FIX/FIXML Implementation

Sam Johnson, David Rhodes
marcusevans professional training, February 2001

Introductions, Quickly

- Who are we, and why are we here?
- What is TransactTools, anyway?

- Who are you?

Agenda

- Another Overview of FIX
  - Really quick, like 10 minutes

- FIXML and What It Really Means
  - 30 minutes
- Implementation: FIX-Enabling Your Business
  - 45 minutes

- Actually Getting Connected
  - 1 hour

- Betting the Business on FIX
  - 30 minutes

What is FIX, really?

- FIX is a peer-to-peer networking protocol with a very narrow focus: wholesale financial transactions.

- FIX was created before companies were all interconnected via the Internet and private networks.

- FIX was built with two fundamental objectives:

    - Reliability and timeliness of communication
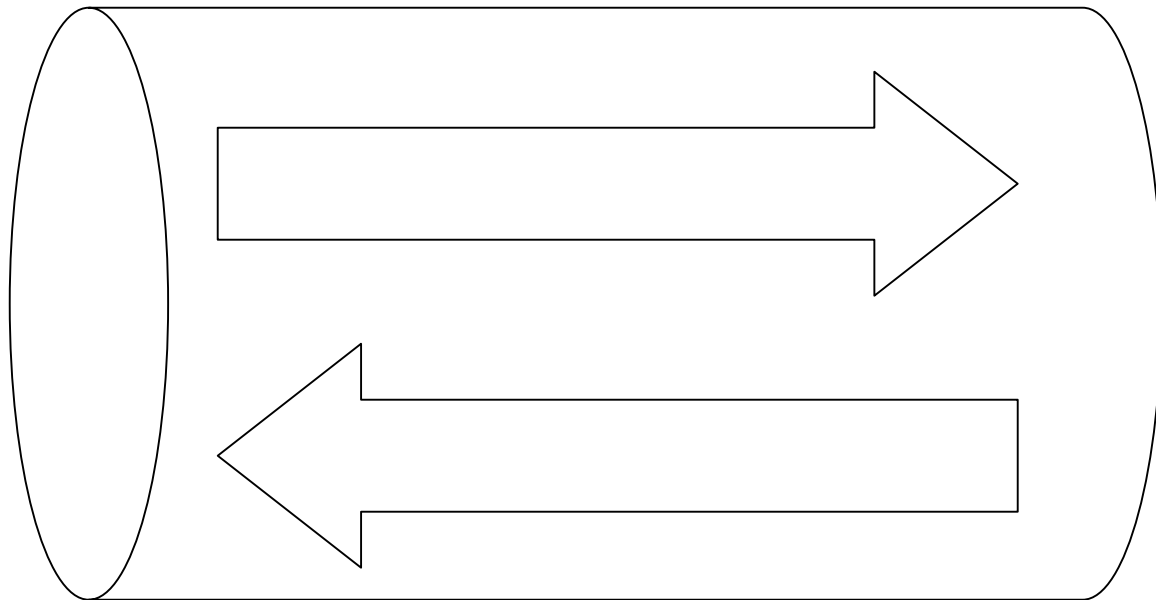   - Flexibility of business content

