PRISMTECH
Productivity Tools & Middleware

"Using the Eclipse Framework to Build an MDD tool chain for the DDS Application Domain "

For OMG RTE workshop, July 2007

Robert Kukura (robert.kukura@prismtech.com)

CONTENTS

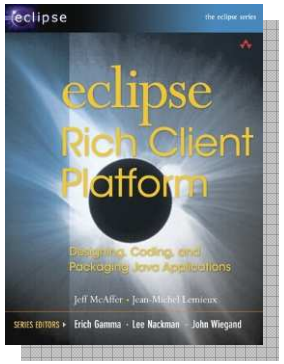
- *What is Eclipse*
- *Why do we need MDE*
- *Using Eclipse to build an MDE toolsuite for DDS*
- *Example: An MDE-enabled DDS product-line*
- *Demo: OpenSplice PowerTools™*

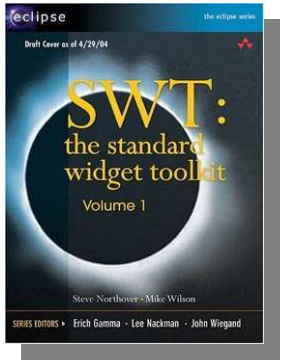
PRISMTECH
Productivity Tools & Middleware

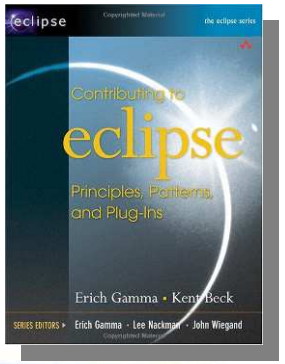
… What is Eclipse …

Eclipse is all these things…

- A Java IDE
- An IDE Framework
- A Tools Framework
- An Application Framework
- An Open Source Enabler
- A community
- An eco-system
- A foundation

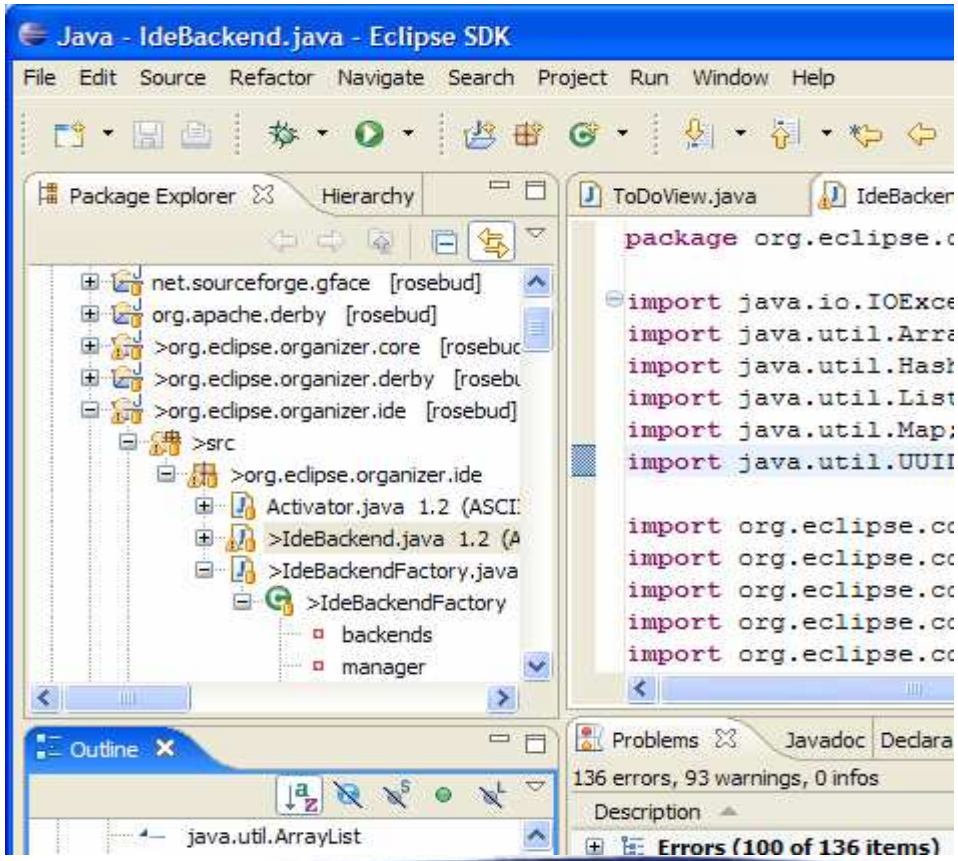






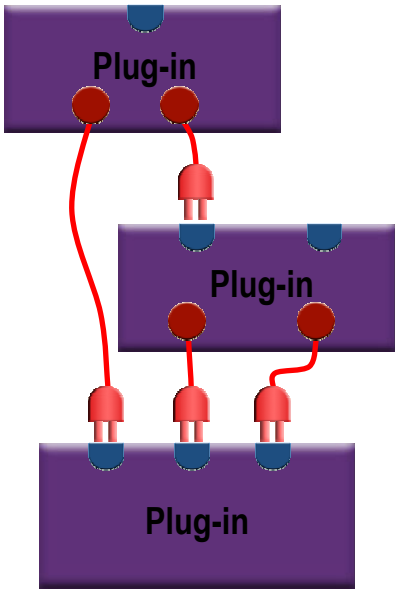
Eclipse is a Java IDE

- Widely regarded as ***the*** Java development environment
- With all the bells and whistles…
  - Language-aware editors, views, …
  - Refactoring support
  - Integrated unit testing and debugging
  - Incremental compilation and build
  - Team development support
    - Out of the box support for CVS
  - …



Eclipse is a Tools Framework

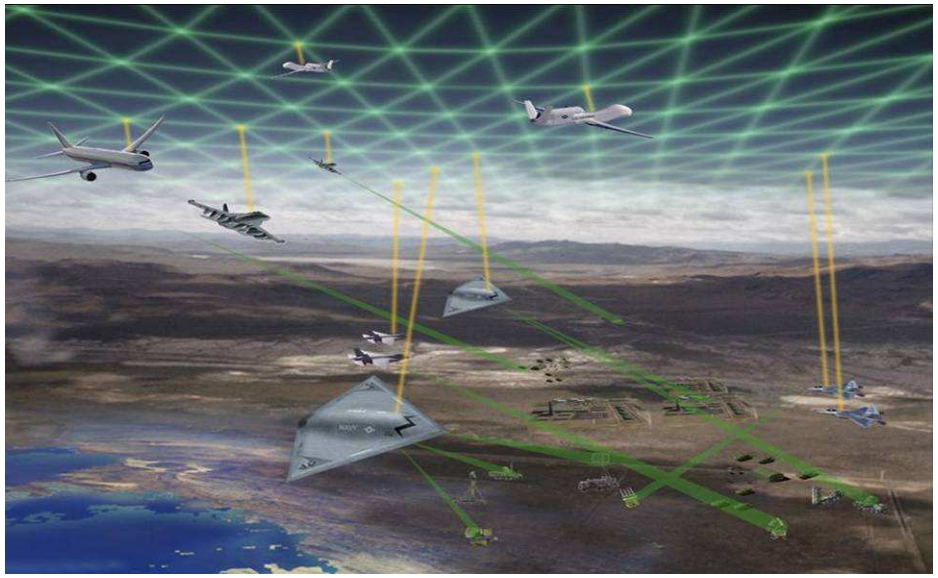
- Extensibility through OSGi implementation
  - Plug-ins make Eclipse whatever you need it to be
- Focus on developing a universal platform of frameworks and exemplary tools
- Tools extend the Eclipse platform using plug-ins
  - Business Intelligence and Reporting Tools (BIRT)
  - Eclipse Communications Framework (ECF)
  - Web Tools Project (WTP)
  - **Eclipse Modelling Framework (EMF)**
  - Graphical Editing Framework (GEF)
  - Test and Performance Tooling Project (TPTP)

