The development of Web Transactions

Mark Little, Distinguished Engineer, HP

Overview

- ACID transactions and why they don't cut it in the world of Web Services
  - Consider long-duration activities
- What is BTP?
- Is BTP the final solution?
- What else is required for transactional Web Services?
  - The transaction system is just one piece in the puzzle

Traditional transaction models and ACID properties

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

Standard implementations

- Object Transaction Service
  - Two-phase commit and two-phase locking
  - Any two-phase transactional resource may be enlisted with an OTS transaction
    - File system, database, …
- Java Transaction Service
  - OTS language mapping
- Java Transaction API
  - Simplification layer on top of JTS
    - XA specific
      - Only XA compliant resource managers can be used

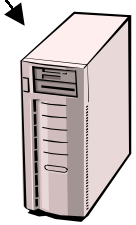
Transactional roles - abstract

- 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 enrol in it

2PC – phase one

Client

Request to Confirm **Context**

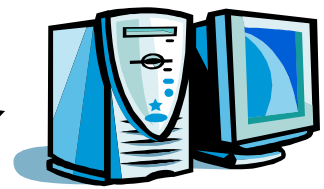


Transaction **Coordinator**

Reply

Preparedness? **Context**

1.**Application**/ Functionality

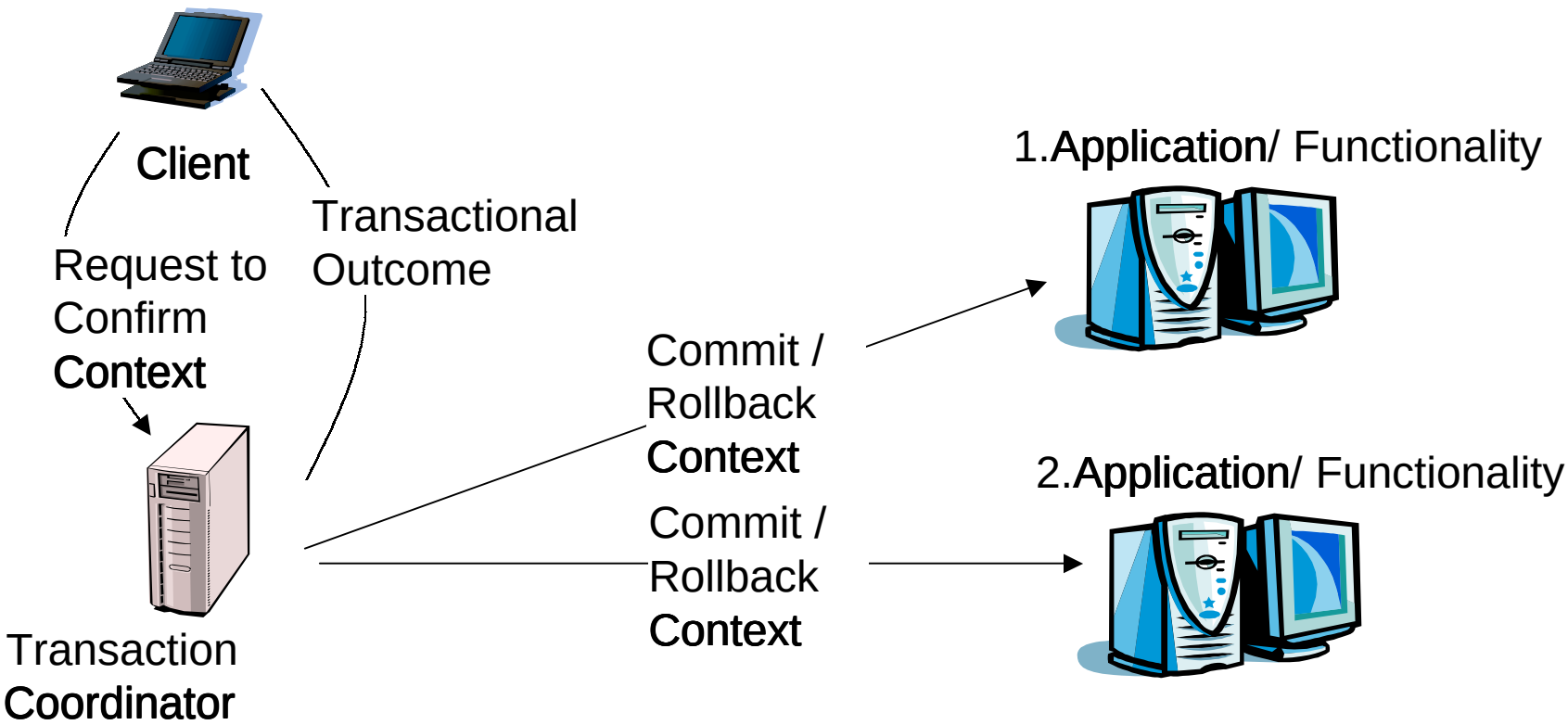


Preparedness? **Context**

2.**Application**/ Functionality

Reply

2PC – phase two



2PC is a consensus protocol and does not define transaction qualities - e.g., two phase locking.

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!