Overview of FIX (again)
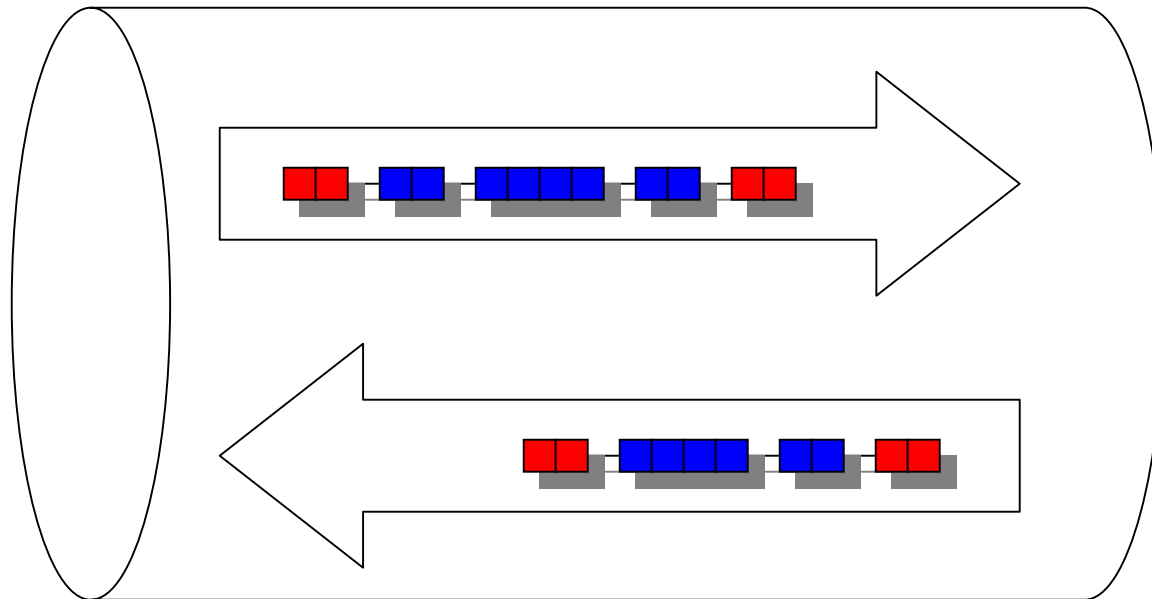but from a Systems Perspective This Time

What is FIX, really?

- First, FIX is a transport-independent session protocol that guarantees reliable real-time delivery of data between two directly-connected points.
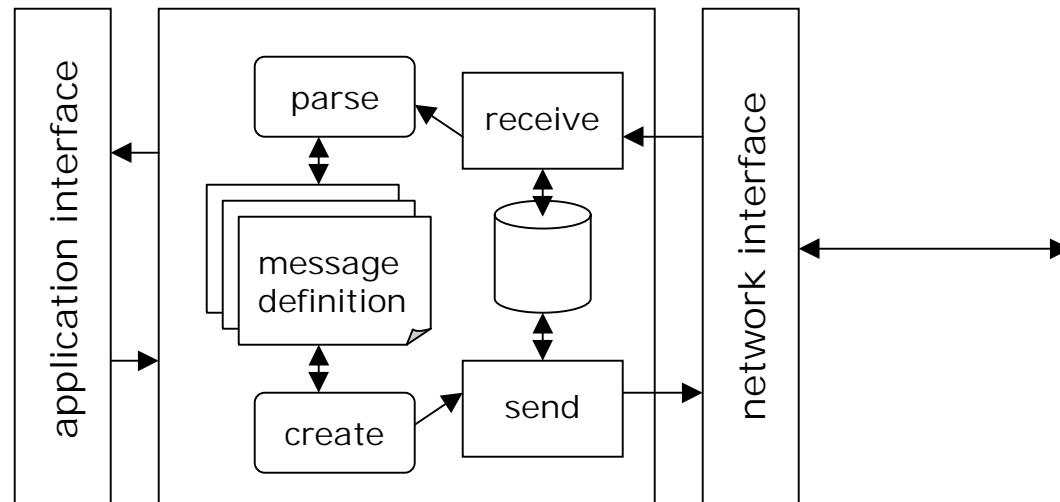
What is FIX, really?

- Second, FIX is a set of flexible and extensible business message formats.

A FIX Engine, Under the Hood

- A FIX engine is simply a piece of software. It maintains a network connection, creates and parses messages, and recovers if something goes wrong.

What's Special about FIX 4.2?

- Number of fields almost doubled from 4.1
- Number of pages more than doubled

- Thankfully, the number of appendices doubled

- 18 more business messages

  – XML envelope (to wrap FIXML data)
  – Converts, Forex, Derivatives, Options, Bonds
  – Exchange support – Market Data, status
  – Japanese trading

So Then What is FIXML, really?

