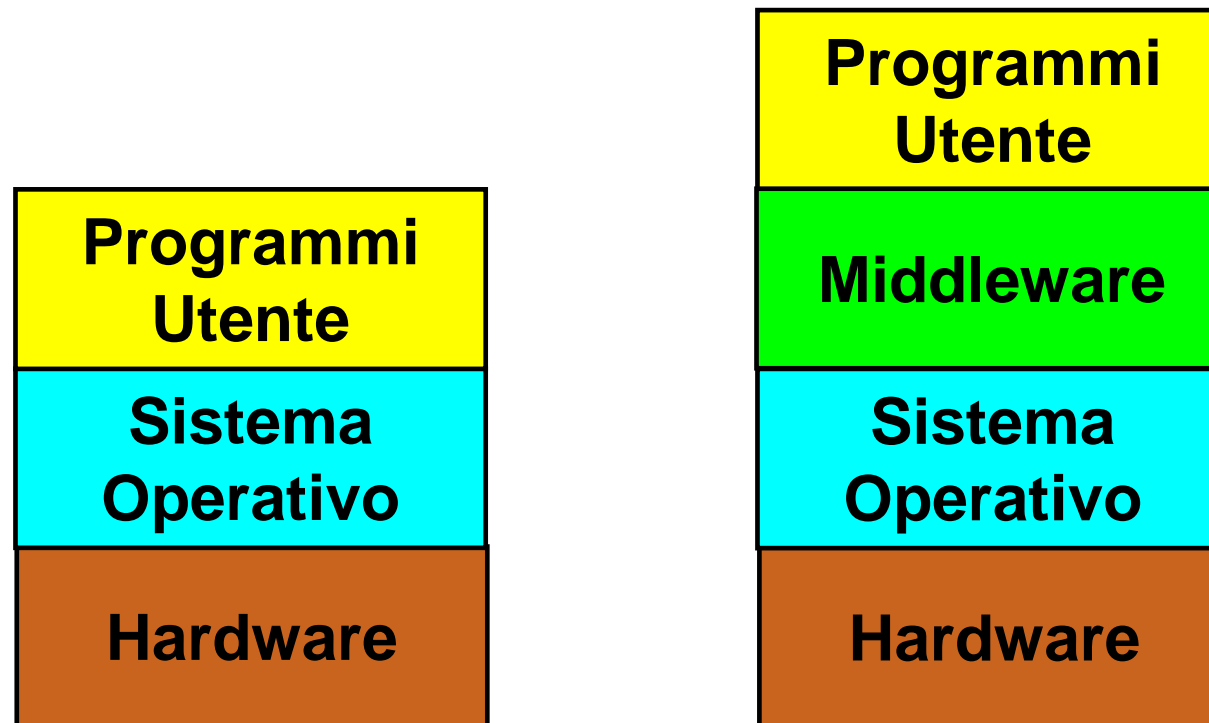




RTI Data Distribution Service Introduzione

Luigi D'Andrea
Progettazione e Sviluppo Software

Cosa è un middleware



Cosa e' DDS?

- Uno **standard** aperto per il network programming.
- Una API che supporta il design pattern Pubblica e Sottoscrivi.

Perché utilizzare DDS?

- Incorpora la metodologia di progetto data-centric.
- Progettato per ottenere alte performance
- Progettato per funzionare su sistemi real-time.
- Riduce i costi
 - Niente fai da te...
 - Costi di integrazione
- Riduce il rischio di essere legato a doppio filo con un vendor specifico.

Publish Subscribe Middleware

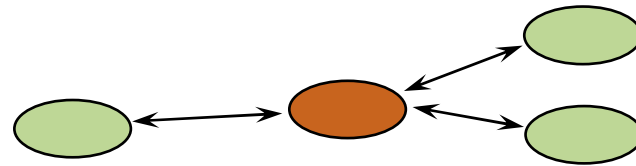
- Il paradigma Pubblica e Sottoscrivi
 - API proprietarie dagli anni '80
 - Prodotti commerciali dagli anni '90, nel settore finanziario-bancario
 - Standard aperti dal 2000: DDS e JMS
- Nelle imprese
 - JMS, IBM MQ, BEA (Oracle) WebLogic, Tibco Rendezvous, RTI DDS
- Nella finanza
 - RTI DDS, Tibco, 29West
- Nei sistemi Real-Time
 - RTI DDS, OpenSplice
- Nei sistemi embedded
 - RTI DDS

