OpenSplice™: “A superior Information-Backbone for mission-critical systems”

September-2006
Hans van ‘t Hag, DDS product Manager
hans.vanthag@prismtech.com
(1) PrismTech at a glance
(2) OpenSplice Background
(3) OpenSplice by Example
(4) OpenSplice Tooling
(5) OpenSplice Performance
(6) OpenSplice Availability
PrismTech at a glance: A global player...

**HQ Sites**
- Gateshead, UK
- Burlington, MA, USA

**Engineering Centers**
- Gateshead, UK
- Saddle Brook, NJ, USA
- Berlin, Germany
- Paris, France
- **Hengelo, Holland**

**Field Offices/Distributors**
- Frankfurt, Germany
- London, UK
- Helsinki, Finland
- Washington, DC, USA
- Houston, TX, USA
- Fort Wayne, IN, USA
- Memphis, TN, USA
- Seoul, South Korea
- Beijing, China

**Product Lines:**
- OpenFusion™ CORBA
- **OpenSplice™ DDS**
- Xtradyne™ Security
- Spectra™ MDD for SDR
PrismTech at a glance: *Fortune 500 Global Customer Base* ...
Interoperable Middleware for all architectural requirements

<table>
<thead>
<tr>
<th>Enterprise</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Source</td>
<td>High Performance COTS</td>
</tr>
<tr>
<td>Low footprint</td>
<td></td>
</tr>
</tbody>
</table>

- CORBA DDS
- RT CORBA DDS
- RTE CORBA & DDS

Equipment

- Near Real-Time Information Processing
  Land, Air and Naval Command & Control, Tactical info. management

- Real-Time Data Processing
  Weapon control, Simulations …

- Device Control
- Signal Processing
- Data Processing

- TAO C++
- JacORB
- RTorb Java
- RTor Ada
- OpenSplice
- JO J2ME
- e*ORB SDR C++
- e*ORB SDR C
- ICO
(1) PrismTech at a glance
(2) OpenSplice Background
(3) OpenSplice by Example
(4) OpenSplice Tooling
(5) OpenSplice Performance
(6) OpenSplice Availability
Thales and PrismTech: “from Proven-Proprietary to Open-Opportunity…”

THALES and PrismTech sign strategic alliance for Mission Systems’ Software Infrastructure, Middleware, Tools and Related Services

PrismTech (USA) will become a Preferred System Integrator for Thales

Press, France and Reston, VA, USA – 18 July, 2006 – Today, THALES and PrismTech jointly announce a strategic alliance that will see PrismTech independently develop and commercially exploit THALES’ Data Distribution Service (DDS) middleware technologies. THALES in turn accesses the trimming competitiveness of their mission-critical systems through PrismTech packaging, enhancing, and supporting them in a commercial-off-the-shelf (COTS) product line. This new DDS product line is being launched by PrismTech and will be marketed under PrismTech’s OpenSplice™ brand name.

For many years THALES had been developing its own proprietary middleware solutions for mission-critical systems but, since the end of the 1990s, THALES has been following an open architecture strategy and has been heavily involved in the development of open standards. A significant part of this strategy was the active role of THALES in the standardization of the DDS specification at the Object Management Group (OMG™) and the implementation of a DDS-compliant product, SPELICE-DDS, which leveraged more than thirteen years of publish-subscribe middleware deployment experience in Combat Management and Air Traffic Control Systems. THALES’ business alliance with PrismTech is seen by both companies as a logical continuation of this strategy which enables OpenSplice™ in the most complex and proven DDS COTS implementations available.

Dominique VERNAI, VP Research and Technologies of THALES, said: “Innovation and technical leadership are key differentiators in all our markets, Defense, Aerospace and Security. THALES has established an innovation eco-system, including academic laboratories, universities and industries, where high-tech DDS play a key role both as research partners and as technology providers. The agreement between THALES and PrismTech exemplifies this strategy. It will ensure sustained R&D collaborations in the area of open middleware technologies and the dissemination in the larger market of a software technology initially developed by THALES for a limited business area. Our main objective in this agreement with PrismTech is to ensure that the SPELICE DDS technology remains the benchmark, fully-compliant DDS COTS, to sustain our open systems architecture strategy.”

