Arjuna Web Services transactions overview

Mark Little

Chief Architect

(mark.little@arjuna.com)

arjuna

middleware for reliability

What is ATS?

arjuna
middleware for reliability

- Transaction service supporting
  - Closely coupled transactions (JTA/JTS)
  - Loosely coupled transactions (Web Services)
    - Arjuna XML Transaction Service component
    - Based on
      - WS-Coordination, WS-Atomic Transaction, WS-Business Activity
        - » Demonstrated interoperability with IBM
      - WS-CAF
        - » Demonstrated interoperability with Oracle and IONA
- Used at HP World for Web Services seminars

The Web services component

arjuna
middleware for reliability

- Support for general coordination framework
  - WS-C but also OASIS WS-CF
- Support for Web services transactions
  - Atomic Transaction
    - Traditional two-phase commit
  - Business Activity
    - Similar to OASIS WS-TXM Long Running Action

Implementation overview

arjuna middleware for reliability

- Separate transaction and coordination core from messaging
  - Provide SOAP portability infrastructure
    - There's no such thing as a standard SOAP server implementation!
- Transaction services plugged in
  - Allows us to leverage existing implementations
    - E.g., ArjunaCore, Tuxedo, CICS, Encina
  - Customers trust pedigree
    - Especially where their money is concerned

Deployment infrastructure

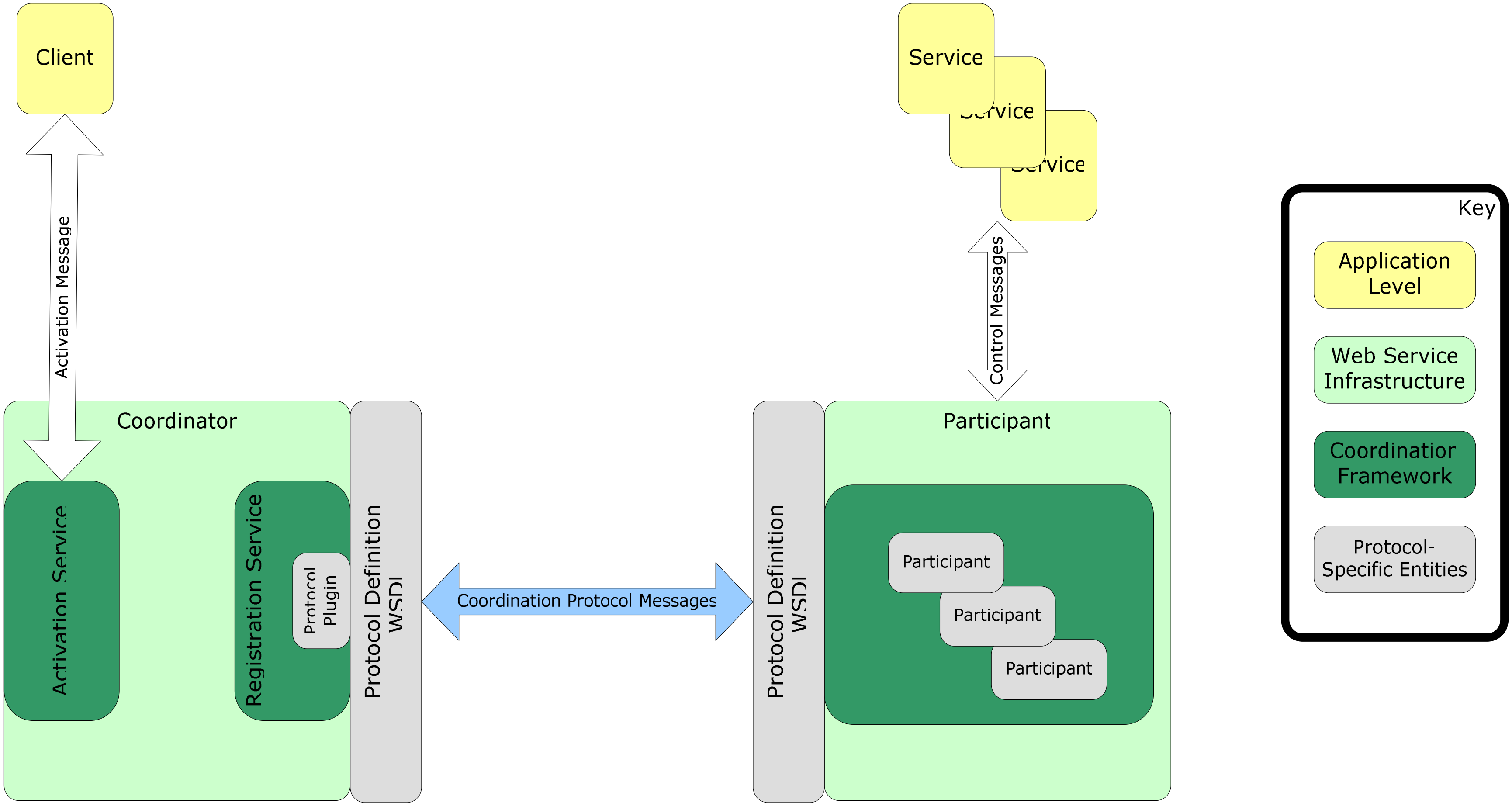
- Web Services transactions will typically glue together J2EE transactions
  - JBoss
  - WebLogic 8.1
  - WebMethods Glue 5.1
- Not application server specific
  - Can run in Apache/Tomcat