Communication Design Patterns



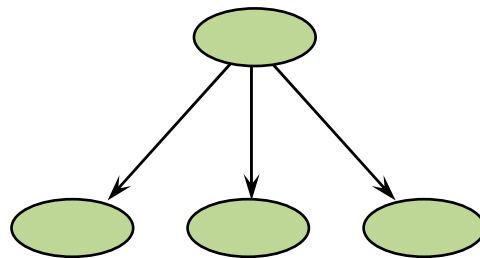
Punto punto

- Telefono
- TCP/serial
- Connection oriented
- Limitatamente scalabile



Client Server

- File systems, Databases, remote method invocation, service oriented
- CORBA, DCOM, RPC
- Connection oriented
- Single point of failure
- Performance bottleneck

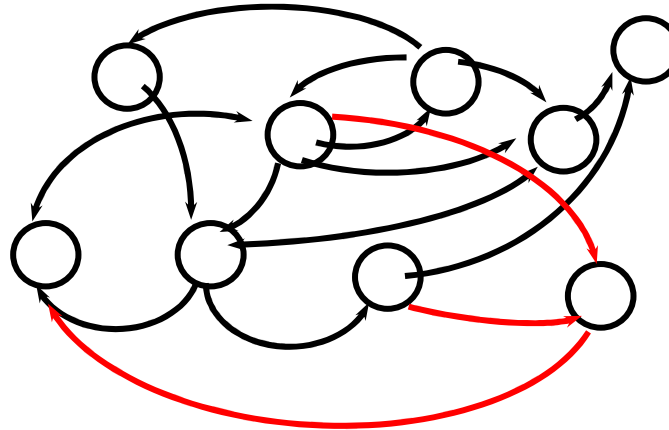


Pubblica e Sottoscrivi

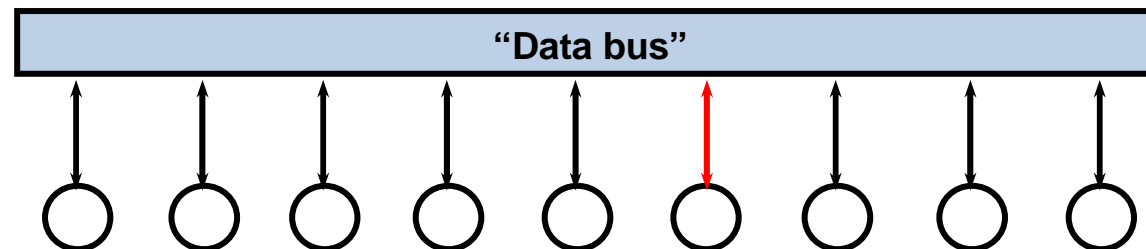
- Riviste, Giornali, TV Broadcasts, data or message oriented
- DDS, JMS
- Connectionless
- Eccelle in comunicazioni di tipo molti a molti.
- Eccelle nella distribuzione di informazioni time critical

Il disaccoppiamento riduce significativamente i costi di sviluppo e mantenimento.

Risultato tipico di comunicazioni di tipo connection-oriented.



La progettazione data-centric semplifica l'integrazione



Data-Centric Design

- **Permette** ai progettisti di lavorare a livello astratto (dati) per definire le relazioni tra componenti (apps).
- **Consente** ai programmatori di creare codice che interagisce attraverso componenti plug and play. Nessun hardcoding di connessioni punto-punto.
- **Autorizza** i sistemisti ad impostare configurazioni a runtime per adattare il sistema a vincoli sopravvenuti.

DDS API Standard

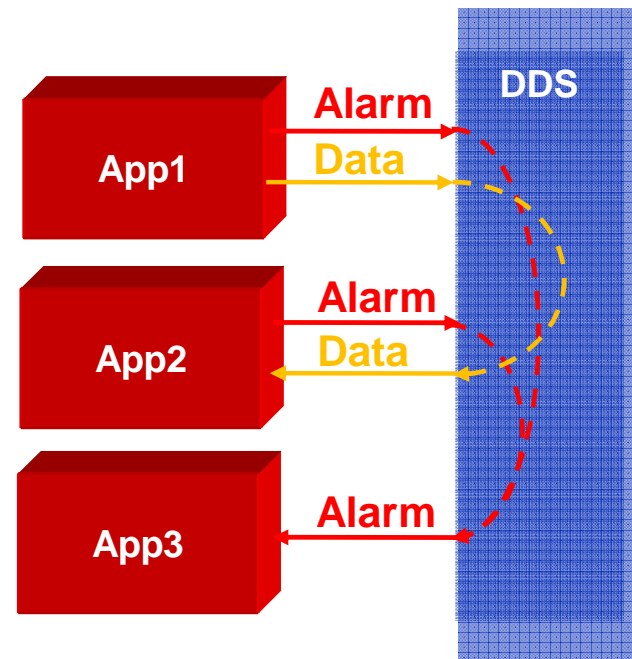
- Data Distribution Service for Real-Time Systems
 - Specifica delle API per la comunicazione Data-Centric Publish-Subscribe per sistemi real-time distribuiti.
 - Adottato per la prima volta nel 2003
 - Attuale livello di revisione 1.2
 - Prossima revisione 1.3
- OMG
 - Organizzazione internazionale per gli standard formata da industrie (end users) e vendor (providers)
 - UML, CORBA, IDL, MDA
- Il ruolo di RTI
 - Membro del consiglio di OMG.
 - Co autore del DDS RFP (Request for Proposal) originale.
 - Co autore delle specifiche DDS
 - Autore della specifica di interoperabilità del DDS
 - Presidente della task force per la Finalizzazione e Revisione del DDS



