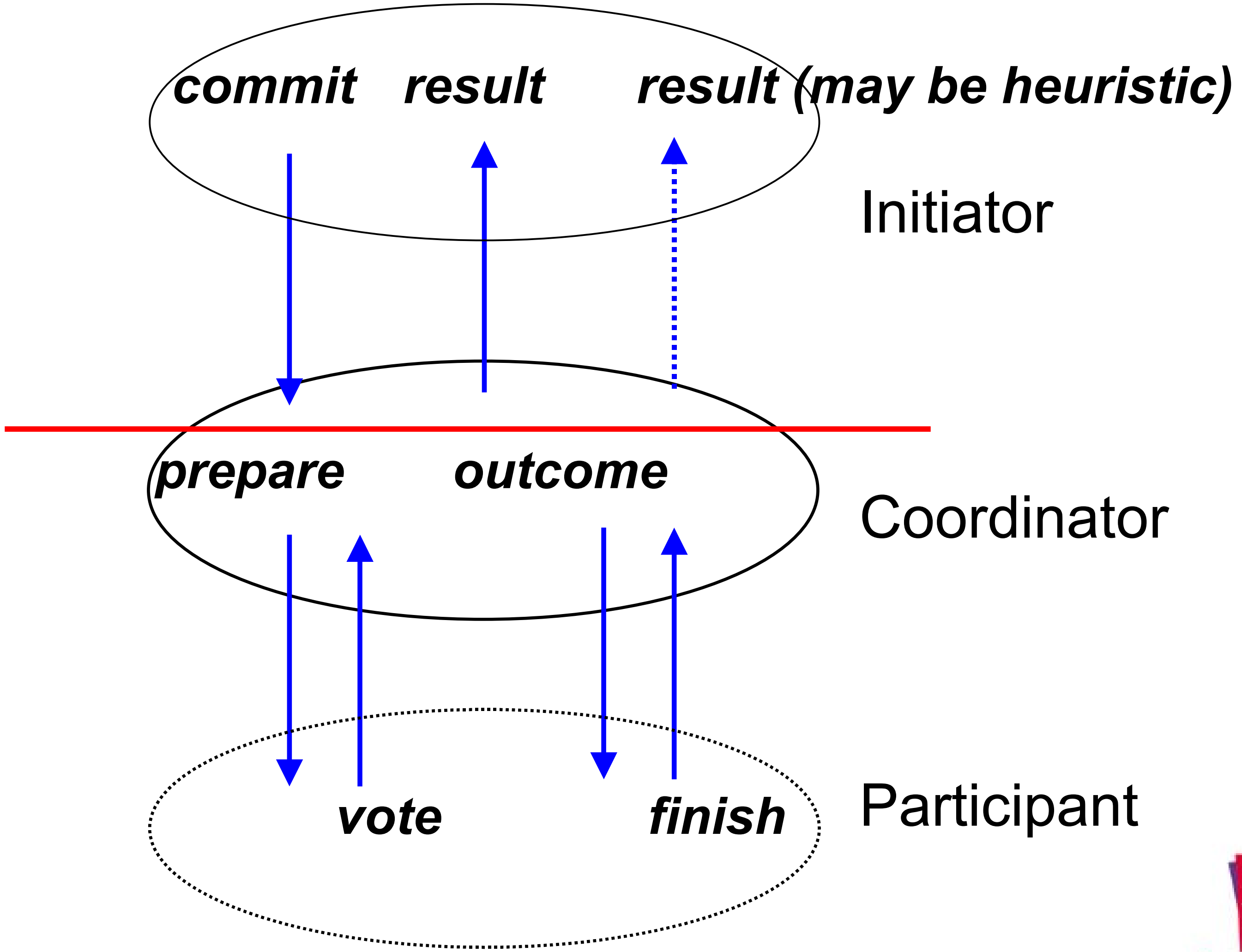
Schematic BTP



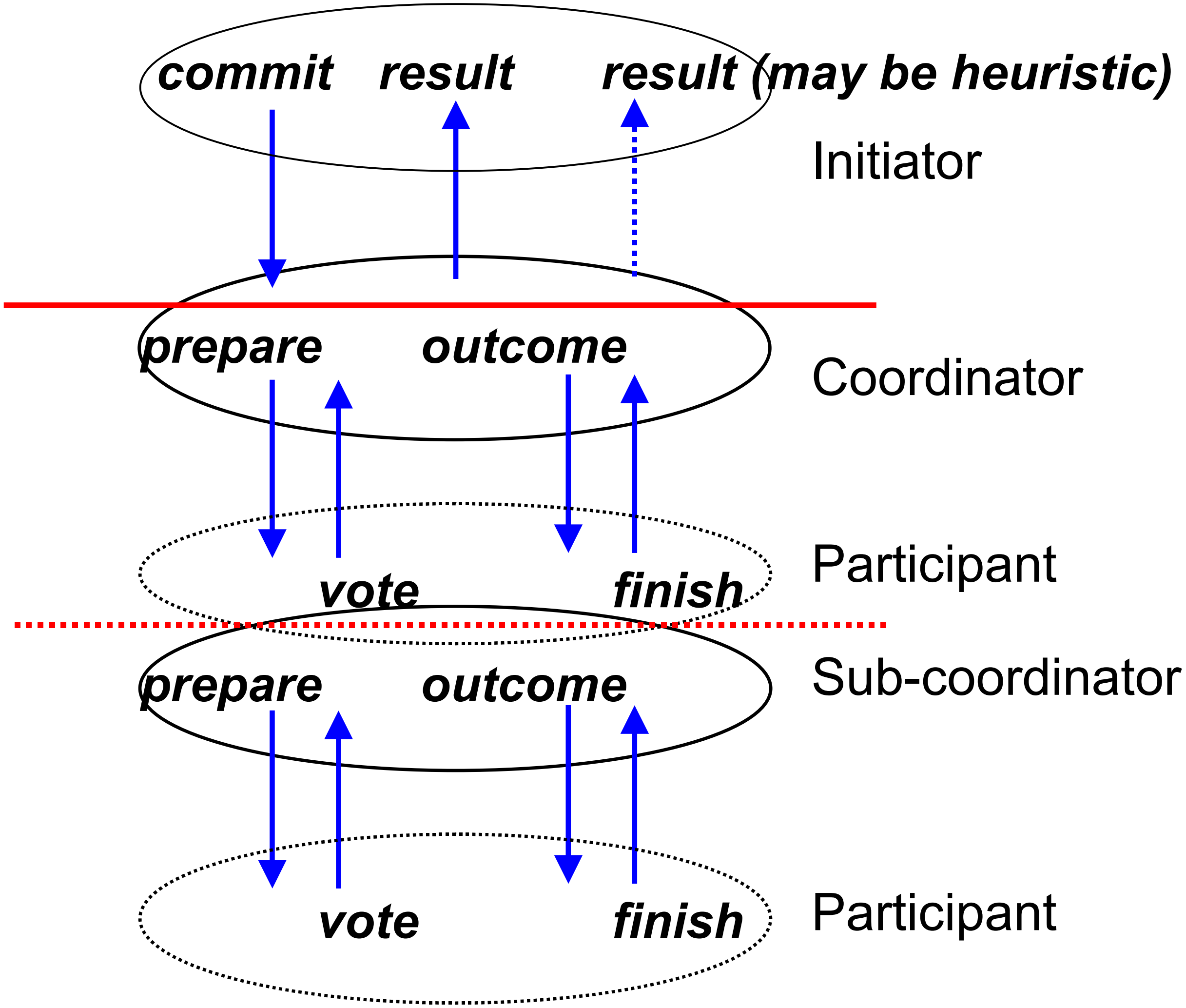
Atoms

- An Atom encompasses a two-phase termination protocol
  - Prepare, confirm and cancel
  - There is an implicit contract between Atom and participant that work must be atomic
    - All participants will do the same thing
    - Does not mandate how to implement prepare, confirm and cancel
  - Does not say anything about isolation
    - Down to individual services to determine
  - Recoverable!

