Trns•port Test Suite Overview

Presented to the
Trns•port Users Group IT TAG

November 5, 2002

Important Disclosure

- I shamelessly plagiarized a lot of this from previous Info Tech presentations
  - September 2001 Strategic Presentation to Trns•port Task Force (TTF)
  - December 2001 Test Suite presentation to TTF
- But I got permission first, so I'm still a good guy

Overview

- What's the Trns•port Test Suite?
- Why do we need it?
- What will it do?
- How will it be developed?
- What are the benefits?

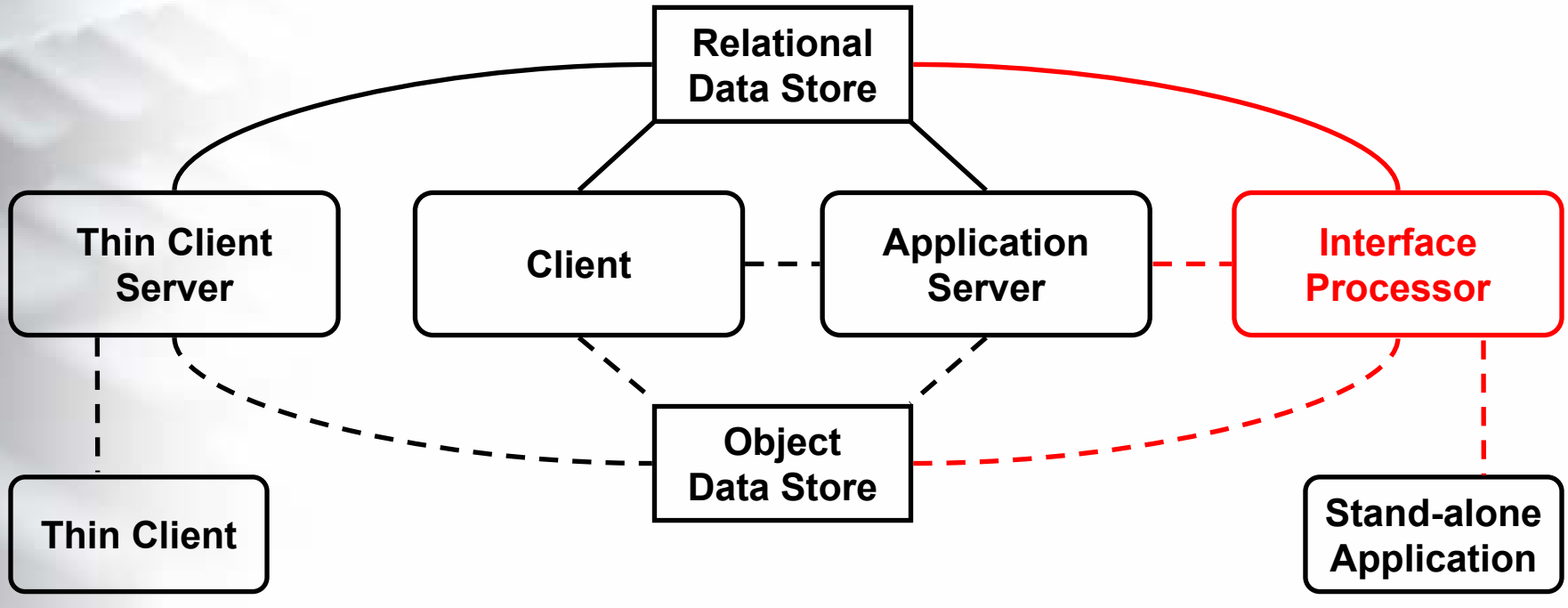
What is the Trns•port Test Suite?

- Basically, three components:
  - A uniform, integrated test plan for all Trns•port modules
  - A comprehensive test database covering the entire Trns•port project lifecycle
  - A set of tools that enable automated testing both during development and at agency sites

Why Do We Need It?