Come utilizzare DDS?

- Definire la struttura (il data type) dei messaggi che saranno passati
- Dichiarare quale flusso di dati (topics) l'applicazione pubblicherà o riceverà (sottoscriverà)
- Inviare i dati o attendere dati in arrivo
- DDS fa il resto...

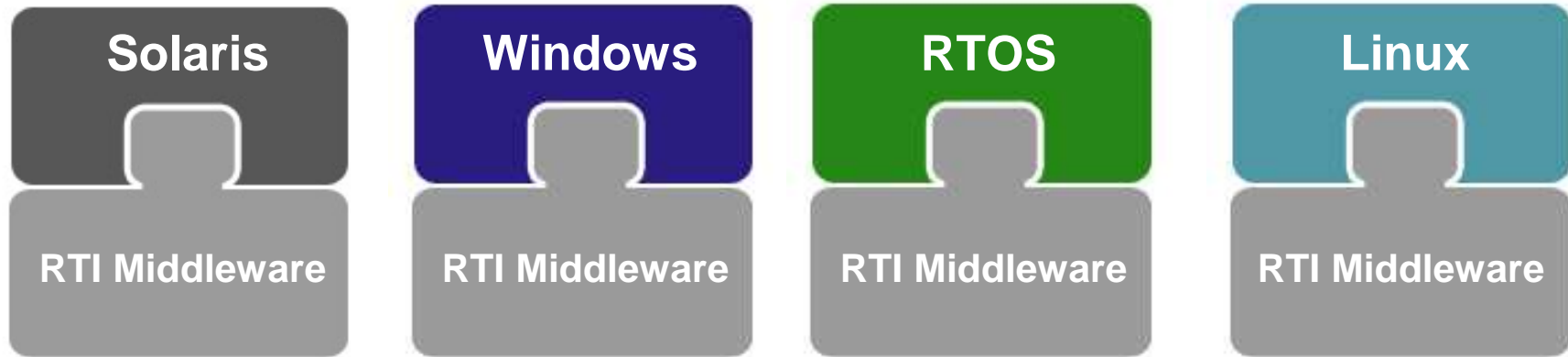
```
struct Event {  
    long id;  
};  
  
struct Buffer {  
    char msg[256];  
};
```



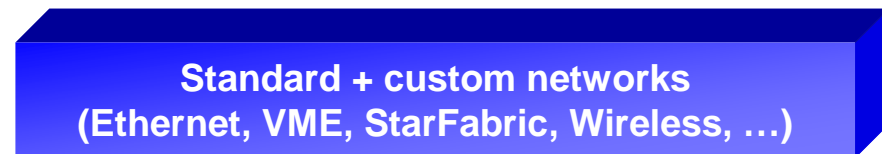
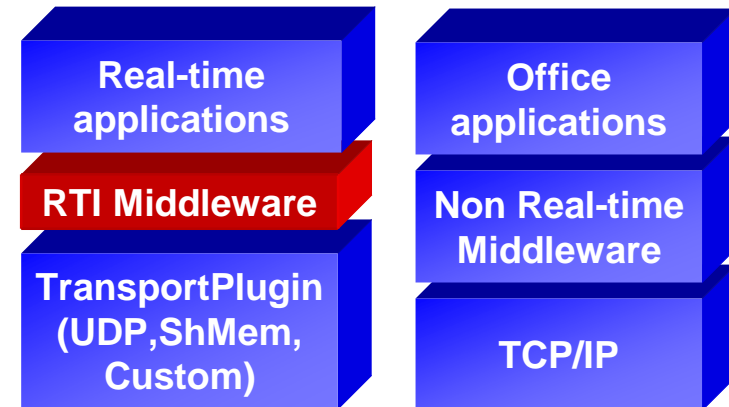
Dettagli...