- FIXML is another, more structured way to format the FIX business messages.
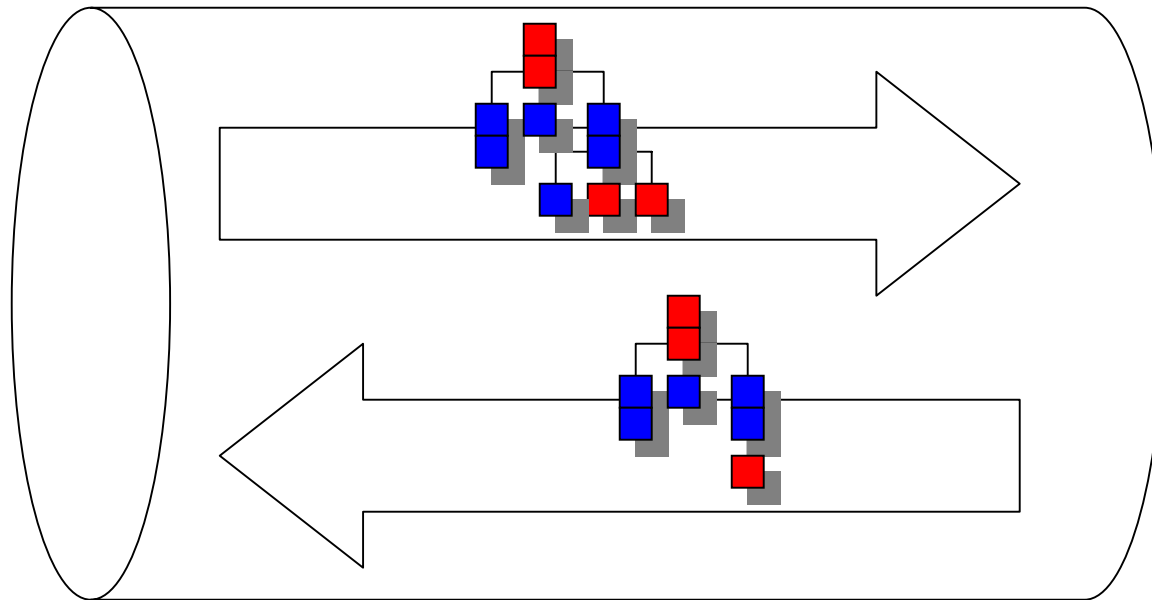
FIXML and What It Really Means in the Grand Scheme of Things

Separation of Session and Business Layers

- With FIXML, the FIX Committee publicly acknowledged a need to think about the FIX session and application layers separately

    – Session can transport messages of any format
    – Application messages can be delivered in ways other than via the FIX Session

- And there's an easy migration path from FIX: the old tag-value format can be used like an envelope for a FIXML message (new fields in 4.2)

Structured Business Messages

- XML introduces structure into the application message. For things like repeating groups of related fields, this is very helpful.

```
<repeating>
 <group>
    <field1></field1>
    <field2></field2>
 </group>
 <group>
    <field1></field1>
    <field2></field2>
 </group>
</repeating>
```

- Strictly speaking, an XML parser can validate that a FIXML message conforms to a DTD in terms of structure only. XML doesn't understand data types.

```
<!ELEMENT StrikePrice (#PCDATA)>
<!ATTLIST StrikePrice
    FIXTag CDATA #FIXED "202"
    DataType CDATA #FIXED "float"
    Min CDATA #FIXED "0"
    Max CDATA #FIXED "99999999.9999"
>
```

DTDs and Validating Parsers

- A Document Type Definition (DTD) describes the conditions necessary for a well-formed XML document:

  - Optional and required elements
  - Structure and grouping of elements
  - Attributes associated with elements

- For example, HTML documents conform to a DTD

- A validating parser (such as a web browser) can use a DTD to check an XML document to make sure that it's correctly constructed.