- Trns•port keeps getting bigger, more flexible, more complex architecturally
- Testing is more critical than ever
  - Agency staffs under pressure
  - Costs are climbing quickly
  - Old, successful testing methods aren't as effective any more

Trns•port Logical Architecture



Solid lines show ODBC connections

Dashed lines show HTTP connections

Black – Current **Red – Future**

Lots of Platform Combinations

- Today, just two platform options for each tier means 128 combinations to test
- Some tiers have more than two options
- Multiple supported versions of each Trns•port module compound the complexity

Lots of Modules…

- BAMS/DSS
- CAS
- CES
- Estimator
- Expedite
- FieldManager
- Intranet
- LAS
- PES
- SAPW
- SiteManager
- SitePad
- SiteXchange
- TRACER

…With Different Backgrounds

- CES, PES/LAS/CAS, BAMS/DSS, SiteManager, FieldManager all developed independently
  - By different groups
  - For different user communities
  - To meet different goals
  - Using different methods and technologies, including those for testing

But That's Not All

- Every agency's infrastructure is unique
- Multiple versions of service packs, hotfixes, patches, browsers, ODBC drivers, etc.
- Each agency uses the product differently

The Bottom Line

- Complexity makes testing a challenge
- Existing testing processes evolved for smaller, less integrated systems
- Resources don't exist to manually test every permutation and combination at every stage
  - This is why regression failures and platform-specific problems occur

Improving the Testing Process

- Uniform, integrated test plan
- Automate testing
  - System testing
  - Regression testing
  - Platform testing
  - Installation and update testing at agency sites

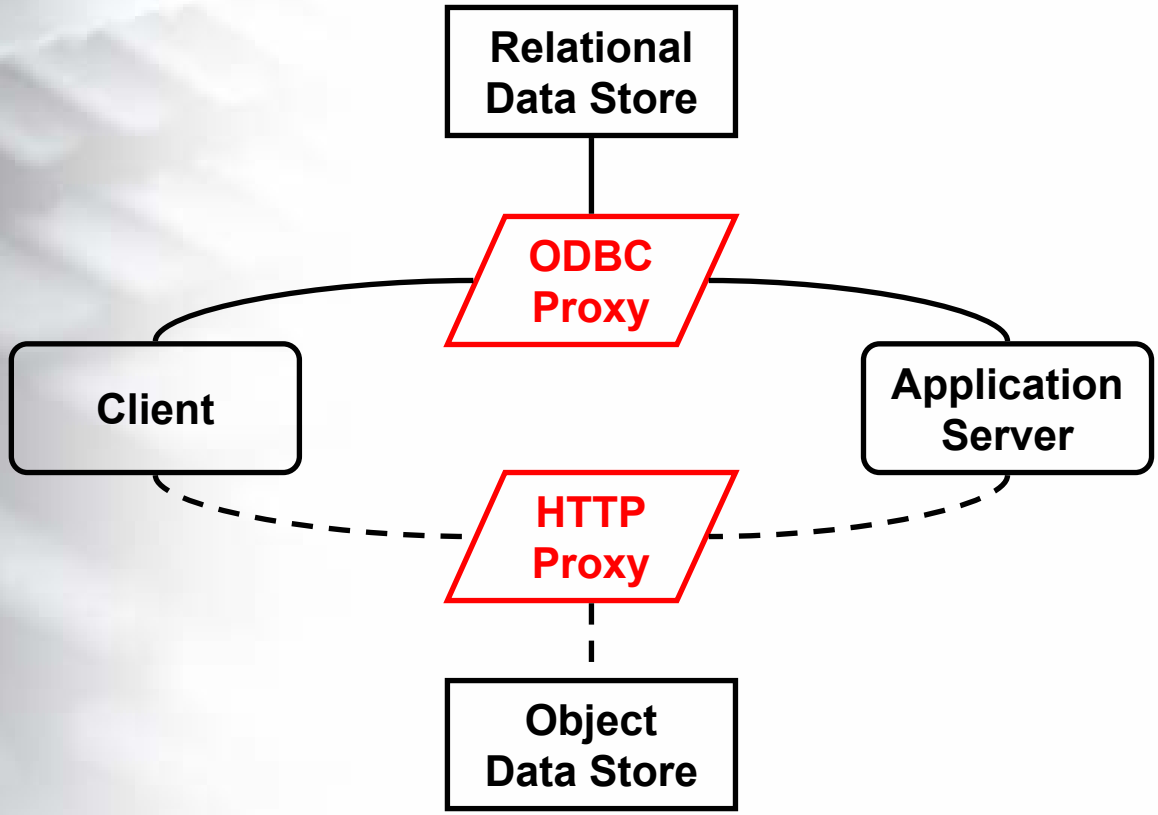
Trns•port Test Suite Components

- Test Plan
  - Test processes
  - Test cases and Test Case Repository
- Test Database
  - Spanning all platforms, all modules
- Test Automation Framework
  - Automated execution of tests
  - Learning and replaying tests