Keith STEELE, CEO of PrismTech, added: “Partnership is the keyword in the relationship announced with Thales today. This is a true win-win outcome, not only for PrismTech and Thales, but also for the wider market. PrismTech believes that we can fully exploit and further develop best in class field-proven technology in our core middleware business and bring it to market in a very timely manner. Thales is able to guarantee that an innovative technology investment vital to their business but not seen as a ‘core’ functional area go on to become a bottom-line DDS product, guaranteeing ongoing development and support in line with Thales’s often extended program requirements. Thales also benefits from a close ongoing commercial and technical relationship in a key technology area alongside a partner with whom they share common goals and objectives, particularly in respect to product quality and shared commitment to open standards. The wider market clearly benefits by being able to now access high-quality DDS COTS software that would otherwise have remained ‘internal’ to the Thales organization.”

Thomas Cléchaire, Technologies and Product Director of the THALES Software Architecture Center of Excellence and who welcomed the announcement, “SPELICE middleware has been enhanced in THALES’ systems and provides both the needed quality of service for mission-critical systems and full compliance with open standards. This partnership enables Thales to access the full power of DDS technology for its future systems. This is an opportunity to bring the DDS technology to the market with the business and technical benefits of DDS.”


OpenSplice v2.0 delivers much more than traditional publish-subscribe messaging middleware by providing a true Real-time Information Backbone ensuring the Right Information is Available in the Right Place at the Right Time

OMG Realtime & Embedded Systems Workshop, Arlington, VA, USA – 11 July, 2006 – PrismTech™, a leading provider of productivity tools and middleware, today announced its next generation Data Distribution Service (DDS) software, OpenSplice™ v2.0. Offering much more than traditional publish-subscribe messaging middleware, OpenSplice v2.0 provides a true real-time ‘Information Backbone’ – ensuring the right information is available in the right place at the right time.

To ensure operational advantage, mission-critical systems increasingly require distributed, real-time information availability for applications ranging from small-scale embedded control systems to large-scale enterprise information management systems. Real-time information from multiple sensors in these systems must be delivered in a reliable and timely manner to multiple software components, which combine and process this information in order to either present it to human operators or use it to control multiple distributed actuators on time and in a safe manner.

Furthermore, demanding fault-tolerance and other Quality of Service (QoS) requirements are also pushing developers to look beyond traditional middleware technologies (e.g. non-real-time publish-subscribe or service-oriented client-server) for acceptable solutions. These requirements are increasingly seen in application domains such as combat management and surveillance systems for military services, air traffic control systems in civil aviation and surveillance control and data acquisition (SCADA) systems in industrial control.

OpenSplice v2.0 is now able to provide a commercial-off-the-shelf (COTS) solution for these demanding real-time net-centric systems meaning that systems providers no longer have to design, build and support in-house proprietary middleware.

Fully supporting the Object Management Group’s (OMG’s) Data Distribution Service (DDS) specification, spanning from 30+ man years of mission-critical systems development experience, and based on proven software deployed in over 15 armed services worldwide, OpenSplice v2.0...
FIELD PROVEN CHARACTERISTICS

Many different customers: 15 Navies world-wide
Many different ships/missions: 22 Ships classes (from FPB’s up to Destroyers)
Large-scale & mission-critical: 150 CPU’s, 2200 applications, 4.000 tracks/sec
Real-time and Fault-tolerant: usec. accuracy, no single-point-of-failure
**ARCHITECTURE TRENDS: DDS, A Major Specification**

*in Line with the Network Centric Warfare Paradigm*

---

**DDS mandated for US - OACE**
- DDS is key for success in NAVY OA
- SPLICE extensively evaluated

“**SPLICE provides good performance and scalability as a publish-subscribe middleware for combat system applications**”

(NSWC-DD, OA Technical assessment 07/2004)