- La libreria e' Multithreaded, può essere linkata staticamente o dinamicamente con le applicazioni utente.
- Nessun servizio esterno e nessun demone
- I data types utente sono definiti tramite IDL
 - Una utility compila il file IDL
 - Viene generato codice specifico relativo al tipo di dato definito
- Le API sono disponibili in C/C++/C#/Java
- Supporta la maggior parte dei S.O. embedded o per desktop
- Dati/Messaggi possono essere inviati da qualsiasi linguaggio/piattaforma a qualsiasi linguaggio/piattaforma

RTI: Architetture Supportate

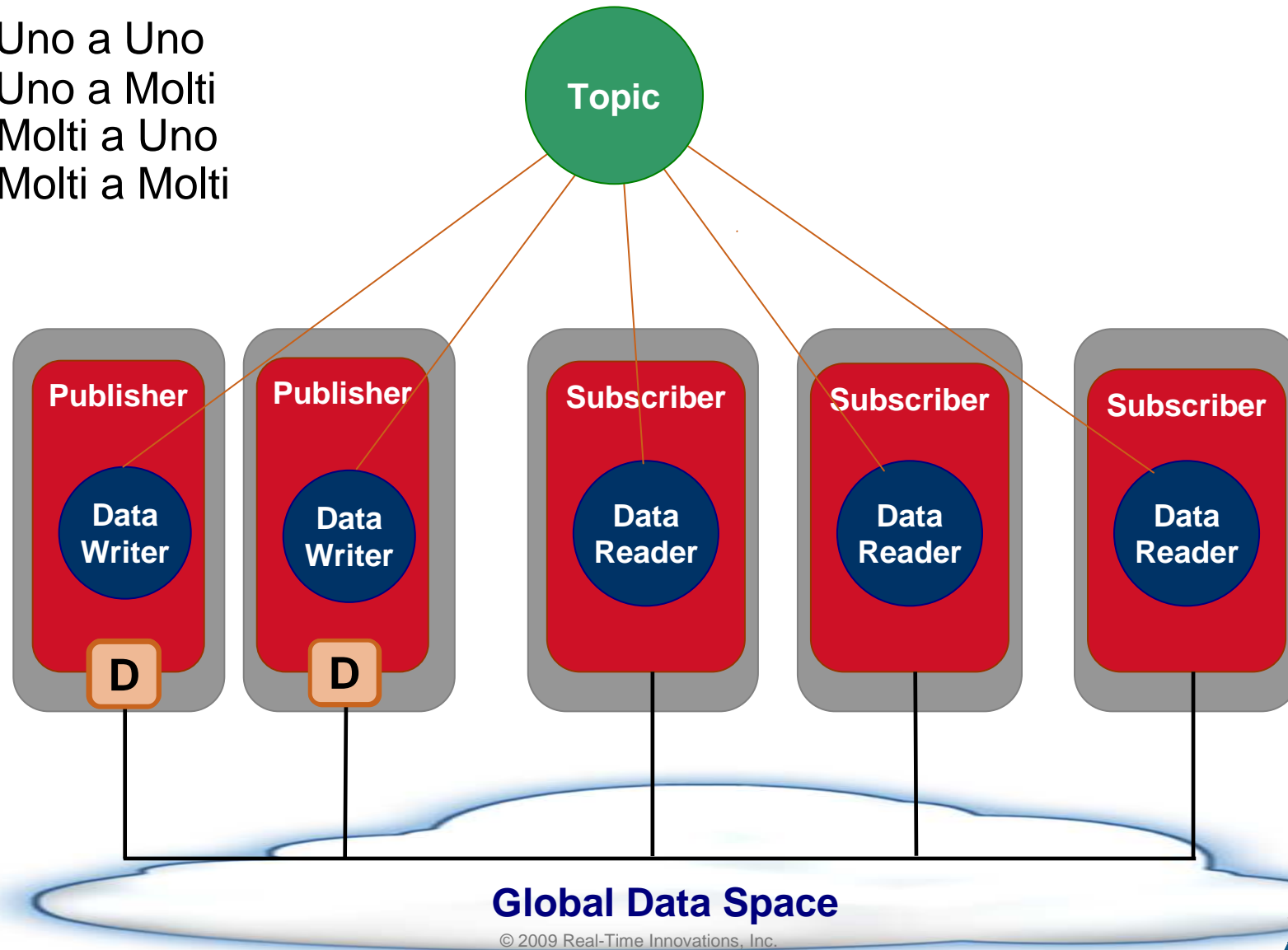


- API identiche per ogni piattaforma
- Indipendente dal linguaggio
C, C++, C#, Java™
- Enterprise and Embedded Support
VxWorks®, INTEGRITY®, LynxOS®
Linux, Solaris, Windows
- **Altre piattaforme su richiesta**



DDS: Scenari Pubblica e Sottoscrivi

Uno a Uno
Uno a Molti
Molti a Uno
Molti a Molti



Scegli una delle seguenti opzioni...