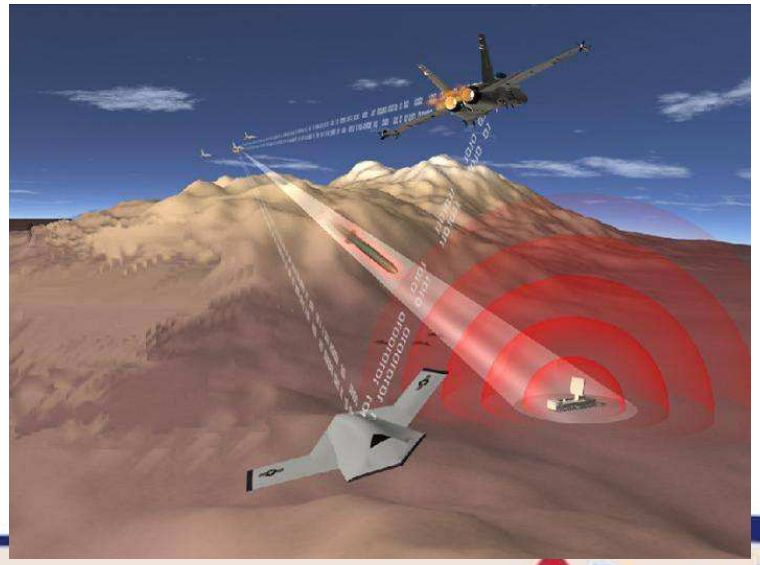


PRISMTECH
Productivity Tools & Middleware

… Why do we need MDE …

More Complex Systems and Requirements

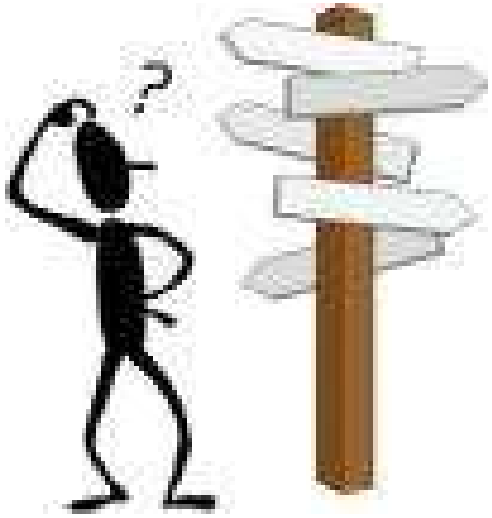




PRISMTECH
Productivity Tools & Middleware

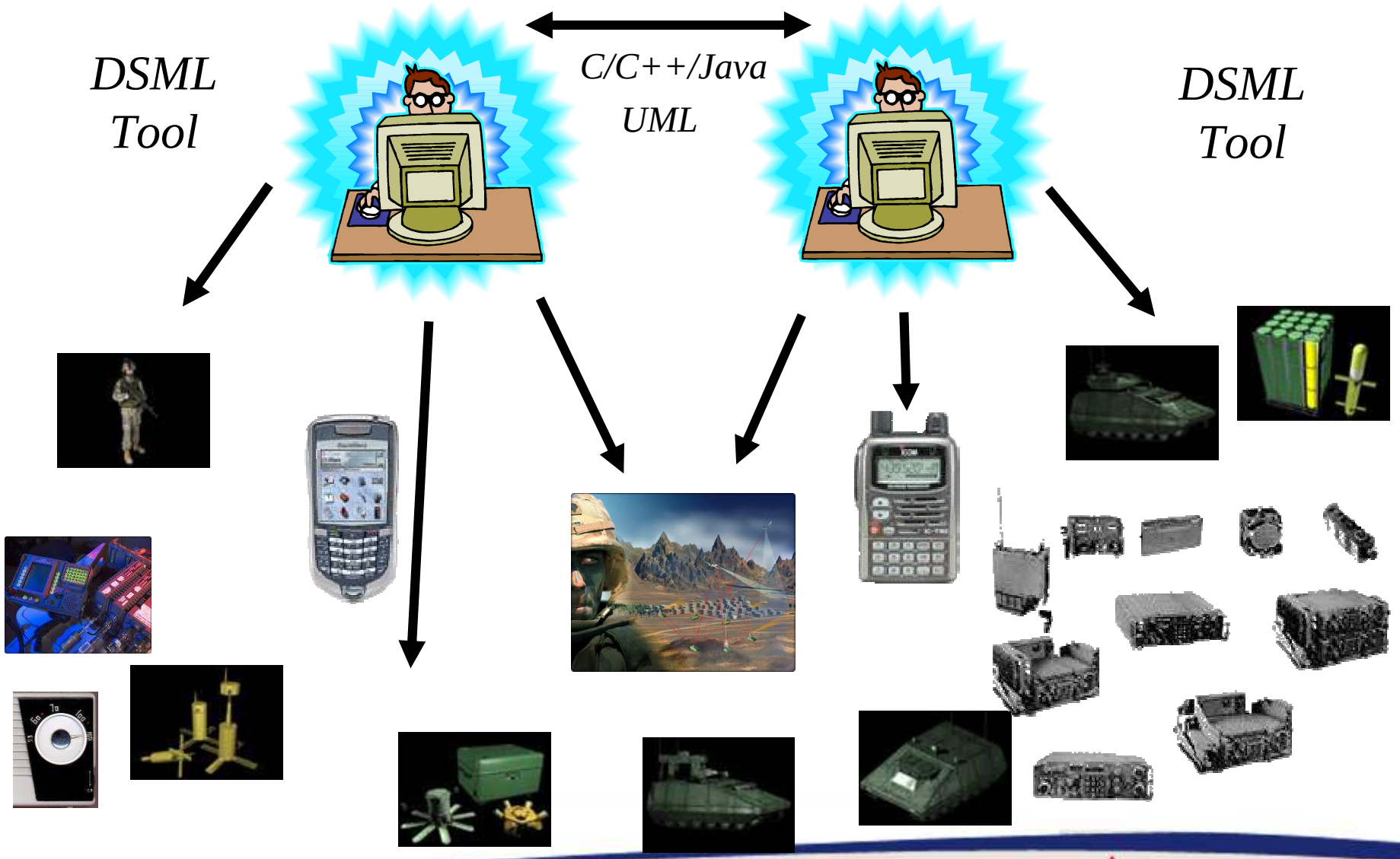
Complex Publish Subscribe Systems

- Object Oriented
- Component Based
- Multithreaded/MultiProcess
- Real-time
- Embedded
- C++/Java/Ada/VHDL
- Platform Independent
- High Performance
- Heterogeneous
- Distributed
- Vital
- Secure
- Fault Tolerant
- Portable
- Standardized
- Declarative
- Imperative
- Dynamic



So what's the big deal?
Each one *by itself* is difficult, let alone doing them all *at the same time*

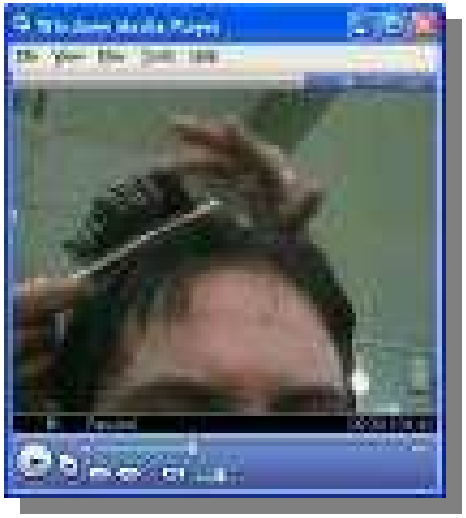
Providing sufficient tools to do the job



Domain Specificity

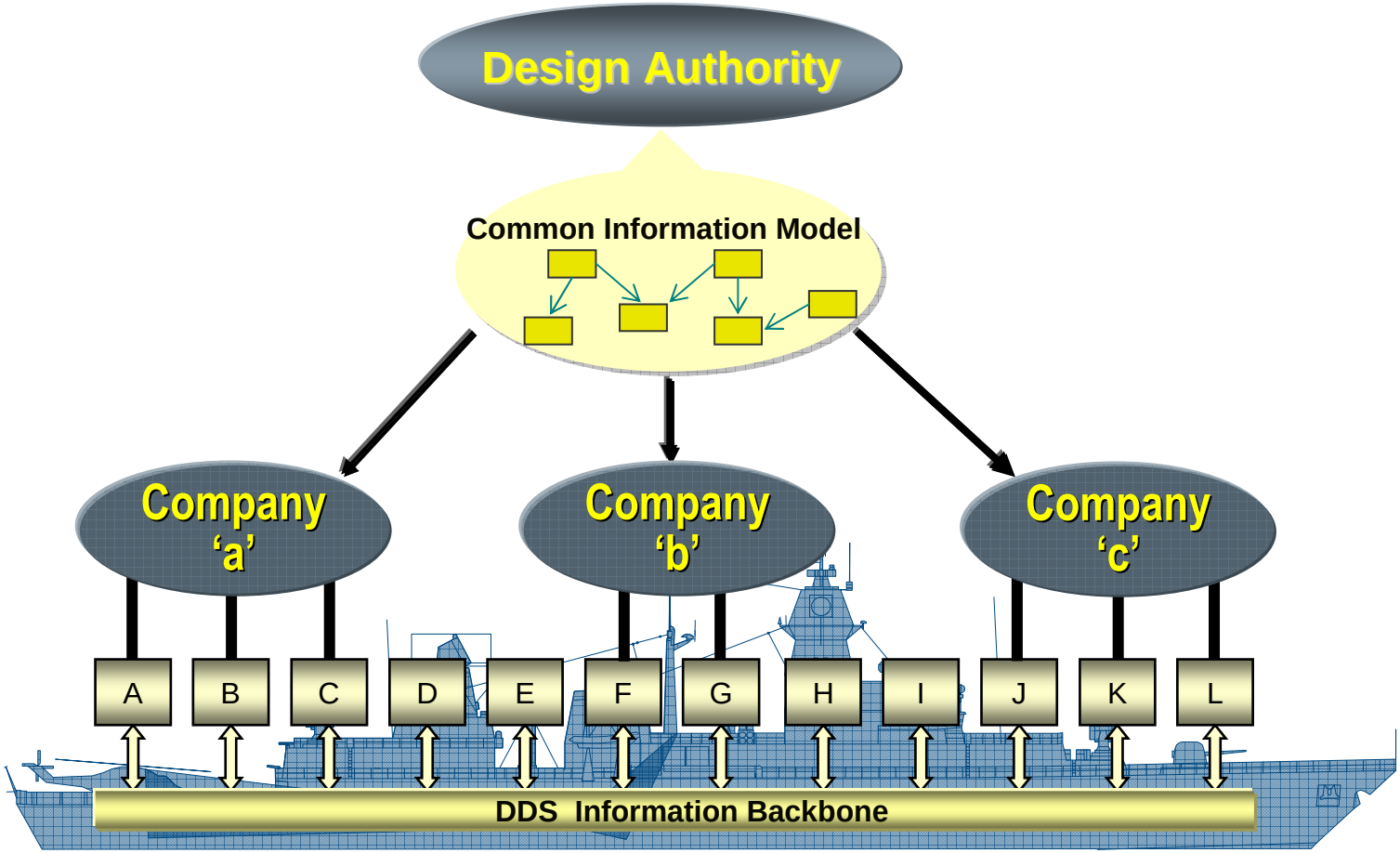
Domain Independent

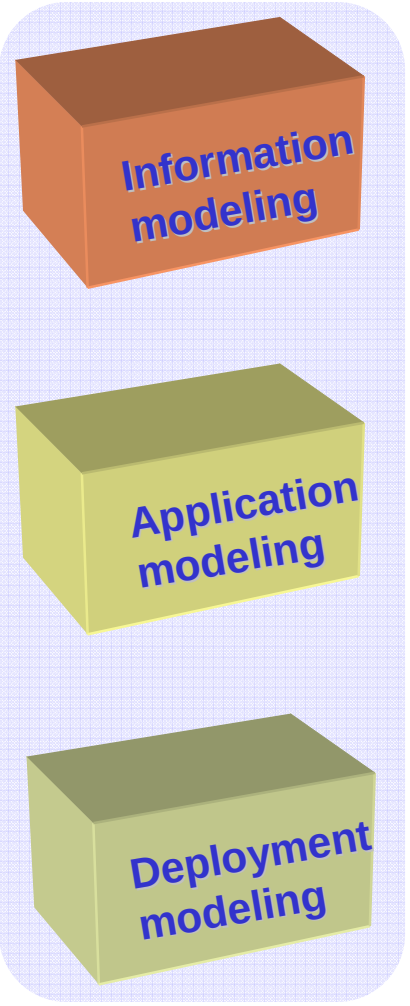
Domain Specific - Tools



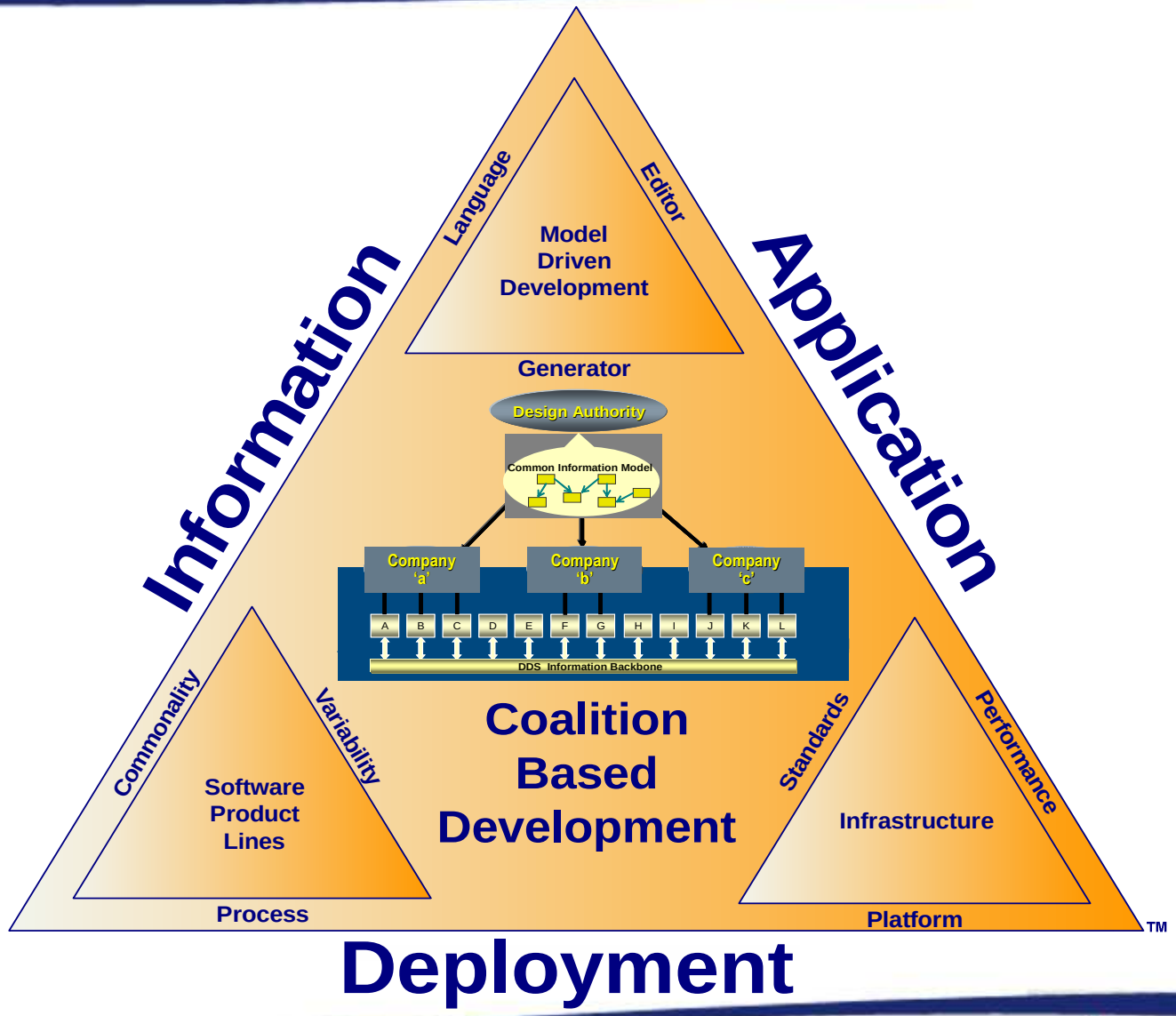
The task at hand

MDE-aspects in 'Coalition Based Development'





The “Holy Grail” ??….