---

**DARPA “ARMS” study**

**Key Trend**
- DoD system requirements are increasingly more dynamic, diverse, & demanding

**Problems**
- Existing architectures
- Existing COTS
- Existing multiple technology bases
- brittle & configured statically
- too big, slow, buggy, incapable, & inflexible
- proprietary & limit effectiveness by impeding
- Assurability of DoD,
- Adaptability & Affordability

---

Today, each system brings its own: networks computers displays software people

**Consequences**
- Hard to meet required performance levels
- Hard to control distributed resources
- High software lifecycle costs
- e.g., many “accidental complexities” & low-level platform dependencies

---

“The finalization and availability of the DDS specification really is a tremendous achievement that addresses a significant need in both government and civilian sectors”

(Dr. Richard Soley OMG Chairman, Consumer Electronics, September 13, 2004)

---

**Recognized potential**
- DARPA recognizes DDS importance
- Dynamic Resource Management potential
OpenSplice™: full Compliance to OMG DDS Spec.

**DDS ‘drivers’**

- **OMG/CORBA**
  - IDL for data-definition
  - Object orientation

- **SPLICE**
  - Content awareness
  - Information Management

- **NDDS/SPLICE**
  - pub/sub messaging
  - real-time networking

**OpenSplice Compliance-profiles**

- **Object-Model**
  - DLRL

- **Persistence**
  - DCPS

- **Content-Subscription**
  - DCPS

- **Ownership**
  - DCPS

- **Minimum-Profile**
  - DCPS

**OpenSplice v2.1**

- **V2.2**
  - DLRL Module
  - Persistence Module
  - Cont. Sub. Module
  - Core Module
DDS Features

Object Oriented information view
- Local object model extending the distributed DCPS data model
- Manages relationships and supports native language constructs

Distributed QoS-driven information management
- Fault tolerance and global persistence of selected data
- Guaranteed data availability supports application fault-tolerance
- Content-aware filtering and dynamic queries:
  - reduces application complexity
  - improves system performance

Real-time publish/subscribe messaging:
- Asynchronous ‘one-to-many’ real-time data communication
- Dynamic data flow based on ‘current interest’ (pub/sub)
- Platform independent data model (IDL)
- Strongly typed interfaces for multiple languages
- Information Ownership management for replicated publishers
OpenSplice™ by Example: “building a mini CMS, …”

**SENSOR PROCESS**
- Optical sensor
- Scans the environment
- Produces ‘Tracks’
- Position of ‘objects’
- Reports ‘**pointTrack**’

**CLASSIFICATION PROCESS**
- Classifies tracks
- Determines their identity
- Analyses the trajectories
- Determines hostility
- Reports ‘**trackState**’

**DISPLAY PROCESS**
- Displays track info
- Both position & identity
- Raises alerts
- Requires ‘**pointTrack**’
- Requires ‘**trackState**’
OpenSplice™ by Example: “the Minimum Profile …”

**Sensor**
- **PointTrack Publisher**

**Display**
- **PointTrack Subscriber**

**PointTrack Topics**
- **PointTrack**
  - `long trackId;
  - Position pos;
  
  - **Key**: trackId
  - **QoS**: best-effort, volatile

**OMG-DDS Real-time Information Backbone**

**Characteristics**
- Basic publish/subscribe data distribution
- Topics (types) specified in IDL
- QoS regarding: reliability, urgency, priority, etc.