And, Eventually, XML Schema

- XML doesn't help much with validating data inside a document—it thinks everything is a string

- Schema initiatives aim to provide content validation by defining data types

- It's unclear which, if any, will prevail

  - XML-Data (Microsoft)
  - DDML (Data Definition ML)
  - DCD's (Document Content Definitions)
  - SOX (Schema for Object-oriented XML)

The Downside

- FIXML messages are large-ish

- Implications for performance in high-volume applications aren't well understood
  - Transfer of larger messages
  - Structure (and content) validation at parse-time

- Nobody is doing it yet

Implementation:
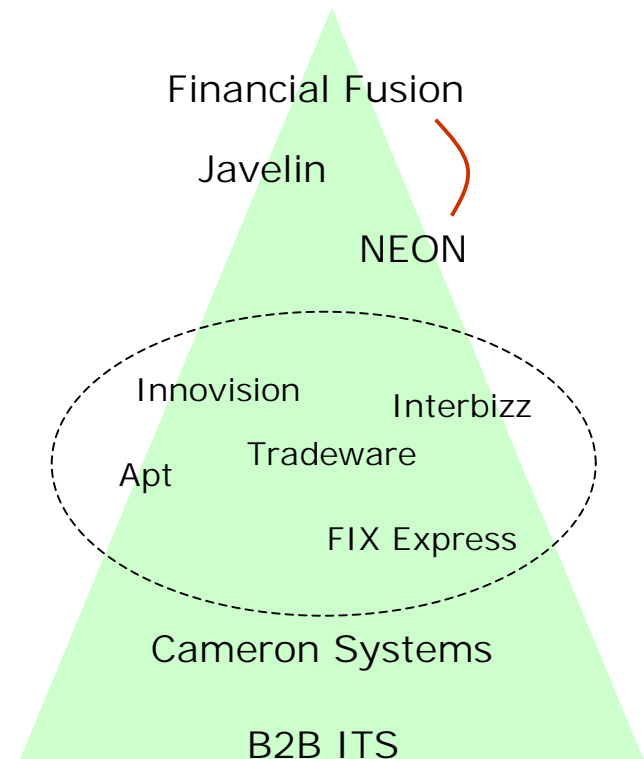# FIX-Enabling Your Business

FIX Engines and FIX Libraries

- A lot of FIX engines turn out really to be FIX libraries

- FIX engines are applications that stand alone and provide an interface to internal applications.

  - Financial Fusion, Javelin

- FIX libraries require that either an interface shell or an application be built around them. They aren't stand-alone applications themselves.

  - Cameron, B2B ITS, and most others

Vendor Solutions

- Pricing: 3 tiers of solution
- Market share: X 2 leaders

- How much can one pay for a FIX engine?

- hmmm… Why?

Financial Fusion

Javelin

NEON

Innovision

Interbizz

Tradeware

Apt

FIX Express

Cameron Systems

B2B ITS

High-Availability FIX Engines

- Several vendors now offer premium "HA" versions of their servers. These are all built pretty much the same
  - As messages are received and sent, they are written to a common persistent store and also propagated among connectors (consistent state)
  - Redundancy at FIX machine and software levels, guaranteeing there's always an entrance to the FIX system

FIX connector (stateless)

shared state*

FIX connector (stateless)

message handler

* may not be persistent
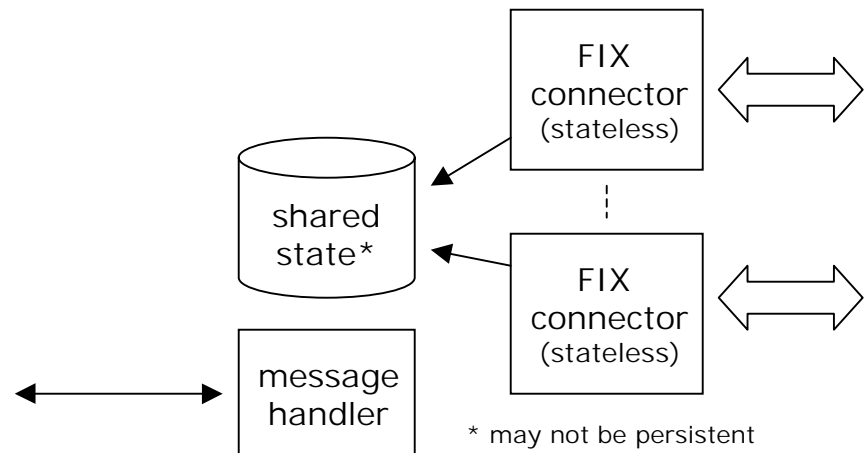
Making the Build vs. Buy Decision