Schematic— “Closed Top” Coordinator



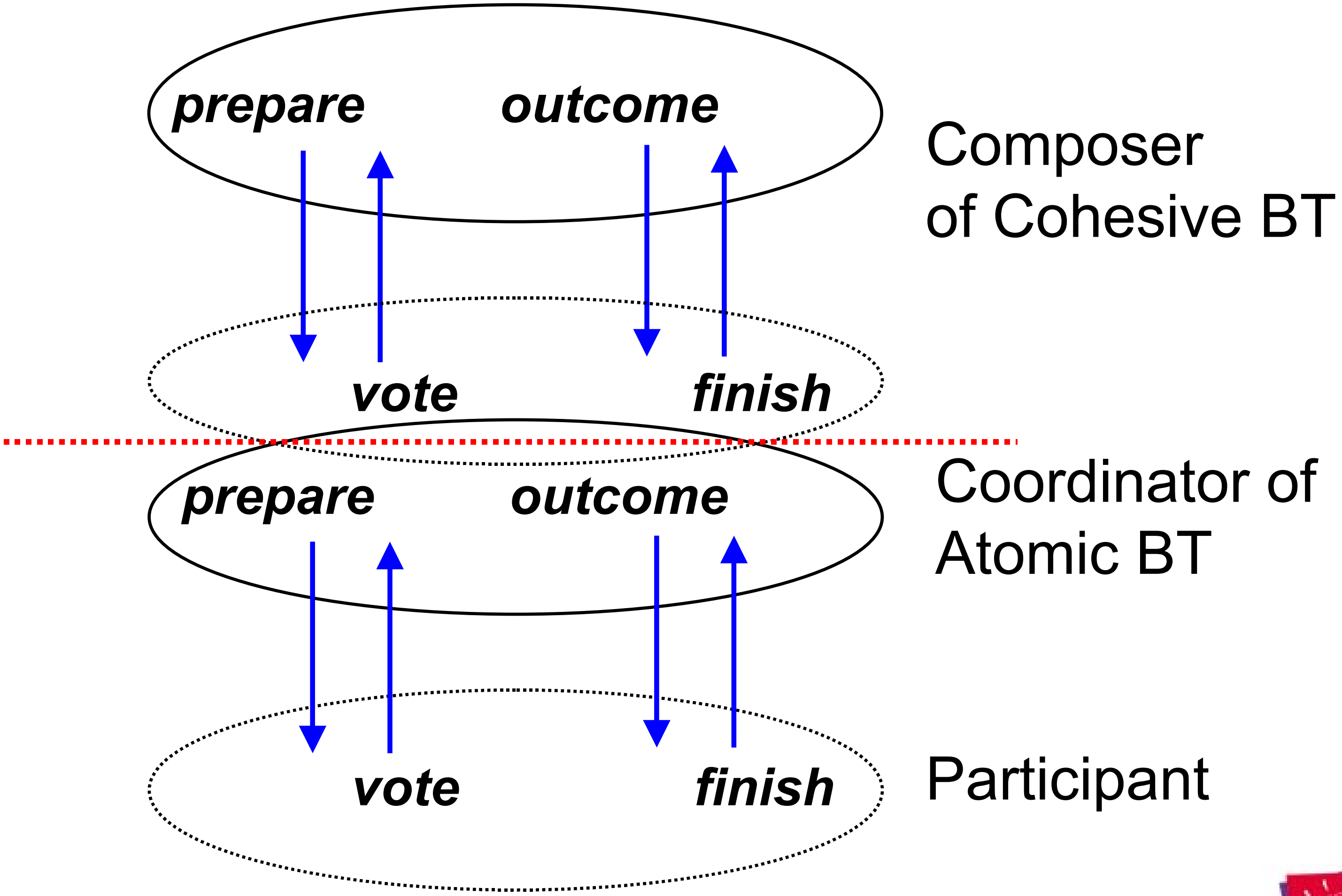
Schematic— Interposed (Sub-) Coordinator