**Features / Advantages**
- Autonomous & loosely coupled applications
- Pub/Sub & QoS driven communication
- Strong-typed interfaces
- Smart networking based on priority & latency budget
OpenSplice™ by Example: “The Ownership Profile …”

Sensor-1
Publisher-1
Strength=2

Sensor-2
Publisher-2
Strength=1

Display
PointTrack Subscriber

OMG-DDS Real-time Information Backbone

Characteristics
- Replicated publishers of data (with own ‘strength’)
- Only highest-strength will be received
- On failure, next highest-strength will ‘take-over’

Features / Advantages
- Fault-tolerance by replication
- Notes:
  - Requires a lot of resources
  - Quality must be expressible as an ‘integer’

Notes:
- Requires a lot of resources
- Quality must be expressible as an ‘integer’
OpenSplice Advantages: Core Module (min. & ownership profiles)

- **Functional**
  - Full OMG-DDS minimum profile support
  - Full OMG-DDS ownership profile support

- **Non-Functional**
  - Low footprint: Pluggable-services architecture
  - Excellent scalability: nodal (shared information, shared code), global ('network scheduler')
  - Maximised efficiency: automatic 'packing' based upon Latency-Budget QoS ('urgency')
  - Maximised determinism: priority-based networking based upon Transport-Priority QoS ('importance')

- **Tooling**
  - 1 IDL pre-processor for C, C++, Java code generation (types & named readers/writers)
  - XML-based service configuration: networking-setup, priority bands, network Partitions, etc...
  - OpenSplice Tuner to monitor & control any DDS entity (QoS-settings, reader/writer generation, ..)
  - MDD productivity suite (see later..)
Replicated durability service for maximal fault-tolerance

Case-2

Built-in persistence for non-volatile data
- State preservation for transient publishers
- Settings persistence surviving system downtime
- Replicated durability service for maximal fault-tolerance

Features / Advantages
- **Case-1**: late-joining of Display process
  - Previously produced TrackStates readily available
- **Case-2**: restart of failed Classification process
  - Internal state (already classified tracks) regained
OpenSplice Advantages: *Persistence Module*

**Functional**
- Full OMG-DDS durability support
- Pluggable ‘durability’ service
- TRANSIENT and PERSISTENT information
- Automatic alignment of replicated services
- Integration of COTS databases (v2.3)

**Non-Functional**
- Fault-tolerant availability of non-volatile data (maintained by distributed durability-services)
- Fault-tolerant state (TRANSIENT) and settings (PERSISTENT) information availability

**Tooling**
- XLM configuration-editor (setup durability-parameters i.e. ‘what partitions to maintain where’)
- OpenSplice Tuner allowing to log/replay information and maintain persistent data-sets
- MDD productivity suite (see later..)
OpenSplice™ by Example: “The Content Subscription Profile …”

**Characteristics**
- Adds ‘content awareness’
  - Content-filtered Topics & query-conditions
  - Supports ‘compound interest’
    - Multi-topics (combine/filter/re-arrange topics)

**Features / Advantages**
- Reduced application complexity
  - SQL-like querying and reconstitution of related data
- Improved system performance
  - Only receive/process what is of interest
OpenSplice Advantages: Content Subscription Module

**Functional**
- Full OMG-DDS content awareness
- Content-filtered topics and query-conditions (on ‘data’ as well as ‘lifecycle’ !)

**Non-Functional**
- High-performance ‘information backbone’ behaviour by ‘built-in’ in-memory DBMS
- Reduced application complexity (SQL-based filters and queries)
- Improved system performance (applications don’t need to process unwanted information)

**Tooling**
- OpenSplice Tuner: examine content-filtered topics and query-conditions
- Create readers/writers on the fly including filter- and partition-expressions
- MDD productivity suite (see later ..)
OpenSplice™ by Example: “The DLRL Profile …”

**Characteristics**
- **Local** Object Oriented Data-Access Layer
- Supports ‘OO’ features:
  - Inheritance, aggregation, composition
- Uses DCPS to distribute **state** by ‘mapped topics’

**Features / Advantages**
- Ease of Management of (related) data
  - Object oriented ‘graphs of objects’ (value-types)
- Supports ‘native language constructs’ (i.e. navigation)
  - Automatic ‘change-management’ of objects
Functional
- DLRL supported by OpenSplice (v2.2)
- Navigable data-objects with encapsulated information and user-defined accessor methods
- Extensive selection and fine-grained listener mechanisms ease application design

Non-Functional
- More intuitive information access (for OO-programmers)
- High-performance/low-overhead due to DLRL-support by DCPS-kernel in-memory OO-database

