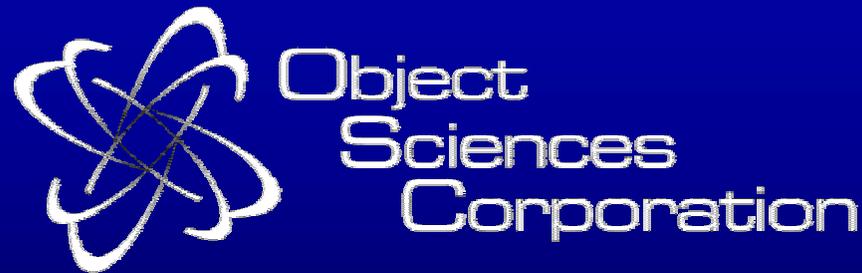
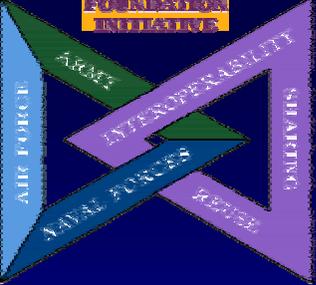


IKE2—Implementing the Stateful Distributed Object Paradigm

J. Russell Noseworthy

j.russell.noseworthy@objectsciences.com

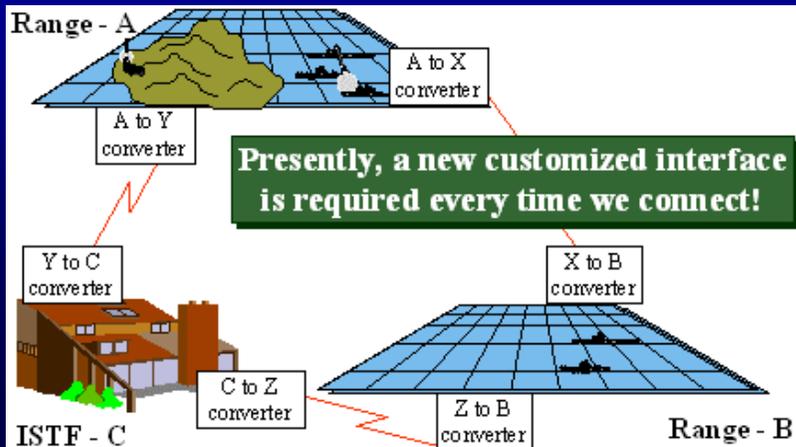
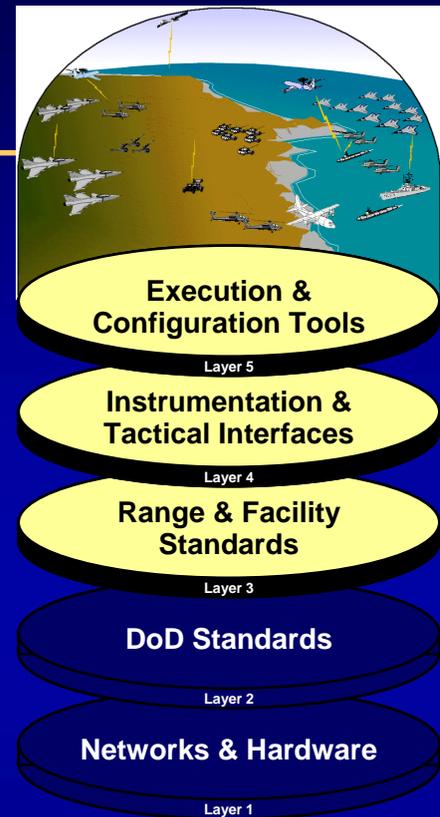