Proxies Make It Possible

- All connections between tiers are either ODBC or (soon) HTTP
  - These can be *proxied*
  - A program can be installed between tiers that forwards all traffic through it
  - The proxies can record every transaction
  - Write programs that use these records to replace either side of the connection

Proxy Logical Architecture



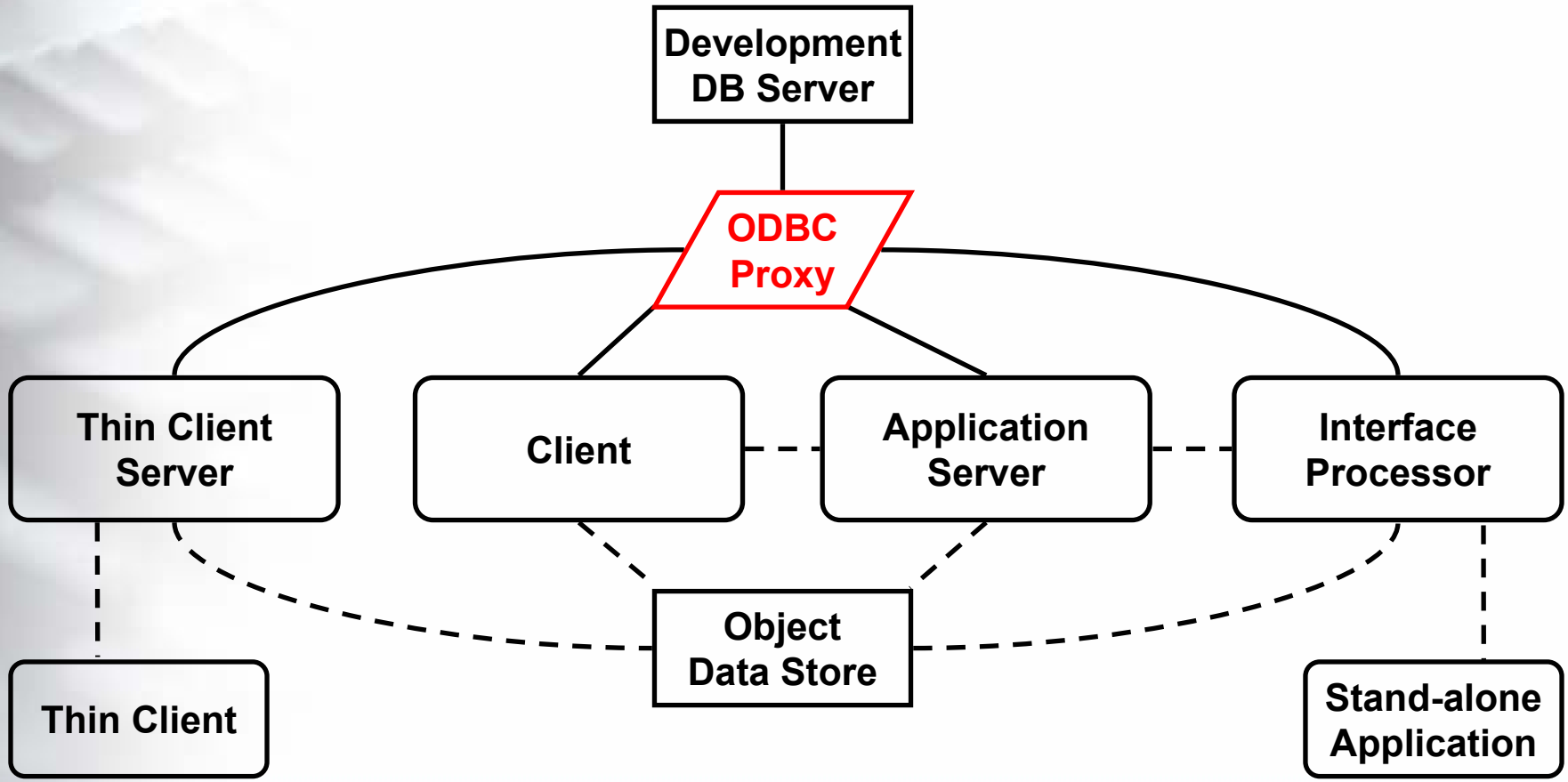
Proxies can be on dedicated PCs or at either end of connection.