Tooling
- DLRL generators
- Graphical DLRL object modeling (MDD productivity suite)
  - 2-way support: generate topic-model from an existing object-model and vice-versa
(1) PrismTech at a glance
(2) OpenSplice Background
(3) OpenSplice by Example
(4) OpenSplice Tooling
(5) OpenSplice Performance
(6) OpenSplice Availability
OpenSplice™ v2.1 Advantages: OpenSplice Tuner

**Functional**
- **Design** stage: deploy the information model even without applications
- **Development** stage: inject test-data by creation of readers/writers ‘on the fly’
- **Deployment** stage: inspection of reader/writer caches, QoS and performance metrics
- **Maintenance** stage: log and/or inject datasets (both volatile and/or persistent)

**Non-Functional**
- 100% Java application, direct or remote connection to any OpenSplice™ system using SOAP™
- Dynamic discovery of all DDS entities (participants, subscribers, publishers, readers, writers)
- Finetune QoS parameters
- Roundtrip-engineering (SpliceTuner as MDD eclipse-plugin: v2.2)
OpenSplice Productivity Tool Suite: Modelers ‘sneak-peak’
OpenSplice™ Advantages: MDD tool-suite (v2.2)

- Complete modeling of system design cycle
  - Information/application/deployment Modeling
  - Context aware guidance / well defined steps
  - Fast, intuitive, correct

- Information Modeling
  - Graphical system-wide information + QoS modeling
  - DDS code-generation of topics and typed readers/writers
  - Documented packages of re-usable topic-sets

- Application Modeling
  - Information-model (parts) import
  - Graphical application modeling
  - Code-generation from patterns (listener/waitset/MVC)

- Deployment Modeling
  - Graphical modeling of OpenSplice-configuration
  - Service configuration (networking, durability)
  - Runtime control (& round-trip engineering) by OpenSplice Tuner
(1) PrismTech at a glance
(2) OpenSplice Background
(3) OpenSplice by Example
(4) OpenSplice Tooling
(5) OpenSplice Performance
(6) OpenSplice Availability
### Portability Challenges

<table>
<thead>
<tr>
<th></th>
<th>DDS1</th>
<th>DDS2</th>
<th>DDS3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DomainParticipant Factory</strong></td>
<td>compliant</td>
<td>compliant</td>
<td>proprietary</td>
</tr>
<tr>
<td><strong>Register Data Types</strong></td>
<td>static method</td>
<td>member method</td>
<td>member method</td>
</tr>
<tr>
<td><strong>Spec Operations</strong></td>
<td>extra argument (newer spec)</td>
<td>compliant</td>
<td>compliant</td>
</tr>
<tr>
<td><strong>Key Declaration</strong></td>
<td>//@key</td>
<td>single #pragma</td>
<td>pair of #pragma</td>
</tr>
<tr>
<td><strong>Required App. IDs</strong></td>
<td>publisher &amp; subscriber</td>
<td>none</td>
<td>publisher</td>
</tr>
<tr>
<td><strong>Required App. Transport Config</strong></td>
<td>code-based</td>
<td>none</td>
<td>file-based or code-based</td>
</tr>
</tbody>
</table>

*DDS2 = OpenSplice v2.0, cleanest/compliant solution*
Latency – Complex Data Type

**DDS2 = OpenSplice v2.0, fastest for larger/complex messages**
OpenSplice™ v2.1: Optimized Performance

LATENCY benchmarking

Benchmarking environment
- OpenSplice: OpenSplice v2.1
- Hardware: DELL precision M20 notebook (2.0 Ghz Pentium-M)
- OS: Windows-XP (SP2)

Benchmark Description
- Ping/Pong: roundtrip latency measurement of small topics (2 ‘long’ attributes)
- Purpose: determine ‘overhead’ of write/read processing and inter-process comm’s
- Availability: benchmark is included in the distribution