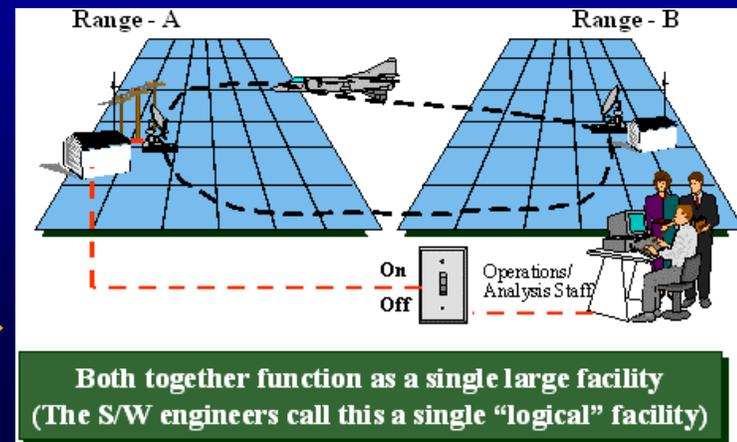
FI 2010, TENA, & IKE 2

How Do They Interrelate?

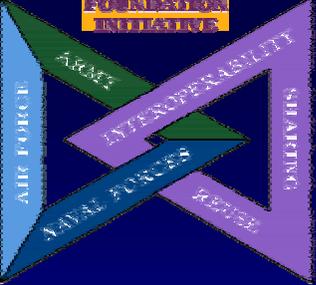
- Foundation Initiative (FI) 2010 is a DOT&E / CTEIP program whose mission is to provide products to:
 - ◆ Enable interoperability among ranges, facilities, and simulations in a quick, cost-efficient manner, and
 - ◆ Foster reuse for range asset utilization and for future instrumentation developments
- Central Product of FI 2010:
 - ◆ Test and Training Enabling Architecture (TENA) to be utilized at test and training ranges to promote interoperability between centers and to foster reusability of assets among centers (<http://fi2010.jcs.mil/>)
 - ◆ IKE 2 is the Middleware Component of TENA