- Oltre ai dati che devono essere inviati, gli utenti, spesso, hanno necessità di specificare come...
 - ...affidabilità (o invia e dimentica)
 - ...quantità di dati (tutto, Ultimi 5, ogni 2 secondi)
 - ...tempo di validità
 - ...permettere molti publisher degli stessi dati
 - ...gestire il failover se un publisher muore
 - ...rilevare le applicazioni “morte”
 - ...gestire interfacce di rete ridondanti
 -
- Nello standard **DDS**, queste opzioni sono controllate tramite la definizione formale di:
Quality of Services (QOS)

DDS Quality of Service

- La definizione formale mette DDS ad una certa distanza dagli altri middleware
- **Controllo dinamico per la compatibilità dei settaggi**
- Oltre 20 politiche definite nello standard
 - **Deadline**
 - **Destination Order**
 - **Durability**
 - **Durability Service**
 - **Entity Factory**
 - **Group Data**
 - **History**
 - **Latency Budget**
 - **Lifespan**
 - **Liveliness**
 - **Ownership**
 - **Ownership Strength**
 - **Presentation**
 - **Partition**
 - **Reader Data Lifecycle**
 - **Reliability**
 - **Resource Limits**
 - **Time-based Filter**
 - **Topic Data**
 - **User Data**
 - **Writer Data Lifecycle**
 - **Writer Resource Limits**

DDS Quality of Service (cont'd)

- QOSs aggiuntivi forniti da RTI per dare agli utenti il controllo dei parametri operativi interni
 - **Asynchronous Publisher**
 - **Batch**
 - **Database**
 - **Default Unicast**
 - **Discovery**
 - **Discovery Config**
 - **Event**
 - **Exclusive Area**
 - **Multicast**
 - **Participant Name**
 - **Profile**
 - **Property**
 - **Protocol**
 - **Publish Mode**
 - **Reader Resource Limits**
 - **Receiver Pool**
 - **Transport Builtin**
 - **Transport Selection**
 - **Type Support**
 - **Unicast**
 - **Wire Protocol**
- Configurabile tramite
 - **File XML** a runtime
 - Codice attraverso le relative API

Invio dei dati

- Sincrono
 - I dati sono inviati dal thread utente tramite una write()
- Asincrono
 - La write() copia i dati in una coda di invio e ritorna immediatamente il controllo al thread utente.
 - Il thread RTI invia i dati
 - **Batched mode**
 - Raggruppa più write() in una singola e efficiente send
 - **Large Data**
 - Per dati più grandi della MTU, DDS opererà una frammentazione e un seguente riassettaggio
 - **Flow control**
 - Può essere definito dall'utente per dare forma al traffico di rete in uscita, (per esempio forzare il limite di banda)

Ricezione dei dati

3 modi di ricevere i dati

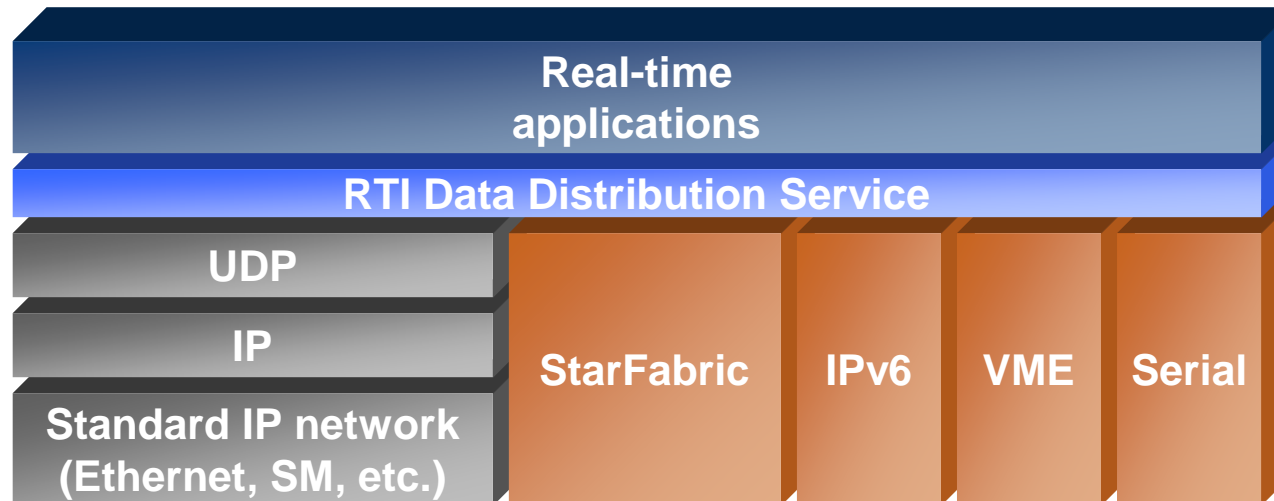
- Tramite **Callback** in un listener del codice utente
 - Il thread interno di RTI chiama il codice utente al momento della ricezione dei dati
- Il thread utente **waits** l'arrivo dei dati
 - Il thread interno di RTI mette i dati in arrivo in una coda e sblocca il thread utente.
- Il thread utente può richiedere, a **polling**, in qualsiasi momento, i dati.