- It used to be about deciding whether you wanted to depend on a vendor product for FIX messaging

| +	-
Saves development cost and time	Little or no control over or access to source code
Vendor responsible for support, enhancements, upgrades	Customer is at the mercy of the vendor

- More and more, as vendors begin to make source code available, it's about deciding whether to reinvent the wheel

+	-
Saves development cost and time	Customer has to learn and support somebody else's code

A Solution that Scales with the Organization

- In many cases, the objective is to FIX-enable an organization rather than a single application.
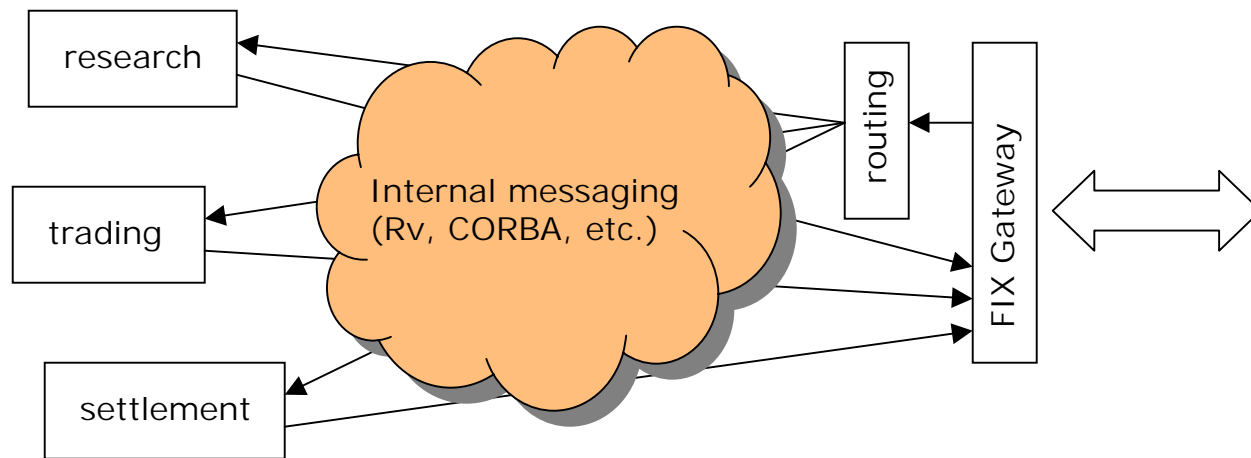
- Typically, the FIX gateway is configured as a router and interfaced with the company's existing messaging

Actually Getting Connected

Differences in Trading Partners Interfaces

- Multiple versions of FIX
  - While many firms have moved to 4.1 or 4.2, the vast majority are still on 4.0
  - If reliant on a third-party order management system or FIX engine, may not be able to move until the vendor does
  - Therefore, may need to support the same business functionality across more than one version of FIX

Differences in Trading Partners Interfaces

- Multiple Configurations
  - Even with a specification, capabilities may differ across firms
  - Often due to different interpretations of the standard, especially in FIX 4.0
    - OrderQty on cancel/replace
      - Some see it as the leaves qty, while some see it as the total order qty
  - Sometimes due to simply not following the spec
    - SendingTime in UTC
      - Some do eastern time or another time zone, making time comparisons difficult
    - OrderQty on order cancels
      - Some send 0, some send the remaining quantity, while spec requires original quantity

Physical Connectivity

- TCP/IP most commonly used transport protocol for FIX
- While testing is often done over the Internet, production configuration is generally over private connections
  - Direct connection
    - Frame relay
    - T-1
    - ISDN (often as a backup)
  - Third-party networks
    - IXnet
    - TNS/MacGregor
    - TradeRoute
  - Virtual private network (VPN)
    - Addresses security, but not performance or reliability issues of the Internet

Physical Connectivity

- Firewall configuration
  - Host firm production servers may be behind firewall, even in private connection configurations
    - Must open access from client hosts or networks to internal hosts or networks
    - Policy decision as to how tight this security should be
  - Client firm may need to open access to specific IP addresses and ports
    - For testing over the Internet
    - For production access, if firewall is between FIX servers and private connection

What About Encryption?