Cohesions

- Similar to an Atom, in that it has a two-phase termination protocol and participants
  - Atoms are the participants
  - Prepare, confirm and cancel are parameterised
    - Work on (set of) Atom id(s)
      - Allows the confirm of a specific subset of work
    - Once subset is determined by business logic, it will be atomic
- All simple when things go well but failures bring complexity!

Schematic—Cohesive Composer “Open Top” Coordinator



Roles and Responsibilities

- Coordinator (atom)
  - Responsible for informing enlisted participants about whether they should accept (confirm) or reject (undo) the work done within the scope of a given atom
- Initiator
  - Communicates with an atom manager (factory) and asks it to start a new atom
    - Once created, information about the atom (the context) can be propagated to web services in order for them to associate their work with it, i.e., work is conducted within the scope of an atom

Roles and Responsibilities

- *Terminator*
  - Completes the atom in a specific state—essentially success or failure
- *Web Service*
  - Part of the actual application/business logic
    - Context flows to this

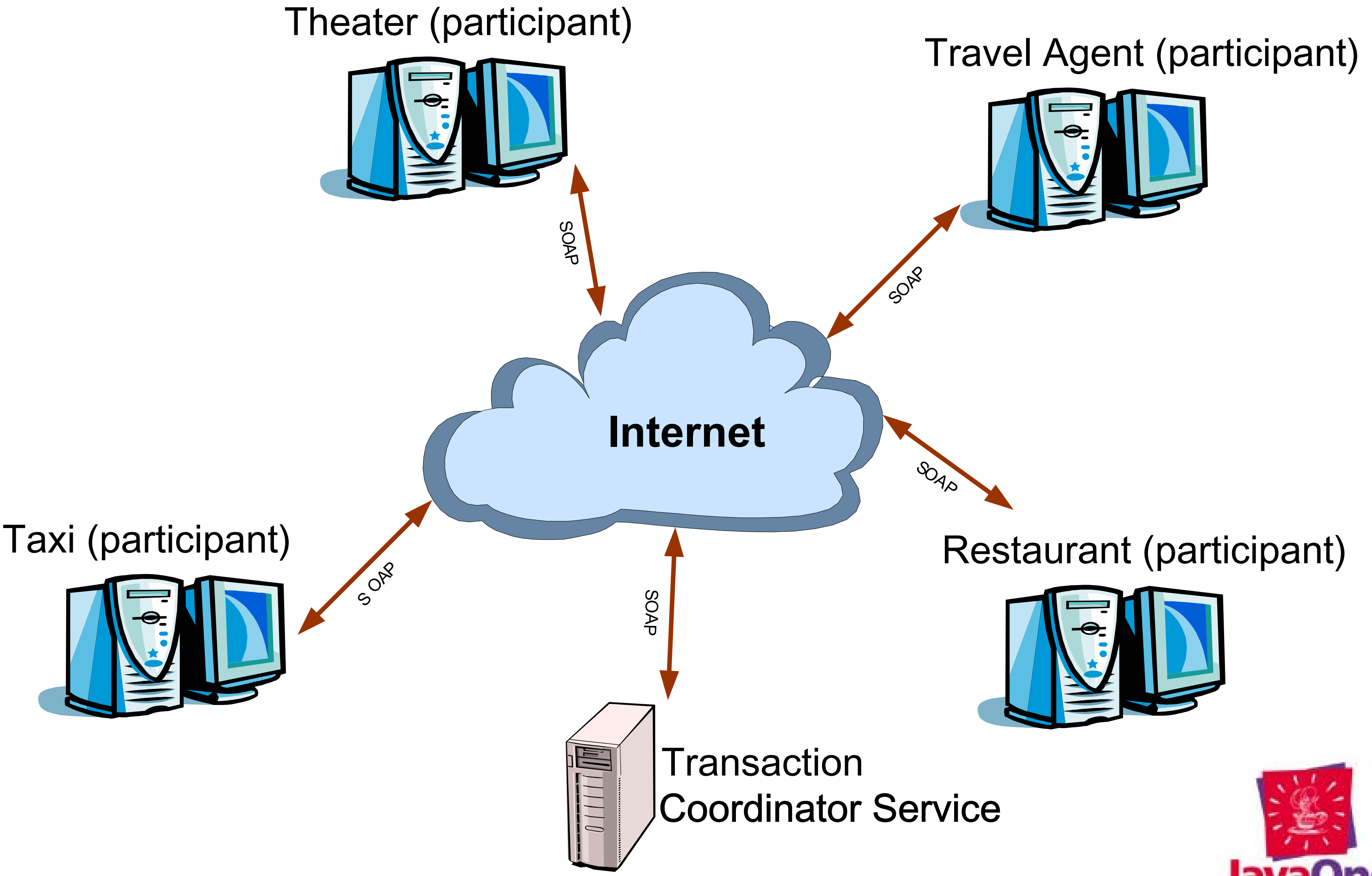
Roles and Responsibilities

- *Participant*
  - Supports a two phase termination protocol via the *prepare, confirm* and *undo* operations
    - Controls the actual fate of work performed by the Web Service
- *Cohesion Composer*
  - The business logic for gluing together the flow of the application into one or more atoms

Interoperability

- BTP has been designed from the outset to allow different vendors to supply different components
  - Coordinator
  - Participant
- Contexts and entire message set has been designed to be interoperable
  - Does not mandate a specific carried protocol
    - Could be SOAP, IIOP, carrier pigeon
  - Only mandates XML format for messages

Demonstration