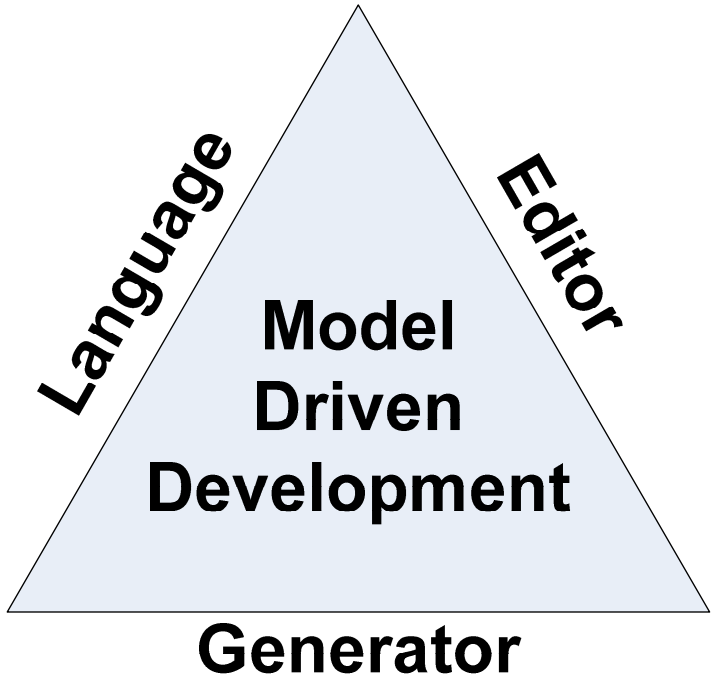
PRISMTECH
Productivity Tools & Middleware

PRISMTECH
Productivity Tools & Middleware

... Using Eclipse to build and MDE toolsuite for DDS …

GOALS:

- Complete modeling of system design cycle
  - Fast, intuitive, correct
- Information Modeling
  - Documented packages of re-usable topic-sets
- Application Modeling
  - Code-generation
- Deployment Modeling
  - Runtime control & round-trip engineering



DDS PowerTools Implementation Technologies

- **Eclipse Based**
  - Open development platform
  - Extensible frameworks, tools

- **Eclipse Modeling Framework**
  - UML Modelling of Domain Model
  - OCL/Java Constraint Implementation
  - JET2 Code Generation (a bit like JSP)

- **Graphical Editing Framework**
  - 2D Drawing API
  - Higher Level MVC Based framework

DDS PowerTools Meta Model Definition

- Meta Model is the heart of the PowerTools - Defined in UML
  - IDL Meta Model Foundation
    - Defined in UML
    - Based on Interface Repository 'information model'
    - Pure IDL – No DDS specific elements
  - DCPS Meta Model woven in
    - Defined in UML
    - Adds DCPS related info to IDL Meta Classes
    - Adds additional DCPS Meta Classes
  - DLRL Meta Model woven in
    - Defined in UML
    - Adds DLRL related info to IDL Meta Classes
    - Adds additional DLRL related Meta Classes.
- Implemented in EMF

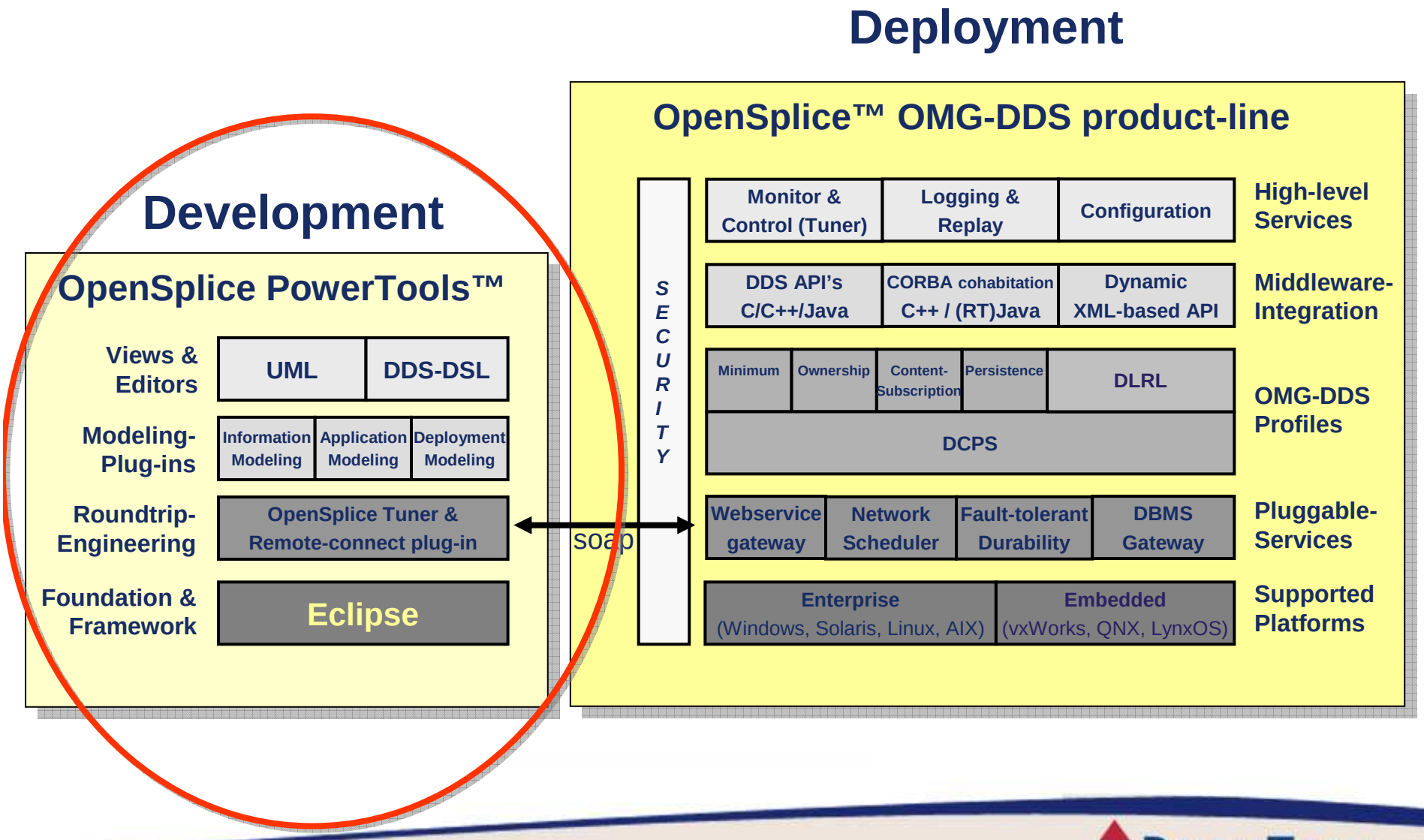
DDS PowerTools Meta Model Implementation

- PowerTools Meta Model Implemented in EMF
- EMF provides the foundation for interop between MDD tools
- Meta Models are specified in UML, including OCL constraints
- EMF Provides Validation Framework to run constraints
- EMF Generates Java classes for Meta Model
- EMF Generates Edit Framework & Editor for Model
  - Java Bean style get/set methods
  - Referential Integrity
  - Cardinality constraints
  - XMI Serialization/Deserialization
  - Notification (event) mechanism
  - Adapter Mechanism.

PRISMTECH
Productivity Tools & Middleware

… RESULT:
An MDE-Enabled DDS Product Line

A ‘MDE-enabled’ DDS product line



Introducing The Example

SENSOR PROCESS

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

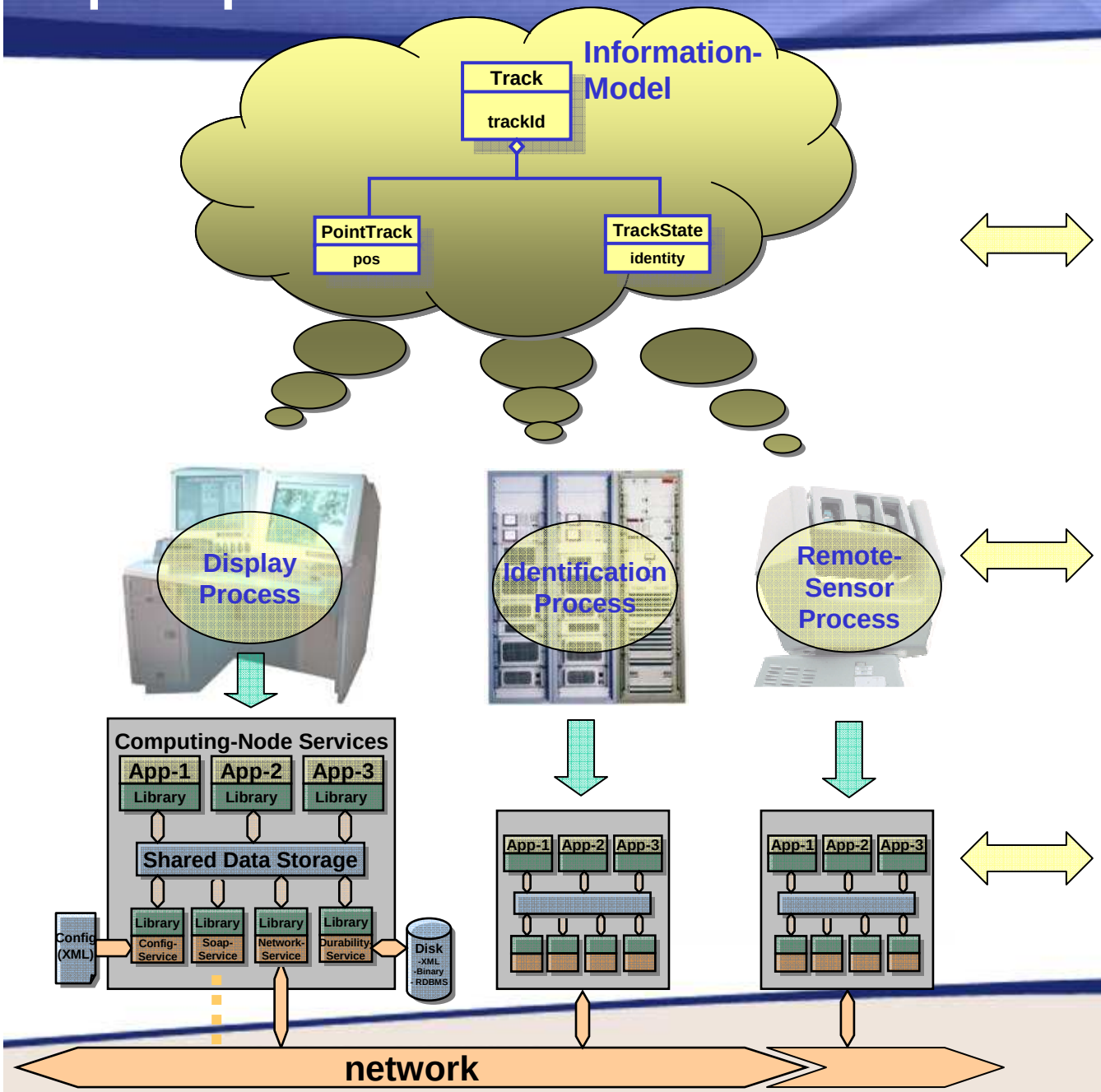
CLASSIFICATION PROCESS

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

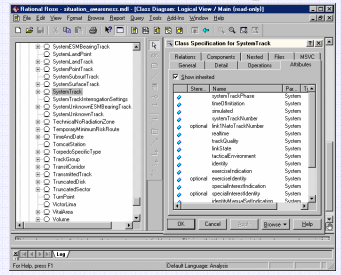
DISPLAY PROCESS

- Displays track info
- Both position & identity
- Raises alerts
- Requires ‘***pointTrack’***
- Requires ‘***trackState’***