Reliable Data Delivery

- L'affidabilità e' **opzionale** e **configurabile**
 - L'utente deve trovare il giusto compromesso di risorse e performance rispetto all'affidabilità o al best effort.
- Il Reliability protocol è **indipendente** dal trasporto
 - L'affidabilità e' mantenuta in caso di perdita di dati durante il trasporto
 - Provata con trasporto che ha alta latenza (secondi) e alta perdita in trasmissione (più 20%), per esempio trasmissione da satellite o altri link wireless
 - I parametri sono "tunabili" dall'utente tramite il Qos del protocollo
- Multicast affidabile per il flusso di dati uno a molti

Pluggable Transport Framework

- Supporta trasporto non IP centrico
- Supporta l'uso simultaneo di diversi trasporti
- Fornisce alte performance (zero-copy interface)
- I trasporti disponibili includono: UDPv4, UDPv6, Shmem, WAN/Secure
- Trasporti personalizzati per VME, RS232, TCP/IP, StarFabric, native InfiniBand





RTI Tools and Integrated Solutions

**Sviluppo, Integrazione
Test e Debug**

RTI Tools and Integrated Solutions

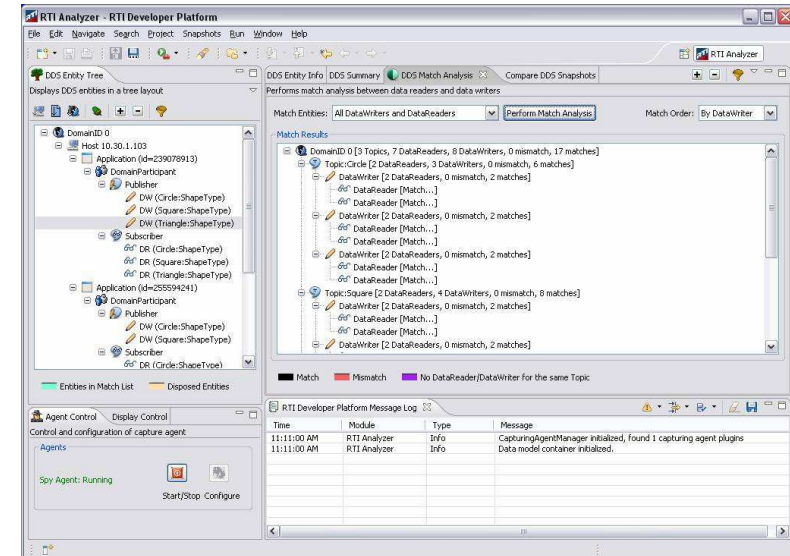
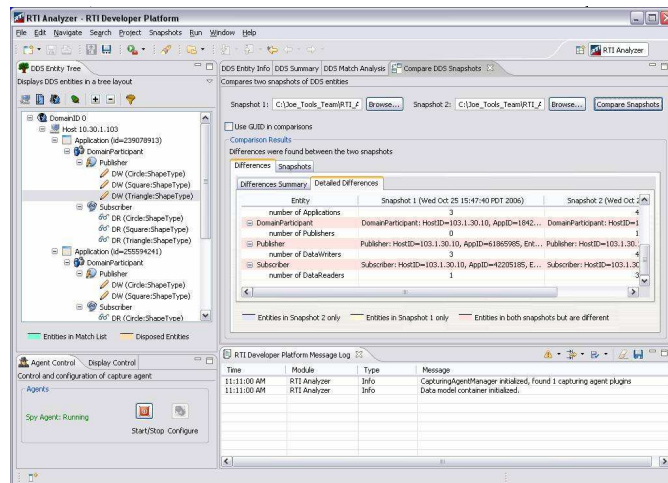
- RTI Developer Platform
 - RTI Analyzer
 - RTI Scope
 - RTI Protocol Analyzer
 - RTI Recorder
 - RTI Persistence Service
 - RTI Real-Time Connect
 - RTI Event Processing
 - RTI Excel Plugin

 - Third-Party Design Tools
 - Sparx Systems Enterprise Architect
 - IBM Telelogic Rhapsody*
 - Artisan Studio*
- * Sembra che abbiano fatto l'integrazione.....