Results

<table>
<thead>
<tr>
<th>Language</th>
<th>Write</th>
<th>Read</th>
<th>Latency (roundtrip/2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>14 usec.</td>
<td>11 usec.</td>
<td>58 usec.</td>
</tr>
<tr>
<td>C++</td>
<td>14 usec.</td>
<td>11 usec.</td>
<td>58 usec.</td>
</tr>
<tr>
<td>Java</td>
<td>16 usec.</td>
<td>14 usec.</td>
<td>65 usec.</td>
</tr>
</tbody>
</table>

Conclusions
- Very fast intra-nodal communication (between applications on a single node)
- Very small ‘DDS’ overhead in general (<< 20 usec per read/write call)
- Very small JAVA-overhead (in OpenSplice, Java is a ‘pluggable-API’ on top of a C-core)
(1) PrismTech at a glance
(2) OpenSplice Background
(3) OpenSplice by Example
(4) OpenSplice Tooling
(5) OpenSplice Performance
(6) OpenSplice Availability
OpenSplice™ Availability

OpenSplice modules
- OpenSplice CORE module
- OpenSplice CONTENT AWARENESS module
- OpenSplice PERSISTENCE module
- OpenSplice DLRL module (v2.2 Q4’06)
- OpenSplice Productivity tools (OpenSplice Tuner, OpenSplice MDD-suite: beta Q4’06)

Platforms & Languages
- Languages: C, C++, Java
- Platforms:
  - X86: Windows (XP) and Linux (2.6 kernel)
  - SPARC: Solaris (8 and up)
  - PowerPC: vxWorks (5.5)

CORBA integration
- Standalone: no CORBA integration/dependencies (available for C/C++ and Java)
- C++ Corba integration (cohabitation: seamless information-passing between DDS/Corba)
  - OpenFusion TAO 1.4.1 (gcc 3.2) : pre-built (out-of-the-box)
  - ANY ORB and ANY gcc-compiler : “self-generation” of pluggable C++ API (source/scripts provided)