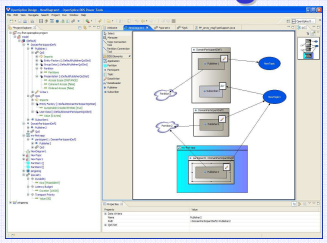
OpenSplice DDS PowerTools™ MDE Suite



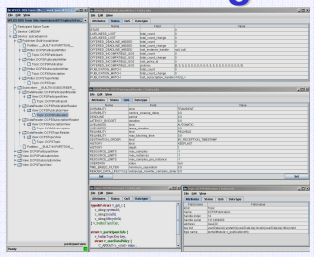
Information modeling



Application modeling

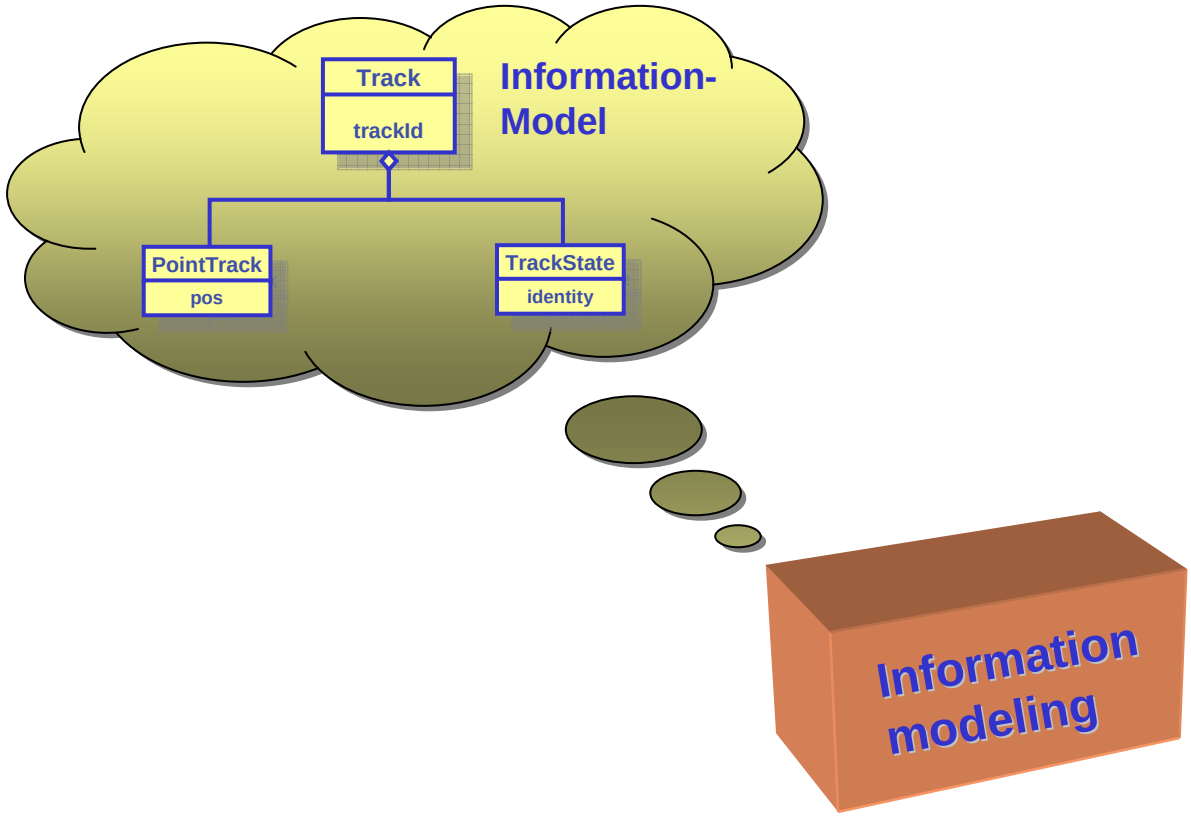


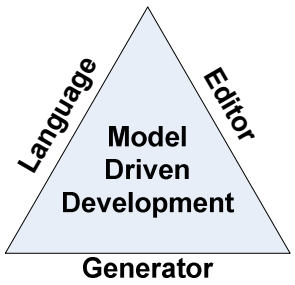
Deployment modeling



PRISMTECH
Productivity Tools & Middleware

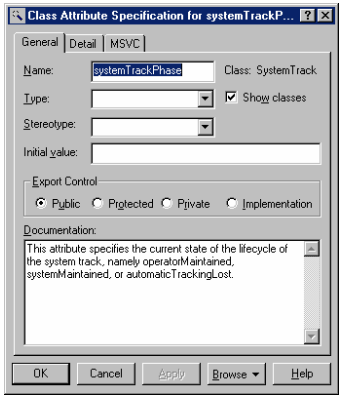
Information modeling



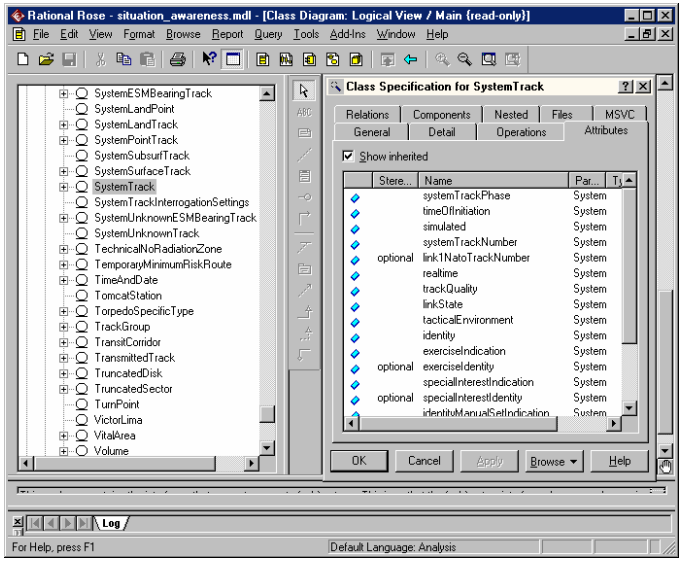


Information modeling: Scope & Purpose

- **Language**
  - Define the ‘spoken language’ in the system
    - Requirements phase: to talk with the customer
    - Development phase: to talk between components
  - DDS-Domain specific ‘Topics’ (Type + QoS + keys)

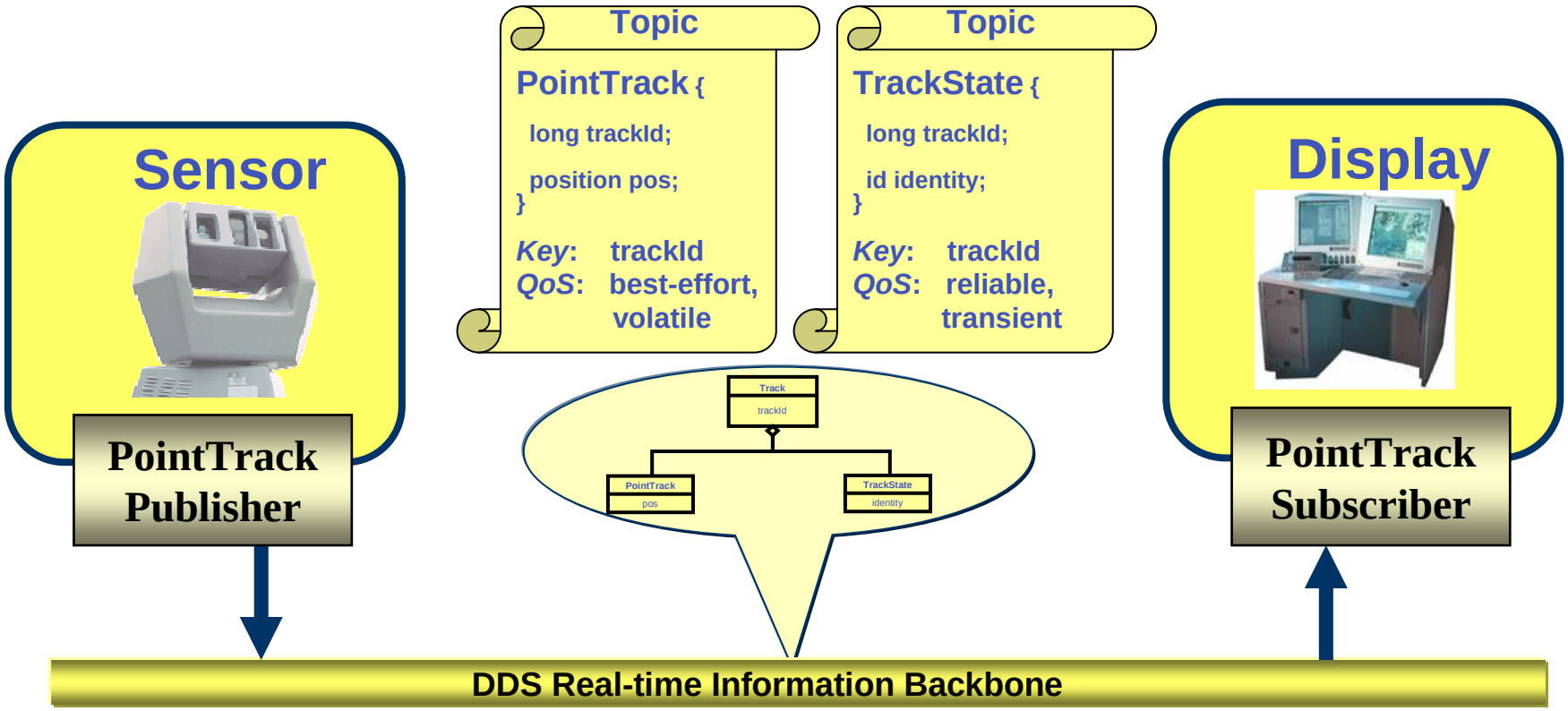


- **Editor**
  - Semantics: UML (‘annotated’)
  - Syntax: IDL (optional MDE-input if predefined)
  - Behavior: QoS (reliability, urgency, durability)



- **Generator**
  - Types (IDL) from UML (or direct IDL-import)
  - Topics from Types
  - DDS-entities
    - Topic-creation, QoS-setup, typed readers/writers

Information modeling (example)



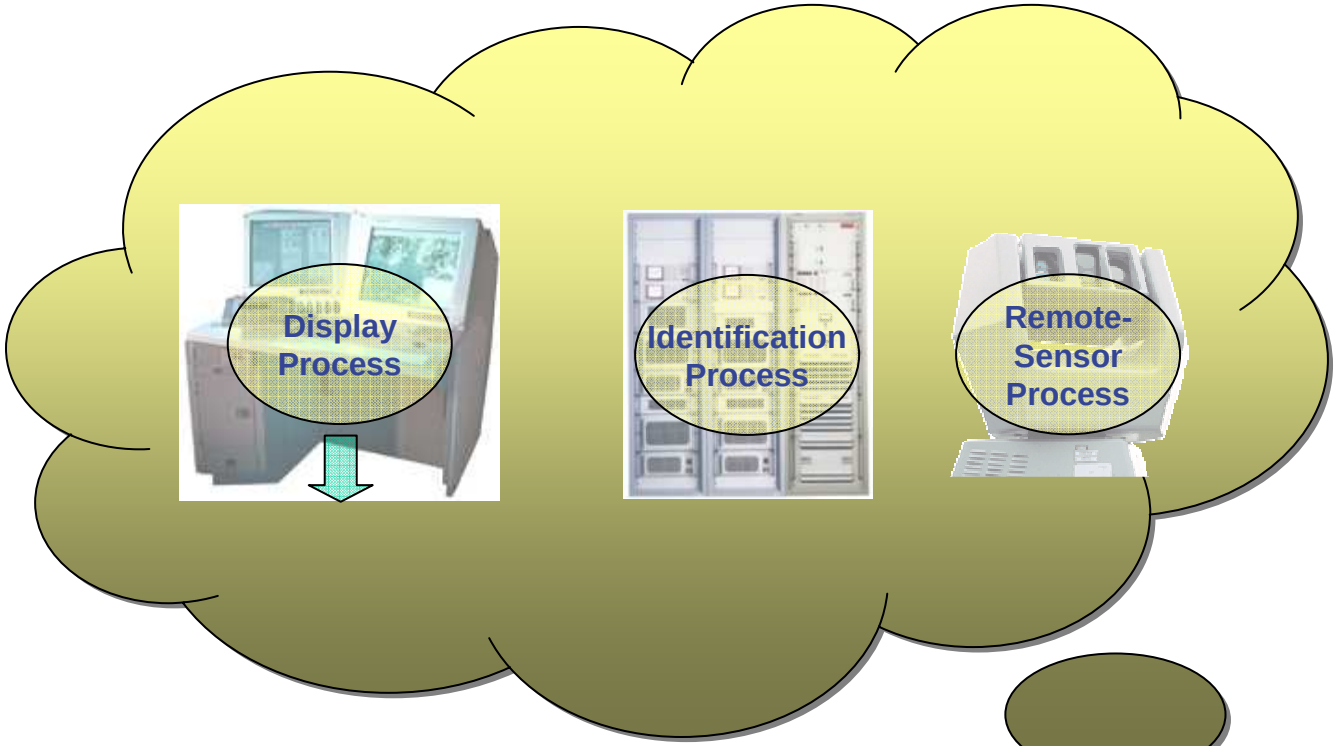
Task: Information modeling

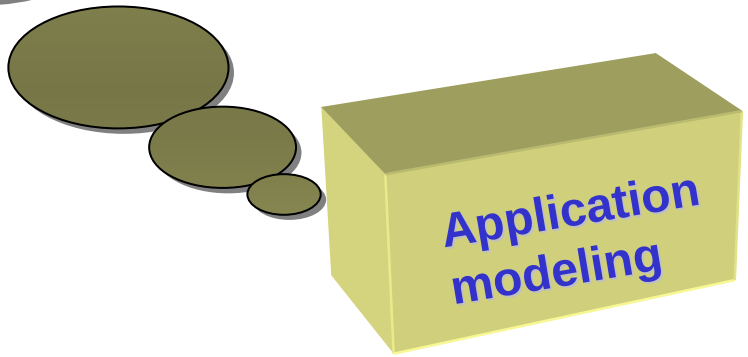
- Model the 'PointTrack' and 'TrackState' topics
- Model the system-wide behaviour of topics (QoS)
- Model the relationships between the Topics (keys)
- Separate these concerns from applications