Conclusion

- ACID transactions are good for some things
  - Never intended as a global panacea
- BTP is a solution to a specific problem
  - Only game in town (as far as standards are concerned)
    - Interesting cast on existing protocols
    - Two-phase commit with extensions
  - Protocol agnosticism may well be important
- For more info go to
  - http://www.oasis-open.org/committees/business-transactions/

Imperatives—Call to Action

Imperatives—Call to Action

“Web Services <u>will only</u> become pervasive within environments of trust, and where they offer business value or ROI”

Imperatives—Call to Action

"Web Services will only become pervasive within environments of trust, and where they offer business value or ROI"

Stock Quote services don't!

Imperatives—Call to Action

"Web Services will only become pervasive within environments of trust, and where they offer business value or ROI"

Stock Quote services don't!

- Consumers and Providers
  - Look into BTP it offers an **opportunity** to layer transactionality into Web services, increasing QoS
  - Look at leveraging BTP in **new business models** or opportunities (more dynamic business, negotiation)

Imperatives—Call to Action

“Web Services <u>will only</u> become pervasive within environments of trust, and where they offer business value or ROI”

Stock Quote services don’t!

- Consumers and Providers
  - Look into BTP it offers an **<u>opportunity</u>** to layer transactionality into Web services, increasing QoS
  - Look at leveraging BTP in **new business models** or opportunities (more dynamic business, negotiation)
- Standards Bodies
  - Initiatives need to **<u>consolidate</u>** and **<u>accelerate</u>** (bindings and support for existing standards within the stack, WSDL, ebXML, RosettaNet, XLANG, WSFL, BPML)

Q&A

JavaOne℠

Sun's 2002 Worldwide Java Developer Conference™

BEYOND BOUNDARIES