Courses & Consultancy
- OpenSplice courses (from introductionary to expert courses)
- PSO (Professional Services): workshops, consultancy, prototyping, outsourcing, ...
<table>
<thead>
<tr>
<th>Product Family</th>
<th>Introduction Courses</th>
<th>Programmer’s Courses</th>
<th>Advanced Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise CORBA</td>
<td>Introduction to CORBA</td>
<td>CORBA programming with C++ (TAO or eORB)</td>
<td>Advanced CORBA programming with C++ (TAO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CORBA programming with Java (JacORB or RT Orb)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CORBA programming with Ada (Ada RTOrb)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RT Java Programming</td>
<td>Real-time CORBA Programming with Java RTOrb</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OpenFusion Naming COS Product training</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>OpenFusion Trading COS Product training</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>OpenFusion Notification COS Product training</td>
<td></td>
</tr>
<tr>
<td>Embedded CORBA</td>
<td>Introduction to Embedded CORBA</td>
<td>- CORBA programming in RTE env. with eORB C++ or C Edition</td>
<td>CORBA programming on DSP with eORB C edition</td>
</tr>
<tr>
<td>OpenSplice</td>
<td>Introduction to DDS: foundation &amp; distributed data modelling</td>
<td>OpenSplice DDS programming with C++, C or Java &amp; deployment tools</td>
<td>- Advanced OpenSplice programming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DLRL programming in C++ or Java</td>
<td>- Using MDD tools for information, application &amp; deployment modelling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Using MDD tools for information, application &amp; deployment modelling</td>
<td></td>
</tr>
<tr>
<td>Xtradyne</td>
<td>Introduction to IIOP &amp; CORBA Security</td>
<td>I-DBC Product training</td>
<td>- Advanced I-DBC Product training</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Tailored IDB-C Product training</td>
<td>- Tailored WS-DBC Product training</td>
</tr>
<tr>
<td></td>
<td>Introduction to WS Security</td>
<td>WS-DBC Product training</td>
<td>Advanced WS-DBC Product Training</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tailored WS-DBC Product Training</td>
</tr>
<tr>
<td>FEATURE</td>
<td>ADVANTAGE</td>
<td>BENEFIT</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>GENERAL</td>
<td>Information-centric</td>
<td>Enable dynamic, loosely coupled syst.</td>
<td>Simplified &amp; better scalable architectures</td>
</tr>
<tr>
<td></td>
<td>Open standard</td>
<td>&quot;Off-the-shelf&quot; solutions</td>
<td>Lower cost, no vendor-lock-in</td>
</tr>
<tr>
<td></td>
<td>Built on proven tech.</td>
<td>Meant for most demanding envntr.</td>
<td>Assured quality and applicability</td>
</tr>
<tr>
<td></td>
<td>TNL / PT 'heritance'</td>
<td>Decade long of 'DDS experience'</td>
<td>Proven suitability in mission-critical domain</td>
</tr>
<tr>
<td>FUNCTIONAL</td>
<td>Real-time pub/sub</td>
<td>Dynamic/asyncrhonous data comm.</td>
<td>Autonomous de-coupled applications</td>
</tr>
<tr>
<td></td>
<td>Persistence profile</td>
<td>Fault tolerant data-persistence</td>
<td>App. fault-tolerance and data high-availability</td>
</tr>
<tr>
<td></td>
<td>Content-sub. Profile</td>
<td>Reduced complexity &amp; higher perf.</td>
<td>Easier application design &amp; scalable systems</td>
</tr>
<tr>
<td>PERFORMANCE</td>
<td>Shared-memory</td>
<td>Low footprint, instant data-availability</td>
<td>Processor Scalability</td>
</tr>
<tr>
<td></td>
<td>Smart networking</td>
<td>Efficient data-transport</td>
<td>Network Scalability</td>
</tr>
<tr>
<td></td>
<td>Extensive IDL sup.</td>
<td>Includes unbounded strings,sequences</td>
<td>Data Scalability</td>
</tr>
<tr>
<td>USABILITY</td>
<td>Multiple language</td>
<td>Any (mix) of C, C++, Java, Ada</td>
<td>Supports (legacy) code, allows hybrid systems</td>
</tr>
<tr>
<td></td>
<td>Multiple platforms</td>
<td>Any (mix) of Enterprise &amp; RTE Oss</td>
<td>Intercons enterprise and embedded systems</td>
</tr>
<tr>
<td>TOOLING &amp; EASE-OF-USE</td>
<td>All meta-data at runtime</td>
<td>Dynamic discovery of all 'entity-info'</td>
<td>Guaranteed data-integrity</td>
</tr>
<tr>
<td></td>
<td>Powerful tooling</td>
<td>Support for complete system lifecycle</td>
<td>Enhanced productivity and System Integration</td>
</tr>
<tr>
<td></td>
<td>Remote connect</td>
<td>Web-based remote access &amp; control</td>
<td>Remote diagnostics using standard protocols</td>
</tr>
</tbody>
</table>

Legend: EQUAL to competition BETTER than competition FAR-SURPASSING competition
OpenSplice Summary

**Functionality**
- Only product with full OMG-DDS coverage
  (DCPS already, DLRL starting at v2.2 in December)
- Only OpenSplice qualifies as a ‘fault-tolerant information backbone’
  (rather than just pub/sub messaging)
- OpenSplice Tuner as superior ‘deployment’ tool available today
  (offering total & remote control)
- Unique DDS-domain specific OpenSplice MDD tool-suite
  (beta in v2.2)

**Non-functionals**
- Optimized Internal architecture for large-scale yet realtime systems
- Fastest DDS today (for structured datatypes > 256 bytes) also with least jitter
- Unique networking-implementation (network-partitions and network-channels)
- MDD suite targeting easier/faster/cheaper development & deployment

**Built on 15 years of experience in most-demanding environments**
- TNL ‘TACTICOS’ history: proven, fielded
- CMS characteristics: large-scale, real-time, fault-tolerant, embedded
For more OpenSplice info: www.prismtech.com
Or contact me: hans.vanthag@prismtech.com

THANK YOU!