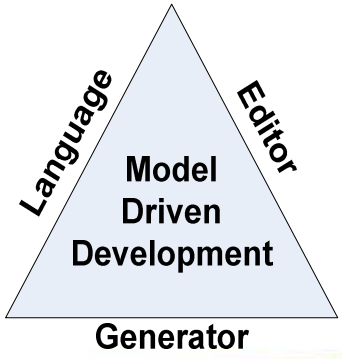
MDE: Features / Advantages

- Graphical modeling of structure and QoS (intuitive, fast)
- DDS-entity code generation (topics, interfaces)
- Allowing for direct utilization (by applications or tool)
- Documented packages of re-usable topic-sets

Application modeling



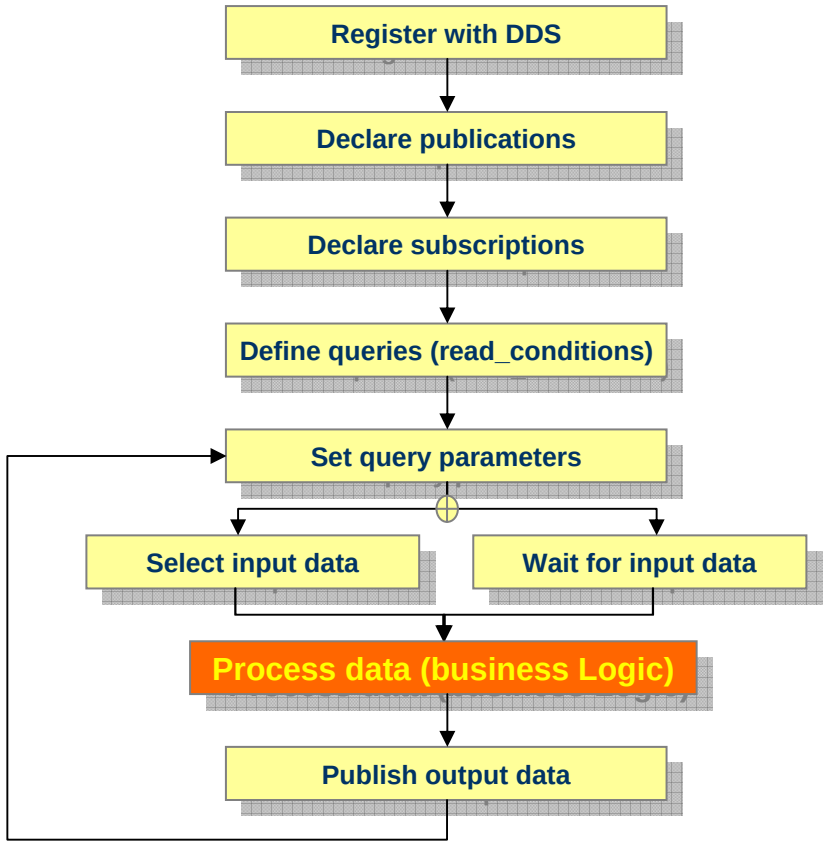




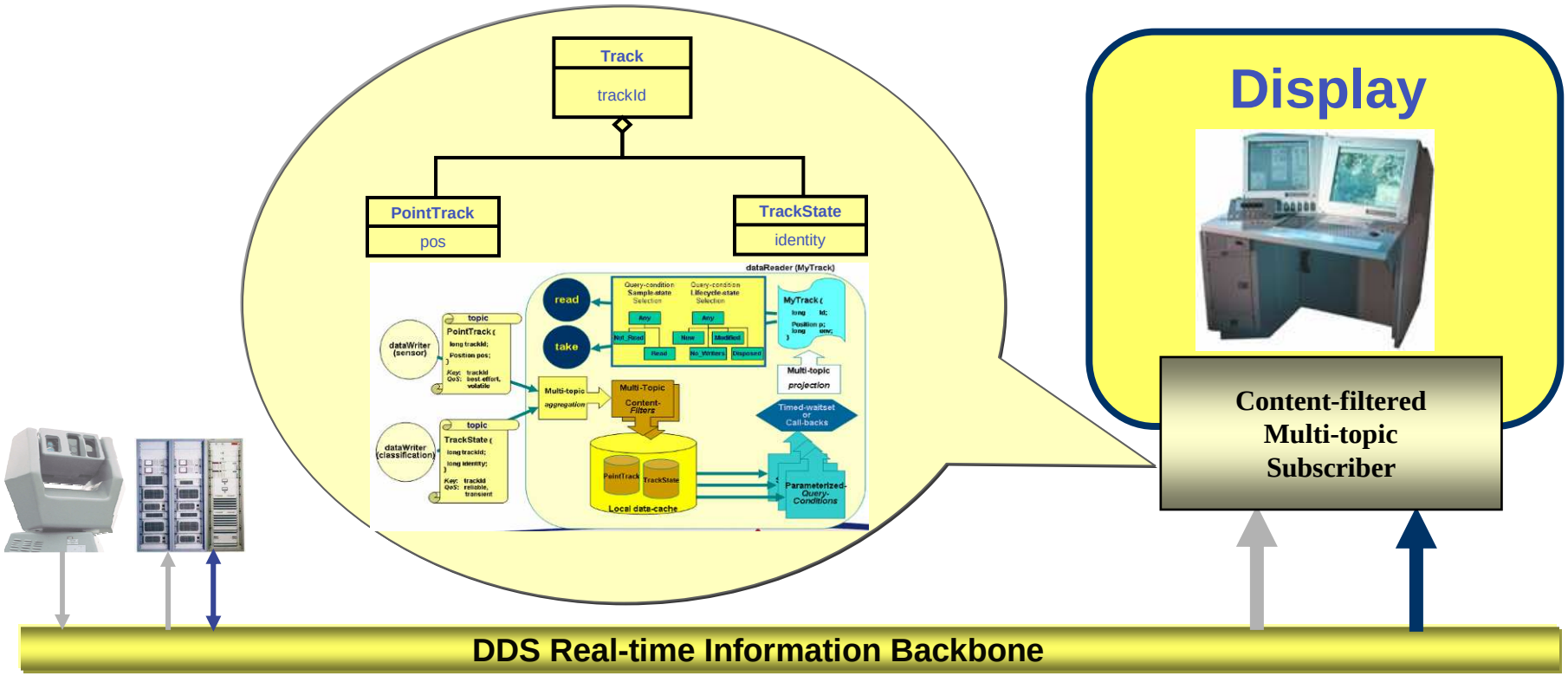
Application modeling: Scope & Purpose

- **Language**
  - Processing-language
  - DDS 'keywords'
    - publishers/writers, subscribers/readers,..
    - Content-filters, queries, waitsets, listeners,…
- **Editor**
  - Templates & Patterns (loop, MVC, …)
  - Application-level QoS modeling (history, filters)
  - Process modeling (waitsets, listeners)
- **Generator**
  - Application-framework from templates
  - DDS entities from (graphical) model
  - Annotated application- and test-code

Simple DDS-Loop Pattern



Application modeling (example)



Task: Display Process modeling

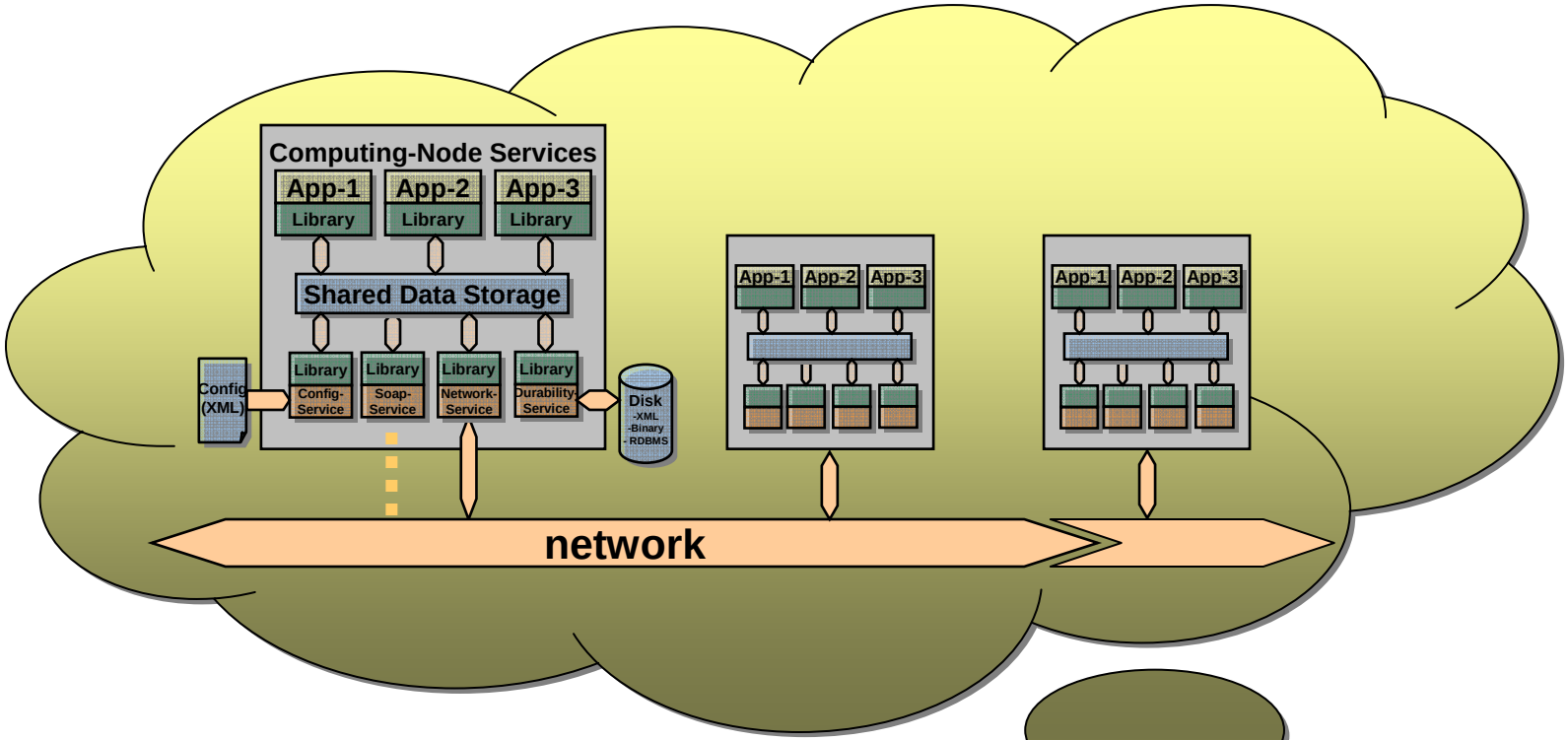
- Model 'aggregate interest'
  - 'multi-topic' or 'DLRL-object' with local 'QoS'
- Model display process (periodic loop)
  - Handle first-appearances of hostile tracks with prio