RTI Analyzer

Accesso a tutti gli oggetti da una singola workstation

- Scopre chi parla con chi
- Mostra i parametri degli oggetti
- Prova diverse proprietà, senza cambiare il codice
- Scopre cosa c'è di sbagliato



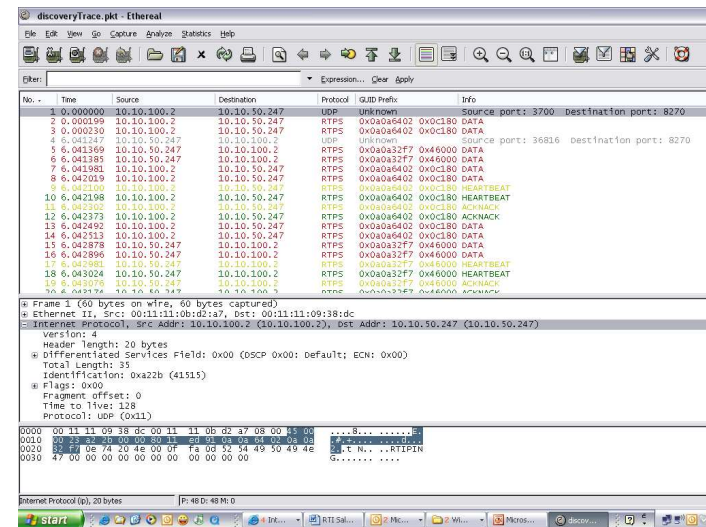
Fornisce differenti prospettive

- Visualizza i nodi per eseguire il debug dell'applicativo
- Visualizza i topic per eseguire il debug dell'applicativo

Da un senso al vostro sistema distribuito

RTI Protocol Analyzer

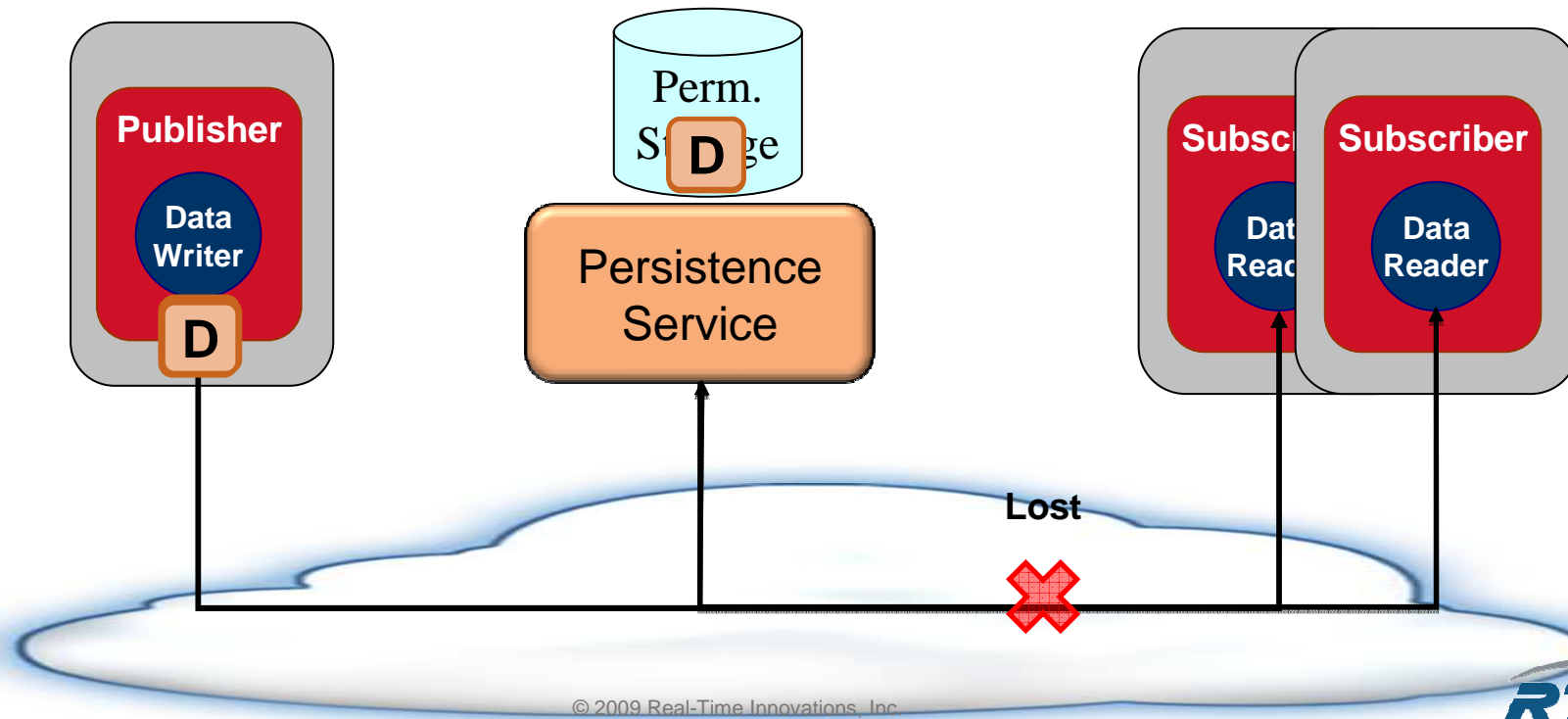
- Analizzatore di traffico e pacchetti
- Cattura i pacchetti RTPS
- Mostra il contenuto dei pacchetti
- Conta i tipi di pacchetti e produce grafici e diagrammi
- In grado di filtrare più di cento protocolli noti.