WS-C Components

- Activation Service
  - Start a new transaction (coordinator)
  - Return a context
- Registration Service
  - Enlist participant with specified coordinator

Diagram

arjuna
middleware for reliability



WS-T components

arjuna middleware for reliability

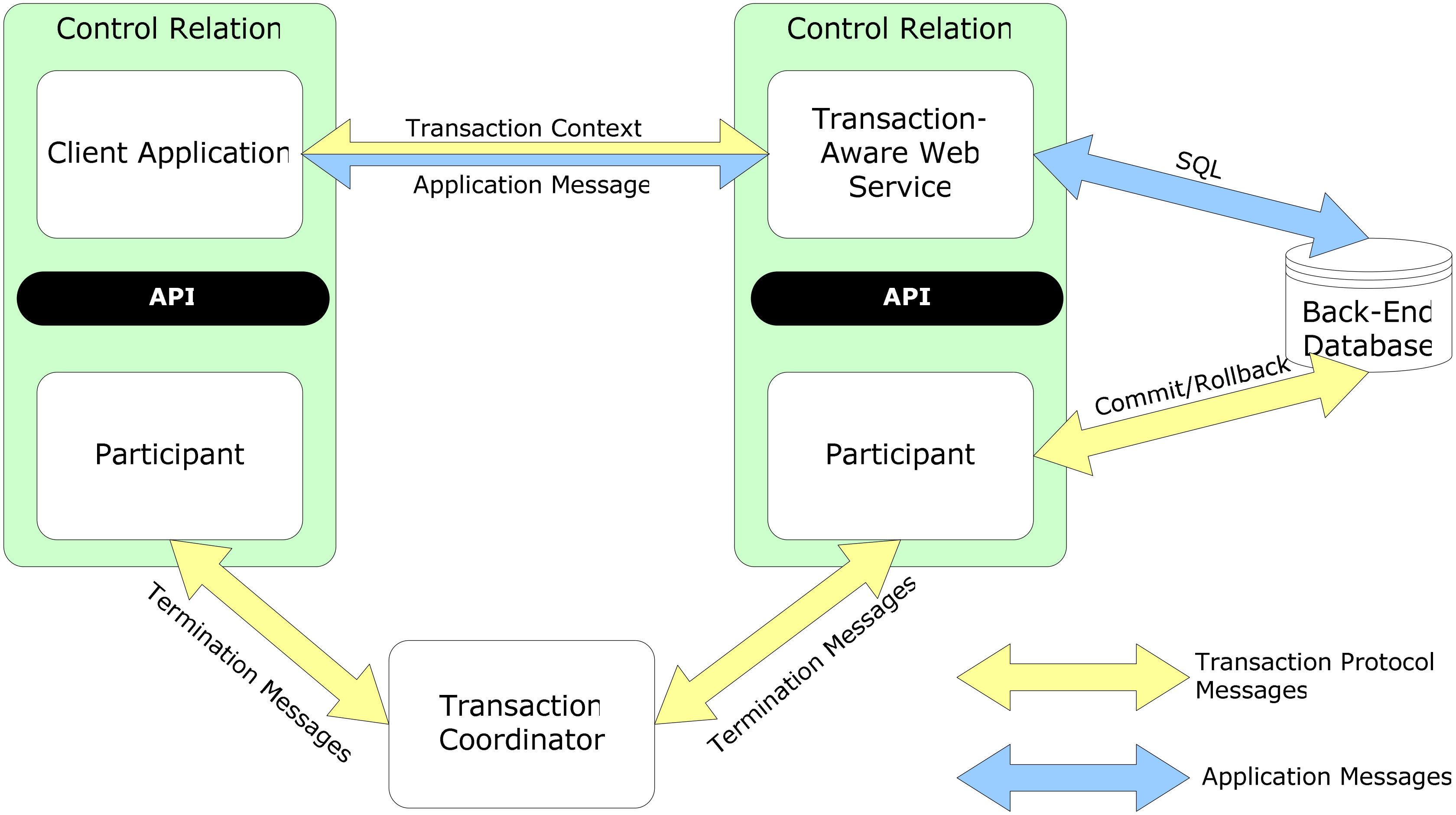
- Coordinator protocols
  - Atomic Transaction
    - Completion, DurableTwoPhase, VolatileTwoPhase
  - Business Activity
    - BusinessAgreementWithCoordinatorCompletion, BusinessAgreementWithParticipantCompletion
  - Similar for WS-TXM
    - Not finalized yet

What do users see?