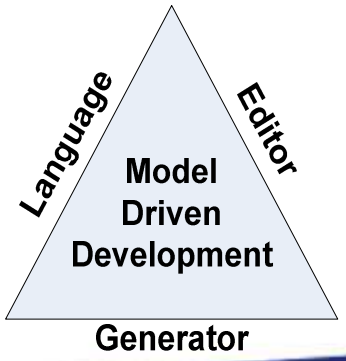
MDE: Features / Advantages

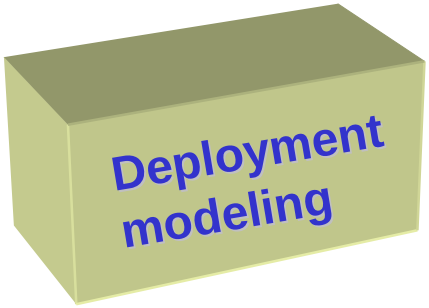
- Import information-model
- Provide Application-framework (from 'loop' template)
- Model DLRL-objects, Multi-Topics, Queries, Waitsets,…
- Developer can concentrate on 'business-logic' (GUI)

PRISMTECH
Productivity Tools & Middleware

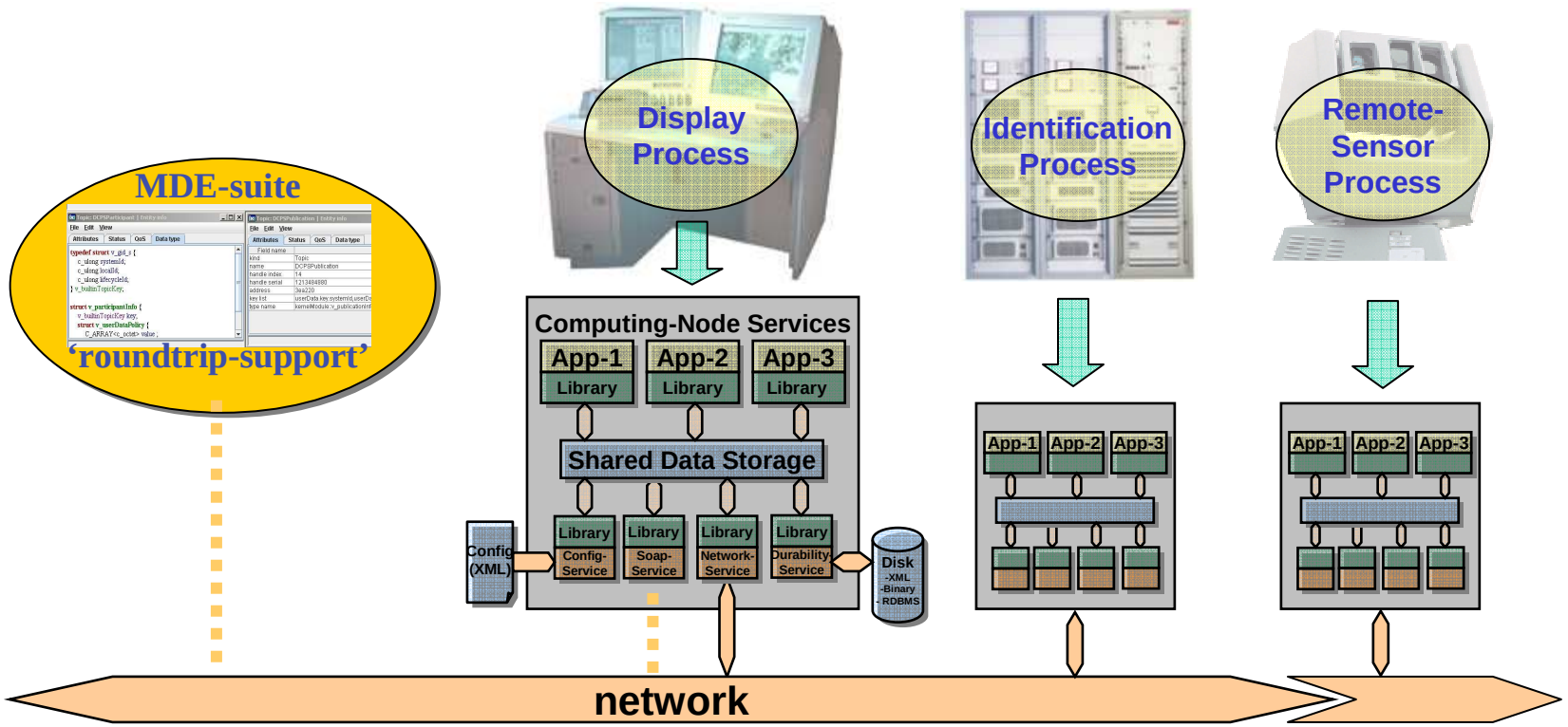
Deployment modeling







Deployment modeling (example)



Task: Deployment modeling ('Control & Monitoring')

- Configure Durability-service (TRANSIENT TrackState topic)
- Configure Network-channels (priority, reactivity)
- Configure Resources (resource-limits)
- Remote control & Monitoring of deployed system

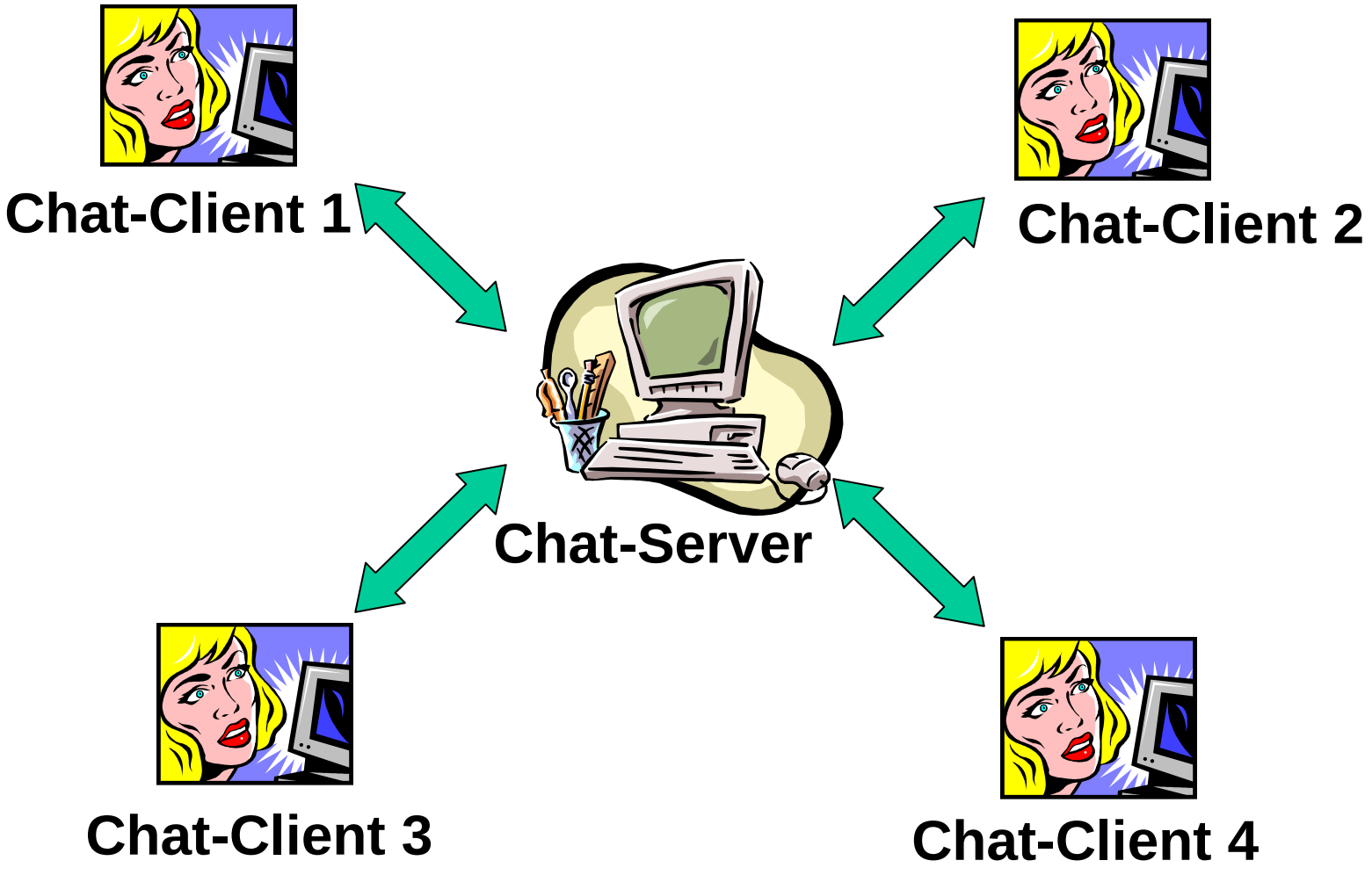
MDE: Features / Advantages

- Round-trip engineering
  - Deploy & tune models (info & application)
- Control & Monitor deployed system
  - Using the domain specific 'DDS-entity language'

PRISMTECH
Productivity Tools & Middleware

... OpenSplice PowerToools™..
Eclipse based MDE-suite: Demo

DEMO – Traditional Chat architecture



DDS Example: C/S v.s. P/S

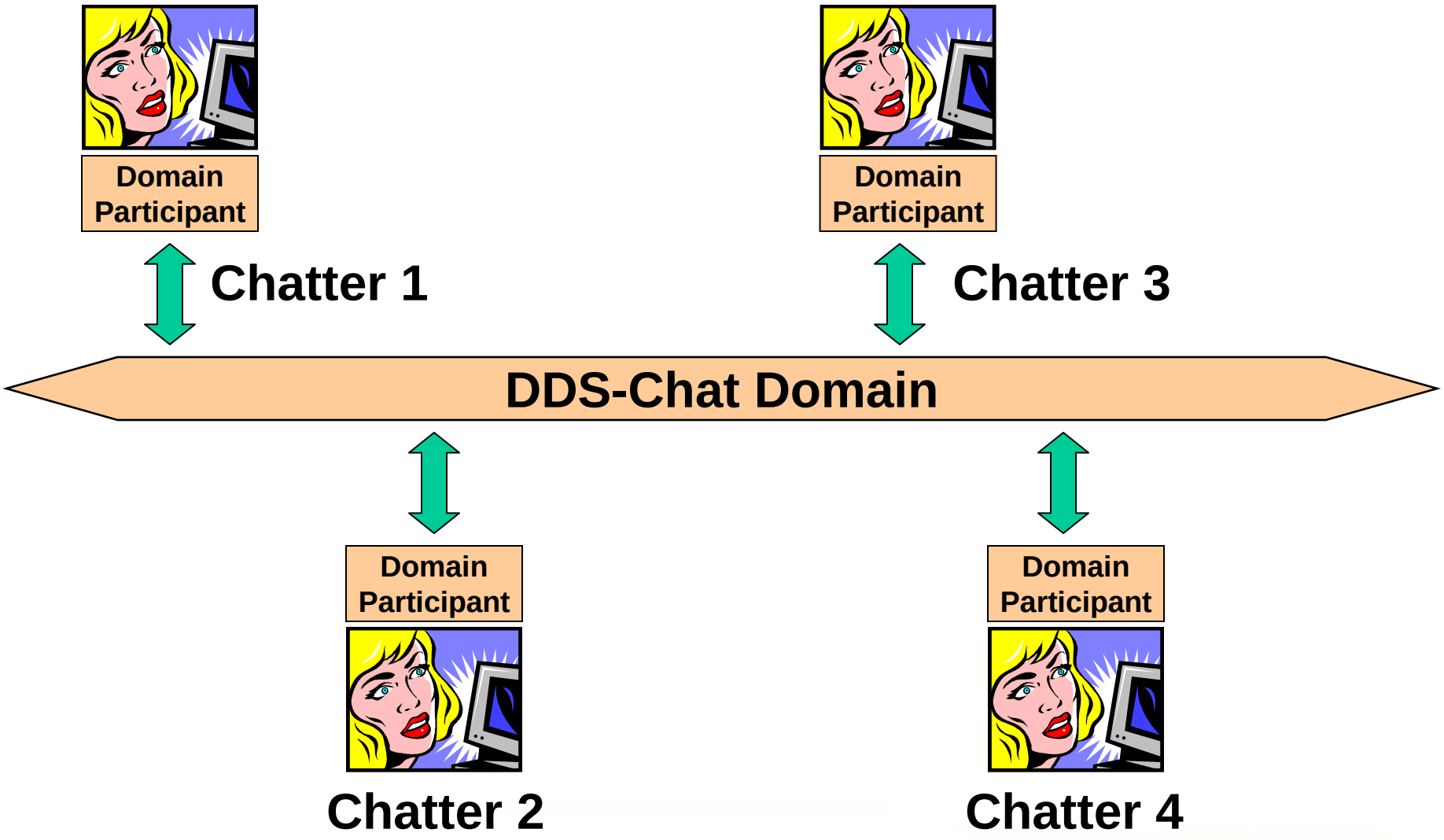
Typical sequence of events on a **traditional** Chat-application:

- **Connect** to the Chat-Server.
- **Transmit** your identity.
- **Download** the identities of the other chatters.
- **Receive** chat messages from other users.
  - These messages are **forwarded** to you by the server.
- **Write** your own chat-messages.
  - These messages are **forwarded** by the server to all the other users.