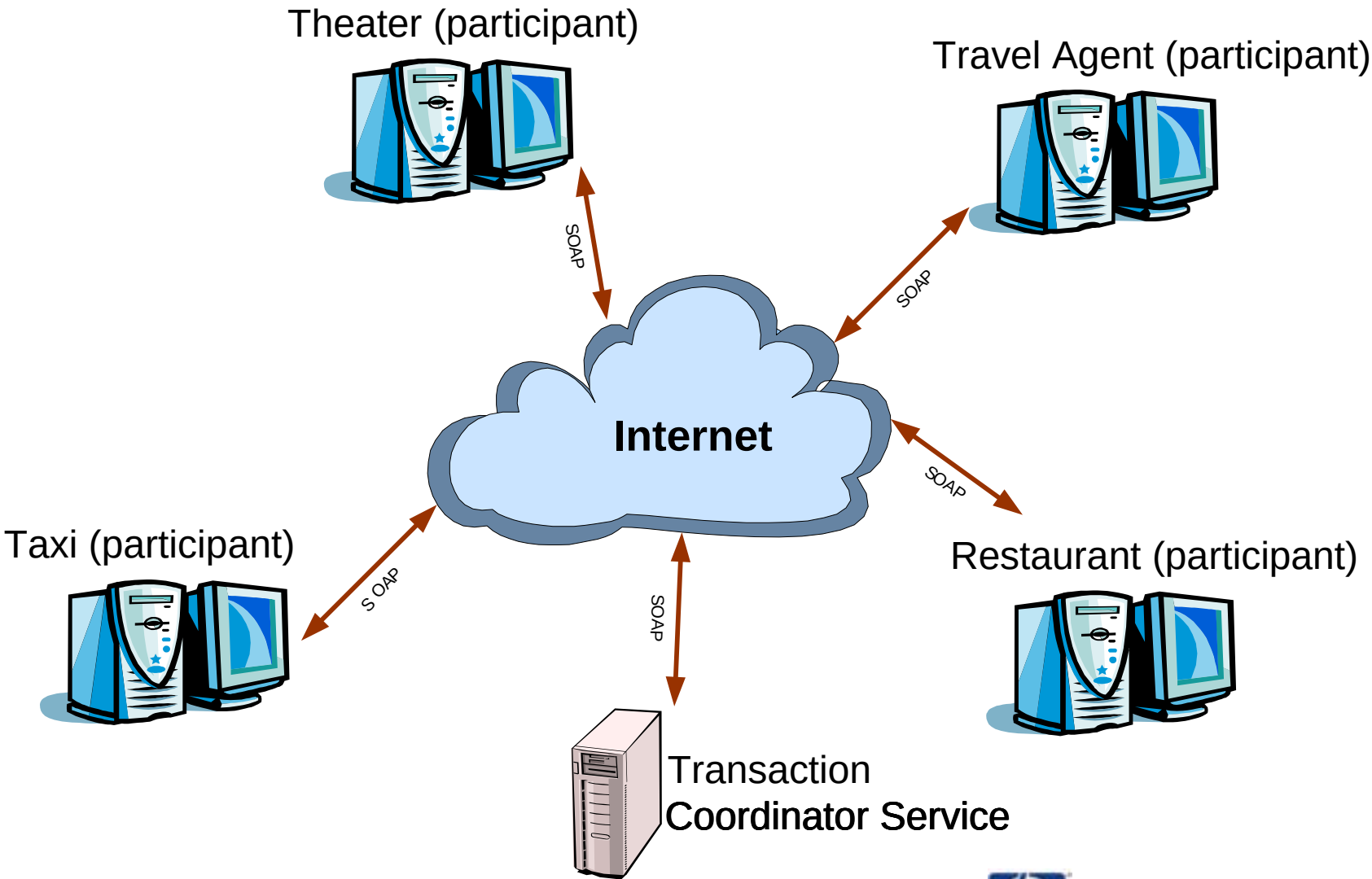
The wonderful world of Web Services!

- Business-to-business interactions may be complex
  - involving many parties
  - spanning many different organisations
  - 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

OASIS BTP

- Two "types" of transaction coordinator
  - Atom
    - All-or-nothing effect for participants
  - Cohesion
    - Selective determination of "confirm"-set
- "Open-top" termination protocol
  - Drive two-phases explicitly
    - No enforced time-limit between each phase
- Participants can be anything, as long as contract with coordinator is obeyed
  - E.g., workflow system

Example interaction



Interoperability

- BTP has been designed from the outset to allow different vendors to supply different components
  - Atom Coordinator
  - Cohesion Composer
  - Participant
  - Web Service (!)
- 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

The final solution for Web Services?

- Does one size fit all?
  - No, as proven by OMG work on extended transaction systems
- OTS, JTA, BTP, … all address specific problem domains
  - Plethora of extended transaction models
- Additional models and implementations may well appear in the future
  - Address specific problem domains

However, where's the beef?!

- Most back-end systems and applications will continue to use ACID transactions
  - J2EE, CORBA, vendor specific implementations
  - SOAP and XML are too slow to be useful everywhere
- This is where the real work begins
  - Most resources that people want already exist in these domains
    - For example, JDBC drivers

The real world

- B2B interactions take place between businesses!!
  - Businesses have existing application infrastructures that they're not going to dump
  - Lots of faith in things like
    - CICS, Tuxedo, HP-TS
    - Oracle, MSSQL
    - JDBC drivers
    - J2EE, CORBA
- World is separated into domains
  - Structured and trusted (e.g., J2EE)
  - Unstructured and untrusted (e.g., the Web)

Where does BTP fit then?

- It is the piece in the puzzle that has been missing up until now
  - It provides the glue between domains
    - Businesses can concentrate on their internal domains with existing infrastructure
    - Then concentrate on the external nature of their applications and transactions
- Is not sufficient by itself
  - Provide end-to-end solution for customers
    - It is critical to the take-up of transactions in Web Services

Conclusion

- ACID transactions are good for some things
  - Never intended as a global panacea
- BTP is a solution to a specific problem
  - Interesting cast on existing protocols
    - Two-phase commit with extensions
  - Protocol agnosticism may well be important
- End-to-end transactionality is extremely important for e-commerce
  - Integration of OTS/CICS/… and BTP will happen