**Usato per analizzare il traffico sulla rete e
aggiustare i parametri per migliorare le
prestazioni**

RTI Persistence Service

- RTI Persistence Service: implementa il DDS Persistent Profile
 - Registra i dati pubblicati e li fornisce ai subscribers quando necessario.
- Fault Tolerance Benefits
 - Assicura la disponibilità di dati affidabili anche se il publisher fallisce
 - Aggiorna i nuovi subscribers senza caricare i writers



Plugin Excel per RTI

- Permette la sottoscrizione tramite un foglio elettronico di Microsoft Excel...
- Tramite MS Excel e' possibile mostrare, rappresentare grafici e applicare formule sui dati che cambiano dinamicamente.
- E' anche possibile pubblicare dati dal foglio elettronico.



RTI all'opera

Come usare RTI...

- Definire i data type
- Generare il codice di supporto al tipo
- Usare le API DDS per creare le applicazioni che inviano e ricevono dati
- Compilare le applicazioni “linkando” le librerie in maniera statica o dinamica
- Configurare il QoS tramite i file XML (può anche essere fatto tramite le API)
- Lanciare le applicazioni – non c’è bisogno di nessun servizio esterno!!!

Tipi di dati

- Definire i data type (tramite file IDL, XML, XSD)
 - Molti vantaggi nell'uso di messaggi strutturati
 - E' possibile utilizzare anche tipi predefiniti per messaggi non strutturati (stringhe o buffer di byte)...
Nuova funzionalità presente dalla versione 4.4.b
- Type-code
 - Auto generato, definizione codificata del data type (campi, tipo dei campi, nome del file)
 - Inviato sulla rete durante la fase di discovery
 - Utilizzato dai tool o dalle applicazioni utente per interpretare i dati senza mai dover compilare il data type

Generare il Type-Support code

- **rtiddsgen** – utility in \$NDDSHOME/scripts
 - Accetta file di ingresso nei seguenti formati:IDL, XML, XSD
 - Genera il codice di supporto per il data type definito in input
 - Specifica del linguaggio di out: C, C++, Java, C#
 - E' possibile specificare example per una piattaforma
 - Genererà un esempio pub/sub e l'infrastruttura di generazione per la piattaforma specificata (es. Makefile)
- **Codice Type-Support**
 - Da compilare e “linkare” con l'applicazione
 - Usato da RTI per serializzare deserializzare allocare liberare strutture dati di tipi definiti dall'utente

Configurazione del QoS

- Può essere configurato a codice tramite API
- Meglio tramite file XML caricato a runtime
- E' possibile definire insiemi di QoS come profili
- I profili possono essere definiti gerarchicamente (inheritance)
- E' possibile legare i profili ai nomi dei topic
- Avere file XSD che possono essere caricati dagli editor XML per semplificare la scrittura dei profili QoS.

Eseguire le applicazioni DDS

- Le applicazioni che usano RTI devono conoscersi tra di loro prima che i dati vengano scambiati
- Il Discovery e' un processo automatico
 - Di default e' usato il multicast, nessuna configurazione e' necessaria. Su reti complesse può non funzionare correttamente
- E' possibile usare `NDDS_DISCOVERY_PEERS`
 - Variabile di ambiente, file, o impostata a codice tramite API
 - Ad es. `NDDS_DISCOVERY_PEERS = host1,host2,host3,192.168.0.1`

Materiale

- Tutto il materiale e gli esempi saranno disponibili da Lunedì 29 Marzo sul sito www.xappsoftware.com all'indirizzo:
 - www.xappsoftware.com/dds/

Q & A

Domande?