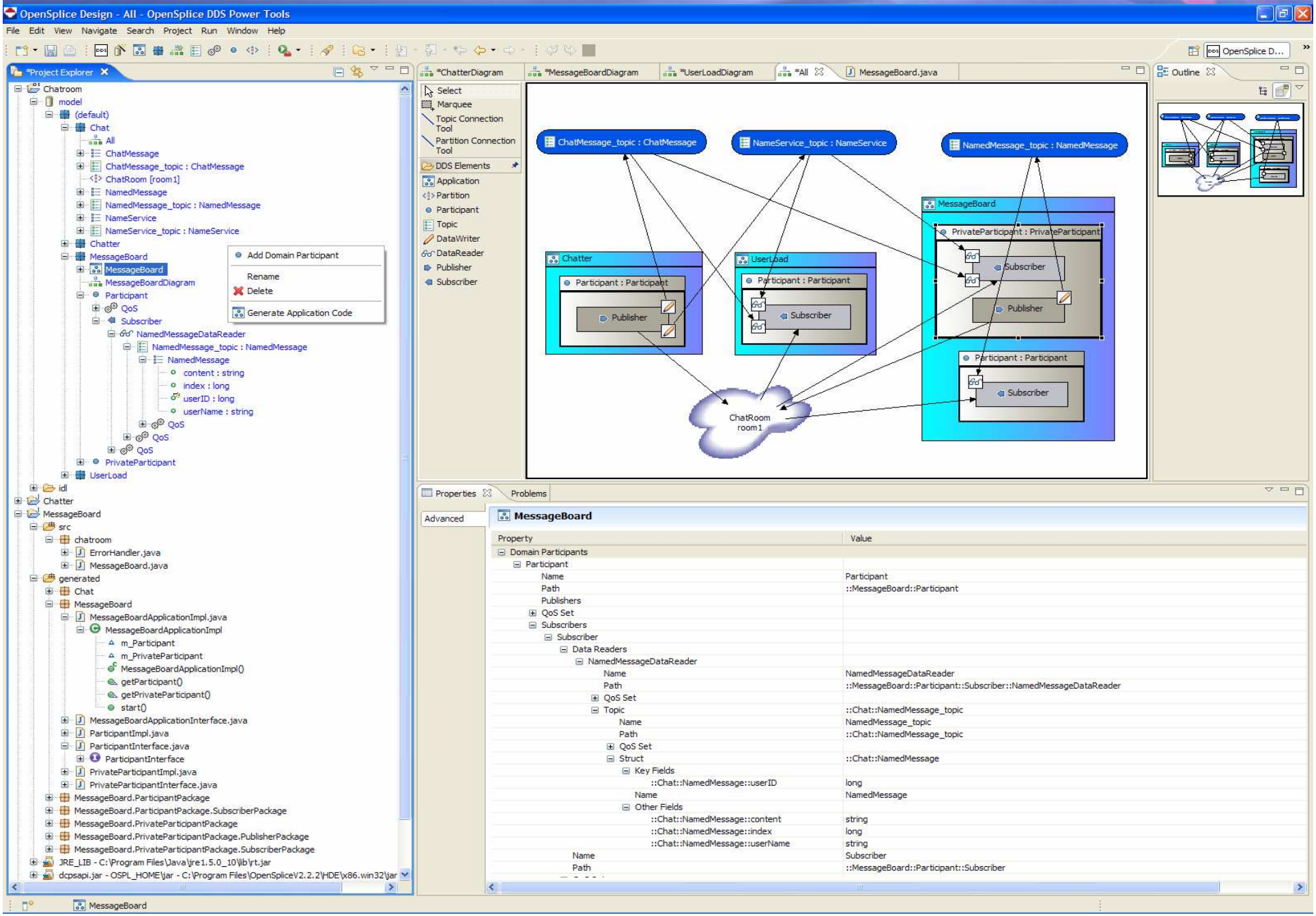
Typical sequence of events on a **DDS-based** Chat-application:

- **Participate** in the Chat-domain.
- **Publish** your identity.
- **Subscribe** yourself to the identities of all other chatters
- **Subscribe** yourself to all chat-messages in the Chat-domain.
  - All messages are delivered to you **directly** by their respective writers.
- **Publish** your own chat-messages.
  - Your messages are **directly** delivered to all the other interested users.