- PGP/DES-MD5 and other current encryption algorithms for FIX are somewhat antiquated
- SSL/TLS (Secure Sockets Layer/Transport Layer Security) is currently being explored as an option for FIX encryption
- SSL/TLS proxying provides a simple way to handle FIX encryption, as it simply fronts existing FIX servers, encrypting messages from and decrypting messages to FIX servers

| FIX Server | ←— FIX —→ | SSL Proxy | ←— SSL —→ | SSL Proxy | ←— FIX —→ | FIX Server |
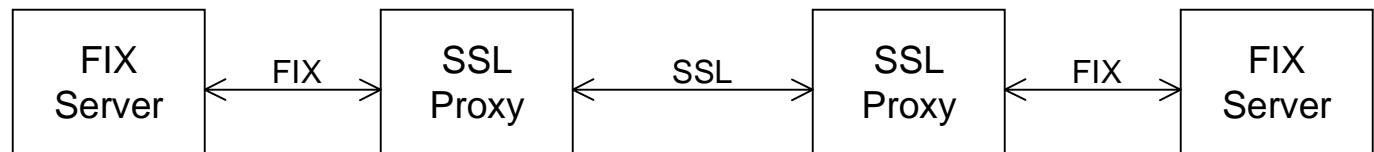
Trading Networks

- Trading networks, such as TradeRoute and GlobalCrossing, offer more than just TCP/IP connectivity
- Hub-and-spoke model reduces some of the complexities of FIX connectivity, but has drawbacks when compared to point-to-point trading links
  - Security. All transactions flowing through one hub increase the chance of being compromised
  - Performance. Hub itself can become a bottleneck
  - Reliability. Hub itself can be a single point of failure
  - Functionality. Reduces the application-level functionality of FIX to a least-common-denominator across participants

Business-Level Compatibility

- Physical connecting customers and understanding their FIX version and configuration is just the beginning
- Must rigorously test all critical functionality
  - Orders
    - Required parameters and allowed values (e.g. Side, HandlInst)
    - Optional parameters and allowed values (e.g. ExecInst, TimeInForce)
    - Optional order types (e.g. Stop, Stop Limit)
  - Cancels
    - Simple
    - After partially filled
    - Partially filled while pending cancel
    - Unsolicited cancels

Business-Level Compatibility

- Changes (cancel/replace)
    - Simple
    - After partially filled
    - Filled while change is pending

Session-level compatibility

- Verify what happens when things get out-of-whack
  - Stop heartbeats on client and host, simulating connectivity problems
  - Send sequence numbers that are too low and see how FIX engines respond
  - Send sequence numbers that are too high and see how the FIX engines recover
  - Create fills "offline" and see how the client FIX engine deals with messages it thinks it missed while not logged on

How do I test all of this?

- Most of this compatibility testing is done manually (!)
  - Resource-intensive. Requires at least two people, one from the client and one from the host company, usually on the phone in front of FIX engines and log files
  - Time-consuming. Creating these scenarios, testing them, and reviewing and communicating the results takes a significant amount of time
  - Error-prone. Since it is a person reviewing the output of the FIX engine, it is not possible to test a large number of scenarios and a number of variables within each scenario without making a mistake or two.

How much testing do I need?

- The more testing you can do, the better.
  - Ultimately saves time and money for both parties
    - Less time is required in production support handling common problems
      - Difficulty logging in again after a lost connection due to problems handling resend requests or gap fills
    - Fewer trades are disputed
      - No more tracking down partial fills that a client's FIX engine missed
      - No more disputing the intent of a cancel/replace on order quantity

Automating the Testing Process: Archipelago

- Archipelago was the first to launch a fully-automated FIX interface certification service consisting of:
  - 7 required session-level tests
  - 7 required orderflow tests
  - 30 optional orderflow tests
  - 4 required cancel tests
  - 15 optional cancel/replace tests

- Archipelago no longer does *any* manual certification testing with trading partners

Production Support and Monitoring

- Successful large-scale point-to-point connectivity requires a great deal of monitoring
  - More than just server and operating system tools
  - Need proactive, rules-based notification of a variety of events that can occur in a high-volume trading environment
    - Connections that have dropped more than X times in some period
    - Cancels or changes that have been pending for more than Y minutes
    - Partial fills send with OrdStatus=6 for customers A, B and C who have had trouble with those in the past

Betting the Business on FIX …?