Place on server to allow emulation of all clients, or on client to allow emulation of all servers.

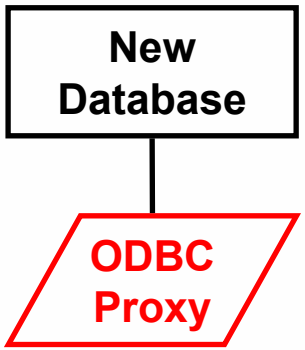
Example: Certify New Database

- Install proxy on existing database server
- Run test script and record results (or use existing record)
- Install proxy on new database server
- Replay all requests
- Compare results to prior ones

Step 1: Run Tests ONCE On Development Environment



Step 2: "Replay" Recorded ODBC Transactions



Only 1 or 2 computers are needed for this step

- Check responses
- If responses match, new database (or version, or ODBC driver) is certified

Compare Efforts

- Full tests (manual or automatic) made only once
- Each new DB is certified automatically
- Same tool works for regression testing of new functions
- New functionality requires adding new tests to existing records (once each)
- Same approach via HTTP works for other types of server tiers

Example: Continual Automated Regression Testing

- Manage all source code in a single repository
  - Developers check in updates when complete and unit tested
- Each weekend
  - Rebuild all tiers
  - Automated regression test all server tiers
  - Report results to developers

Example: DOT Validates New Release

- Install proxy programs in front of each production server
- Record a week's usage
- Install new code on test machines
- Replay the record, check results
- If they pass, update production servers with high degree of confidence

Test Suite Development

- Phased work plan
  - 3 years (maybe)
- Technical Review Team (TRT)
  - Jennifer McAllister, Ohio
  - Tony Compton, Texas
  - Paul Neumann, Colorado

Phase I, 2002 – 2003

- Start with one test contract
  - 5 functions in PES, 5 in LAS
  - Oracle 8i on Windows 2000
- Initial Test Plan
- Initial Test Database
  - Test Database installation, maintenance, and version control processes

Phase I (continued)

- Automated Testing Framework
  - Test Run Framework
  - PES/LAS batch process automation
  - HTTP recording and playback
- Phase I Testing Results Report
- Phase II Work Plan
- Phase I cost: $320,438 (2,831 hours)

Phase II

- Expanded Test Plan
  - Build to 20 contracts and 10 functions per module
  - Add remaining client/server modules, and Intranet
- Expanded Test Database
  - Add remaining certified DBMSs

Phase II (continued)

- Expanded Automated Testing Framework
- Initial onsite test process
  - Agencies will be able to execute the standard Trns•port Test Plan on site

Phase II (continued)

- Research options for ODBC record/playback, binary output validation and GUI test automation
  - Be very careful not to reinvent the wheel
  - But be mindful of cost to agencies
- Phase II Testing Results Report
- Phase III Work Plan

Phase III

- Expanded Test Plan
  - High data volume test cases, e.g., historical pricing
  - Add Estimator, SAPW, SitePad, SiteXchange
- Expanded Test Database
  - Add data representative of 3 years of production use

Phase III (continued)

- Expanded Automated Testing Framework
  - ODBC record/playback
  - Automated validation of binary output
  - GUI test automation
- Onsite test record/playback process

Phase III (continued)

- Phase III Testing Results Report
- Phase IV Work Plan (?)

Things To Stay Aware Of

- Test database requires deep analysis
- GUI automation is essentially a proxy for a human being – hard to do!
- Test data and scripts require careful maintenance
  - Long-term maintenance costs for the Test Suite estimated to be $100,000 per year

Trns•port Test Suite Benefits

- Unified, consistent approach to testing
- Improved cross-module testing
- Reduced costs
  - Maintenance, support, and enhancements
  - Platform testing and certification
  - Agency testing time

Test Suite Benefits (continued)

- Increased quality of software
  - Issues found early in engineering
- Onsite validation at agency sites
  - Using Trns•port Test Plan
  - Using recorded production activity
- Benefits begin to accrue very early

Summary

- The Trns•port Test Suite is the top development priority for the TTF
- Must continue to make Trns•port easier for agencies to use efficiently
  - The product has grown to the point where new testing processes are needed
  - The Trns•port Test Suite will provide these processes

Trns•port Test Suite Overview

Paul Neumann
paul.neumann@dot.state.co.us