Example: DDS based Chat architecture



OpenSplice Power Tools™: Eclipse based MDE-suite



DEMO: DDS-based Chatroom Applications

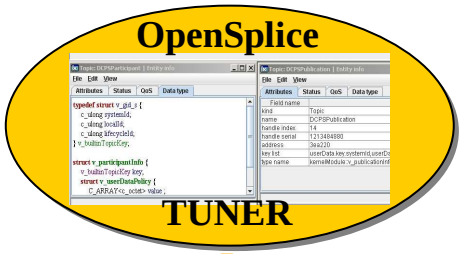
Chatters

Domain Participant

Domain Participant

Chatter Hans
userID = 1

Chatter Robert
userID = 2



ChatMessage {

int userID;
string Message;
}

Key: userID
QoS: Reliable, Volatile

NameService {

int userID;
string Name
}

Key: userID
QoS: Reliable, Transient

Information-model

DDS-Chat Domain

SHOWS ALL MESSAGES

Domain Participant

Message - Board

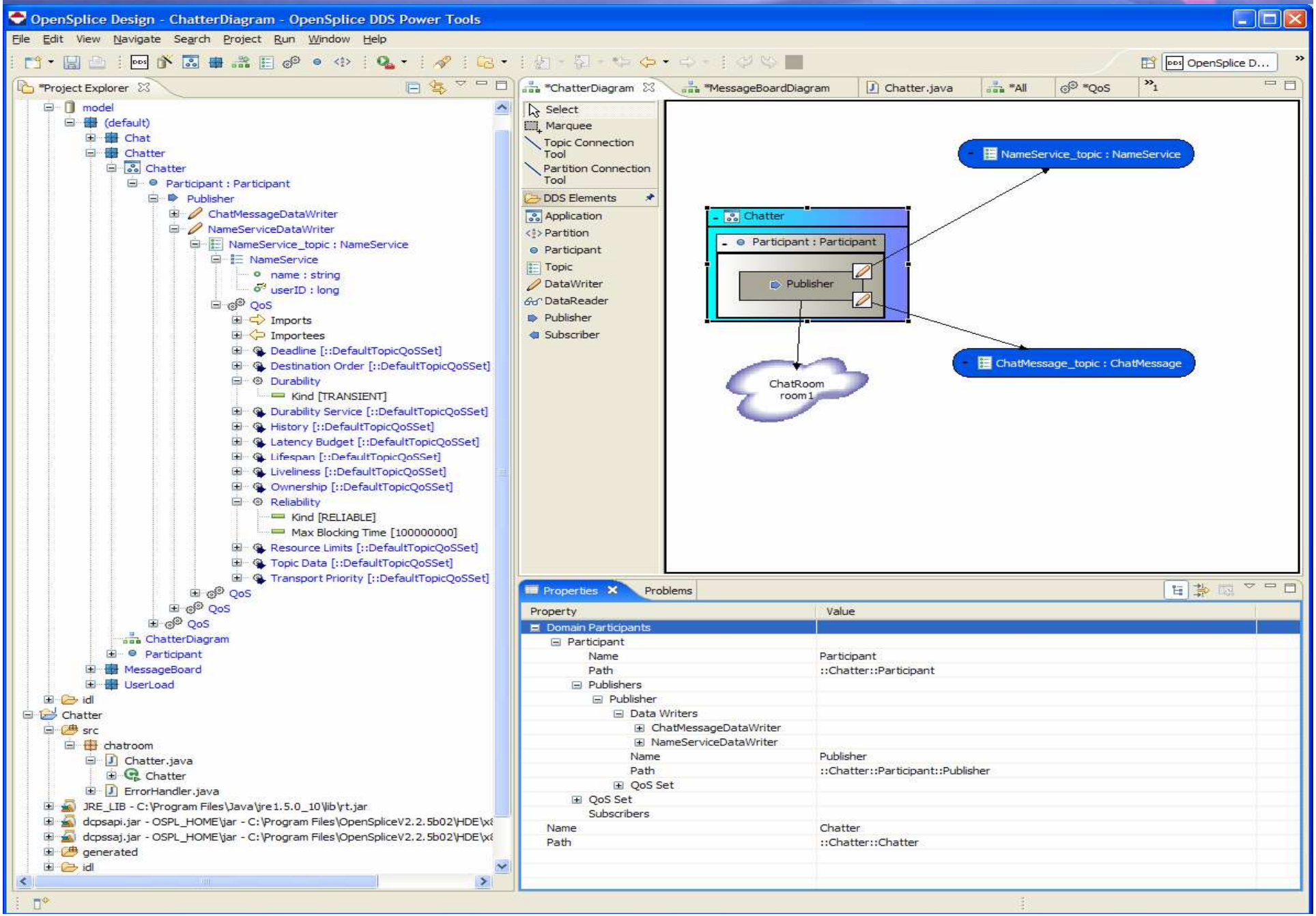
Domain Participant

User - Load

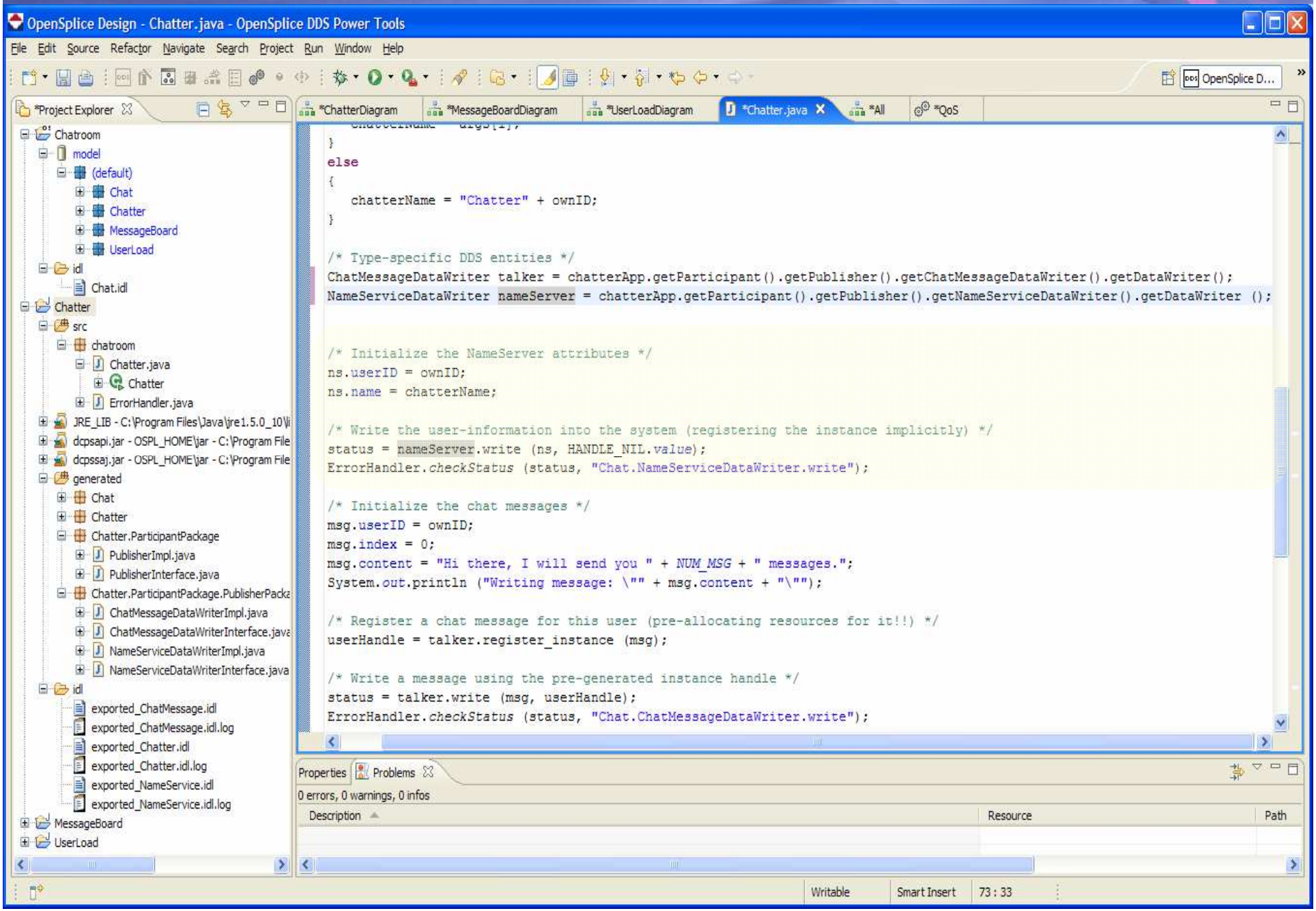
TRACKS ALL CHATTERS

PRISMTECH
Productivity Tools & Middleware

PowerTools™ Graphical modeling: Intuitive, Easy and Fast



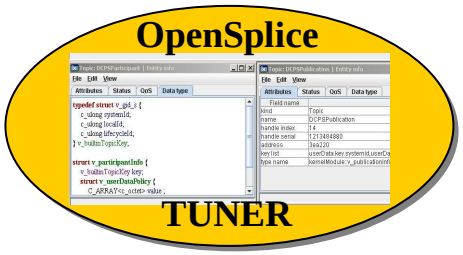
PowerTools™ Code Generation: Reduced complexity



DEMO: DDS-based Chatroom: DCPS, DLRL .. and even SQL

Domain Participant

Domain Participant




```
ChatMessage {
  int userID;
  string Message;
}
Key:  userID
QoS:  Reliable,
      Volatile
```

```
NameService {
  int userID;
  string Name
}
Key:  userID
QoS:  Reliable,
      Transient
```

Information-model

Chatter **Hans**
userID = 1

Chatter **Robert**
userID = 2

DDS-Chat Domain *(ChatRoom Partition)*

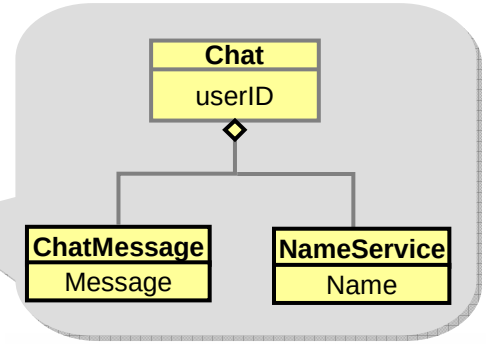
Domain Participant

DCPS
Message - Board

Domain Participant

DLRL-layer

DLRL
OO Message - Board



Object-model

DBMS Service
Filter : userID != 1

mySQL DBMS

SQL
mySQL Message Board

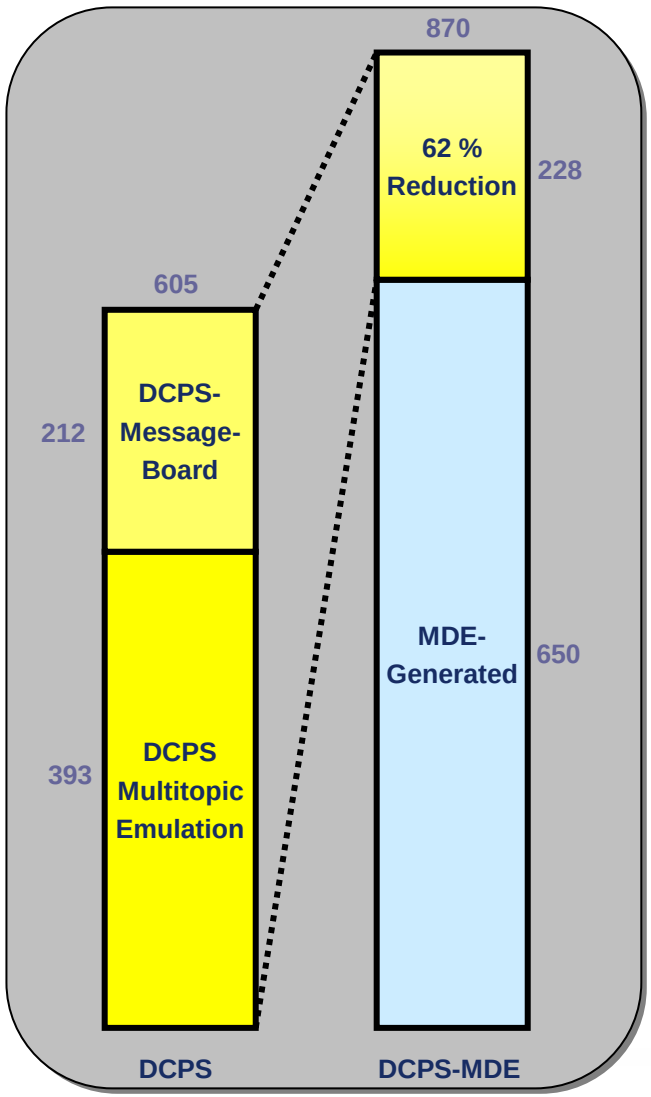
DBMS Service
Filter : userID != 2

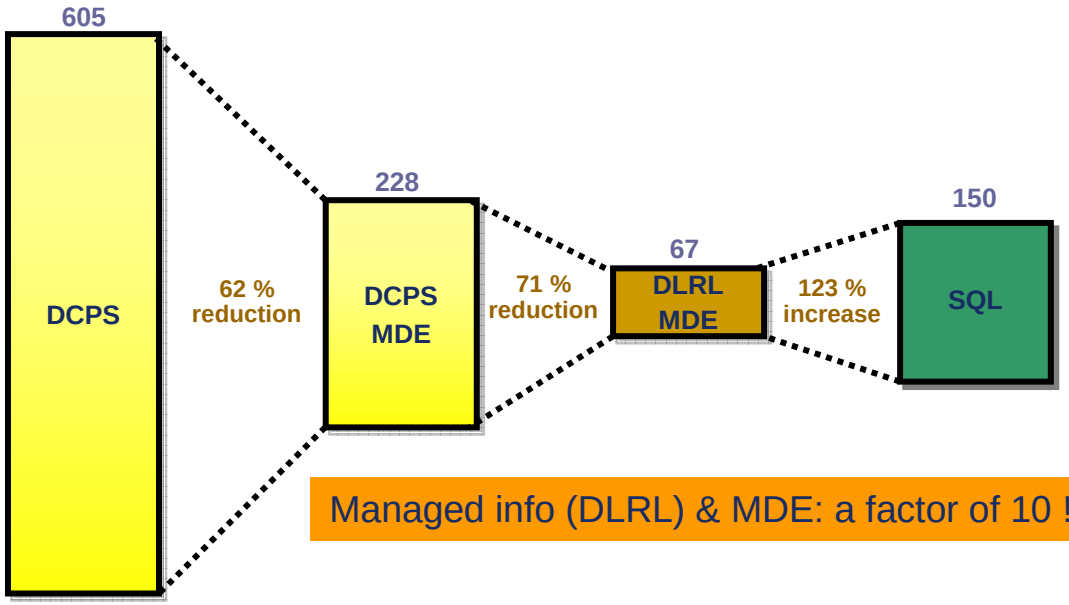
ANTs DBMS

SQL
ANTs Message-Board

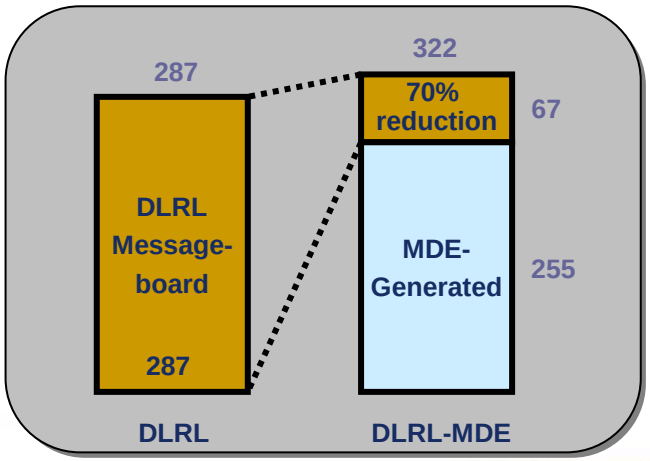
PRISMTECH
Productivity Tools & Middleware

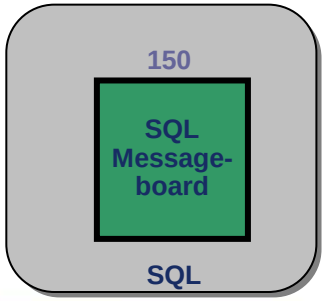
MDE and DLRL: "*Messageboard* example", Reducing Complexity !!





Managed info (DLRL) & MDE: a factor of 10 !

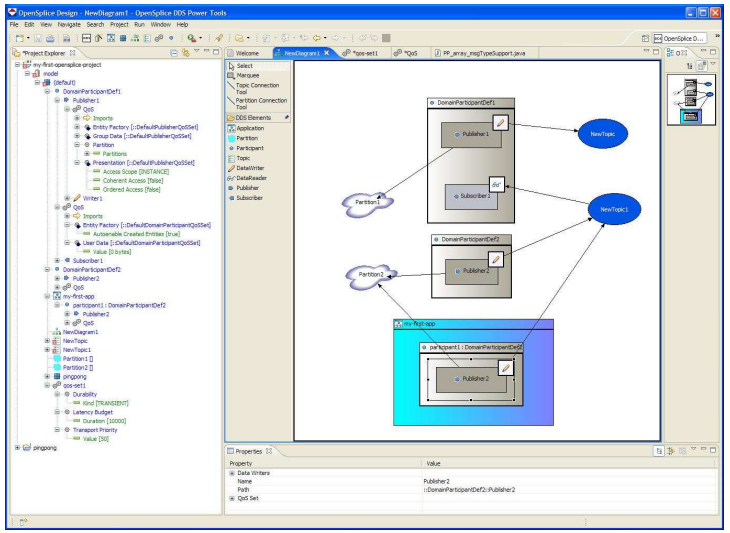




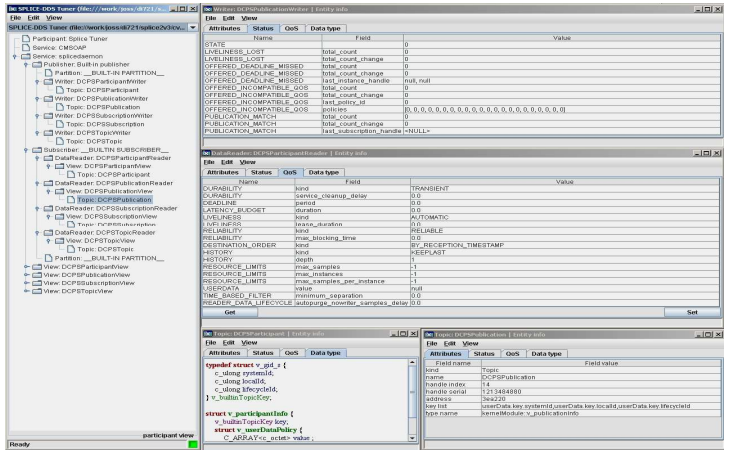
PRISMTECH
Productivity Tools & Middleware

Eclipse-based MDE tool-suite: Summary & Advantages

- **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 DDS-configuration
  - Service configuration (networking, durability)
  - **Runtime control (& round-trip engineering) by OpenSplice Tuner™ Eclipse-plugin**

PRISMTECH
Productivity Tools & Middleware

….QUESTIONS….
?????

- THANK YOU !!