First of all…

- There are too many egos in this business to ever settle on one protocol. Even if it were the right thing to do.

Can FIX Scale to 1000 Connections?

- It can, but not very easily.

- FIX connectivity requires not only that peers speak the same language, but also that they can have meaningful business conversations
  - Trading partner testing is critical

- FIX was designed as a persistent point-to-point protocol, and doesn't fail-over well
  - Production network monitoring is critical

Let's Revisit the whole Peer-to-Peer Thing

- A lot has happened since FIX was originally created

    – WebMethods:
    proprietary XML-based solutions for stateless business-to-business transactions

    – Gnutella, Freenet, OpenCOLA:
    distributed, real-time content routing among peers

    – Napster:
    centralized directory combined with a peer-to-peer transaction platform

Alternate Session Layers

- The fact that FIX relies on a predetermined, persistent machine-to-machine connection is pretty limiting

- The idea of separating application messages from the underlying transport (introduced with FIXML in 4.2) has prompted users to experiment with alternate transports

  - http: polling messages like web pages
  - smtp: mail-based routing
  - beep: standardized reliable point-to-point layer
    - www.bxxp.org
 - instant messenger: anyone??

What is FLIRT, and can it work?

- Simple Object Access Protocol (SOAP) is an open standard for transporting XML documents over HTTP.

  - This is cool because HTTP is pretty firewall-proof
  - This is not so cool for FIX because HTTP is a one-way protocol and FIX is a two-way protocol
  - Even so, the right software can implement a pretty good two-way session on top of HTTP

- FLIRT is FIXML over HTTP, so technically it's not exactly the same thing as SOAP but really there's no difference.

Lessons from Napster

- Napster uses a dynamic, central directory to facilitate getting peers connected.
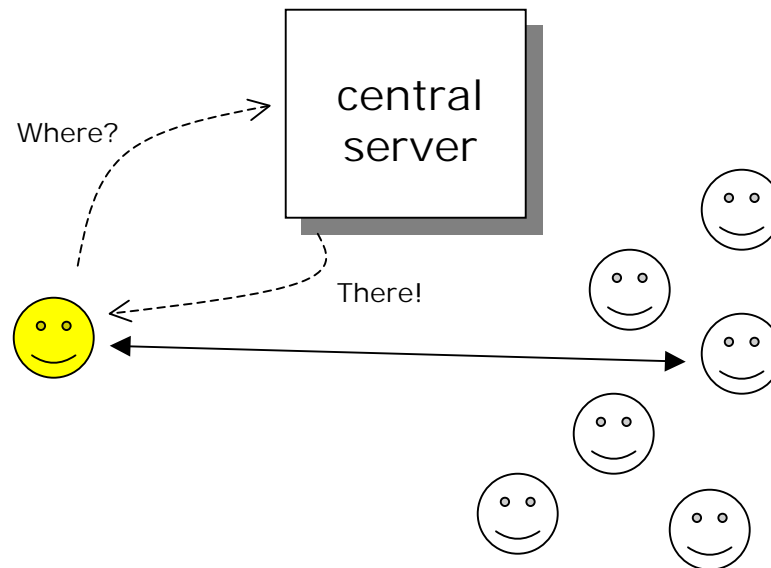
- This is an example of what some p2p luminaries call "distributed enough"

But Whatever Shall We Do In The Mean Time?

- Find a better solution for trading partner interface and capability discovery

- Create better, more automated solutions for peer-to-peer testing, all the way up to the business transaction

- Build peer network monitoring and notification architectures that aren't blind beyond the firewall

Thanks!

transacttools

135 West 29th Street, 9th fl
New York, New York 10001

Sam Johnson
212-244-1343 NY

David Rhodes
708-763-0406 Chicago

www.transacttools.net