- API based on JSR 156 (which we lead)
  - No API defined by specifications (obviously)
  - JAXTX (JSR 156) will be the standard
- Support all participant types
- Transparent context propagation
  - Provide deployment specific “interceptors”

Diagram

arjuna
middleware for reliability



Control Relation

Client Application

API

Participant

Control Relation

Transaction-Aware Web Service

API

Participant

Transaction Context

Application Message

SQL

Back-End Database

Commit/Rollback

Termination Messages

Termination Messages

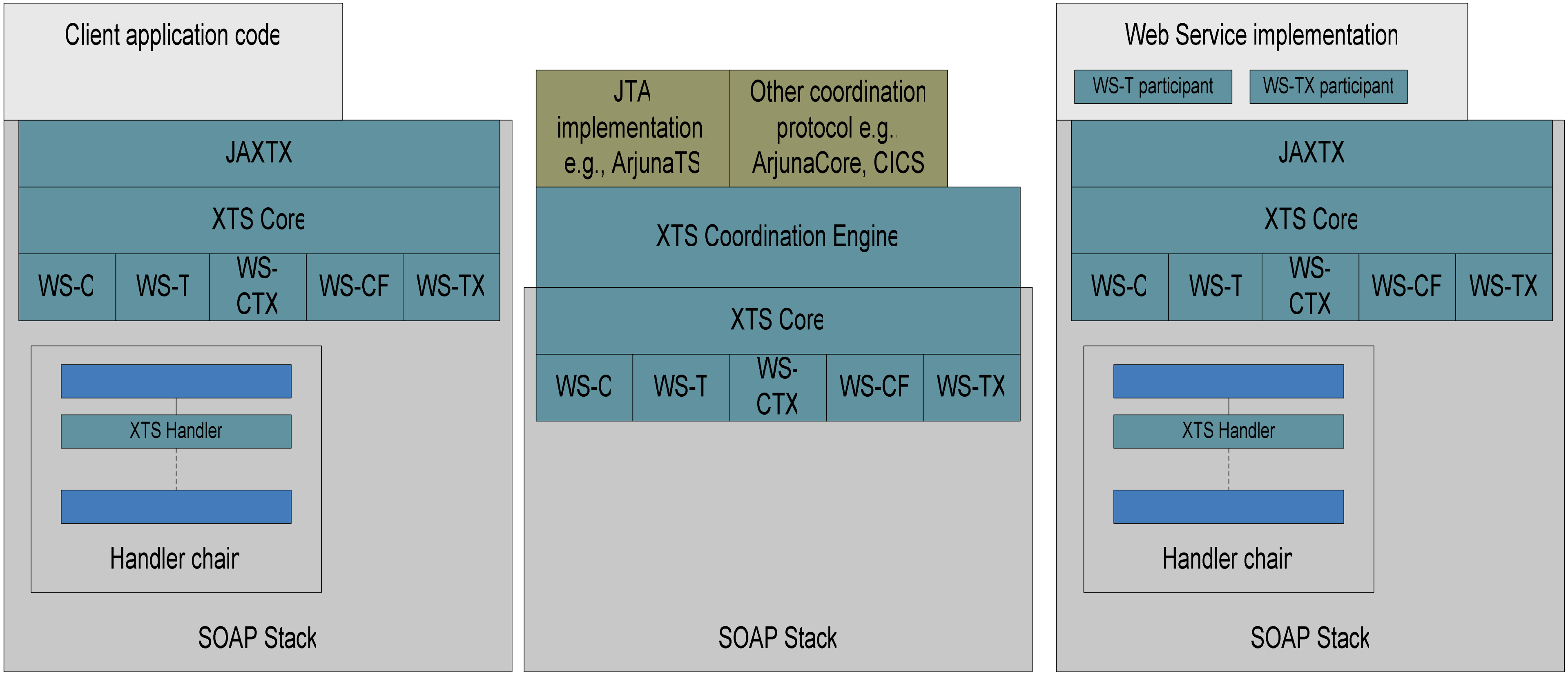
Transaction Coordinator

Transaction Protocol Messages

Application Messages

Overall architecture

arjuna
middleware for reliability



Client application code

JAXTX

XTS Core

WS-C | WS-T | WS-CTX | WS-CF | WS-TX

XTS Handler

Handler chain

SOAP Stack

JTA implementation e.g., ArjunaTS

Other coordination protocol e.g.. ArjunaCore, CICS

XTS Coordination Engine

XTS Core

WS-C | WS-T | WS-CTX | WS-CF | WS-TX

SOAP Stack

Web Service implementation

WS-T participant

WS-TX participant

JAXTX

XTS Core

WS-C | WS-T | WS-CTX | WS-CF | WS-TX

XTS Handler

Handler chain

SOAP Stack

Future

arjuna
middleware for reliability

- Support whatever comes out of the WS-CAF versus WS-C/T standards
- Core infrastructure will remain the same
  - Transactions are transactions, irrespective of the messaging framework
- Architected to be messaging neutral so we will support WS-CAF/C/T amalgam first