← Today

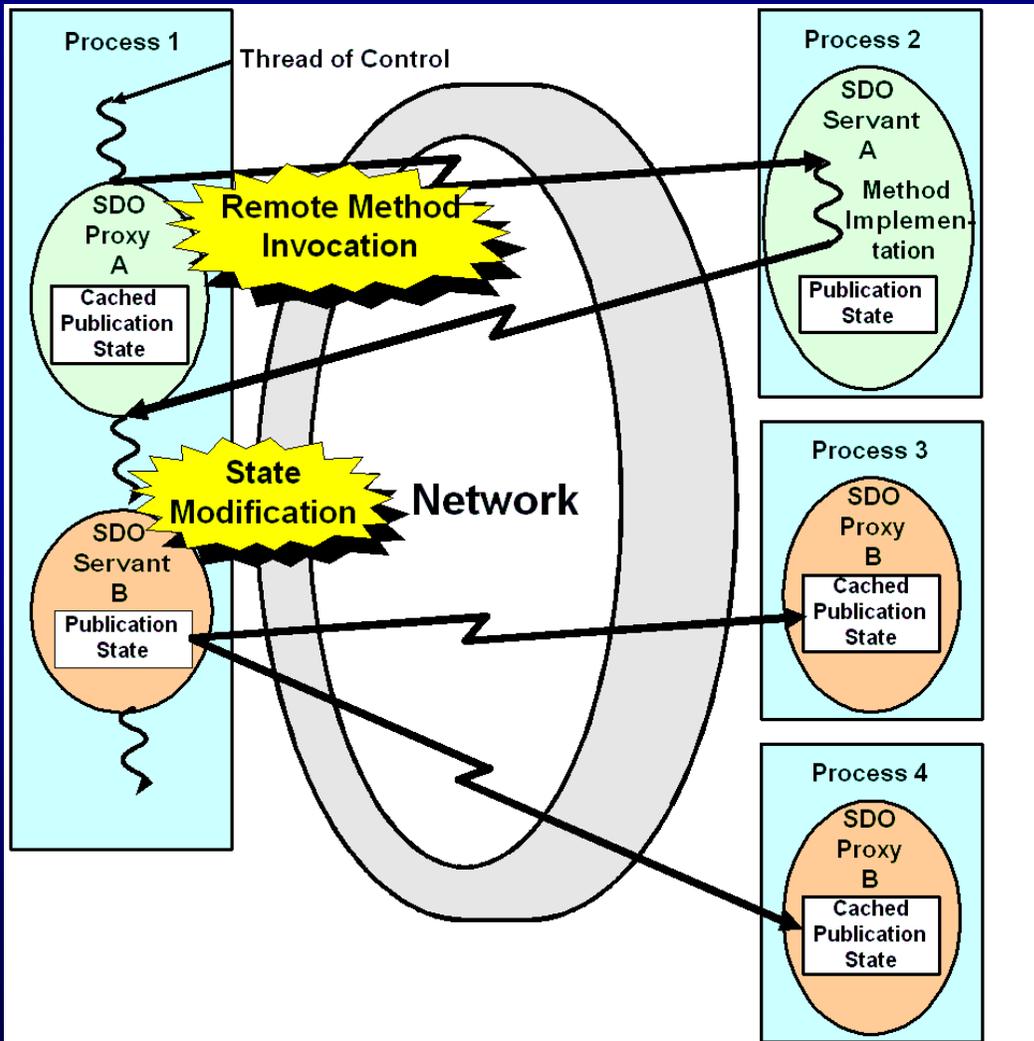


With IKE 2 →

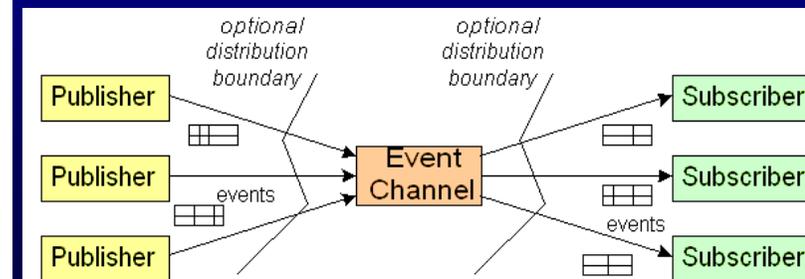


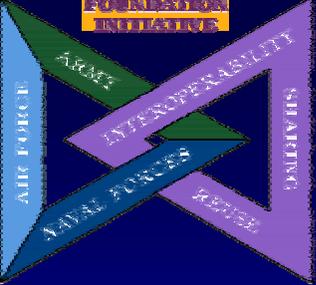
What is an SDO?

Stateful Distributed Object (SDO)



- **Object Interface**
 - ◆ Location-transparent interface to methods
 - ◆ Just like a CORBA interface
- **Publication State**
 - ◆ Data that is disseminated from servants to proxies
 - ◆ Dissemination done via publish/subscribe
 - ◆ Publication state can be read as if it were local data
 - ◆ Publication state is observable

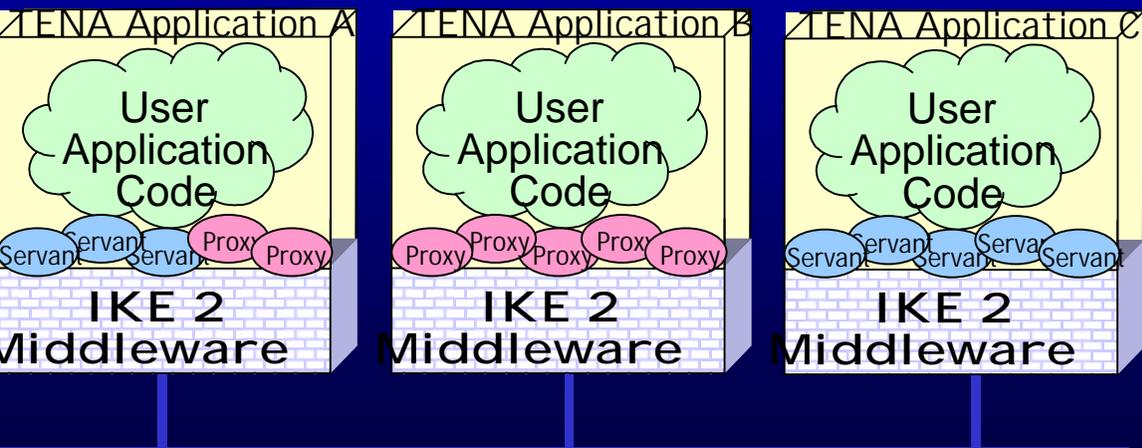
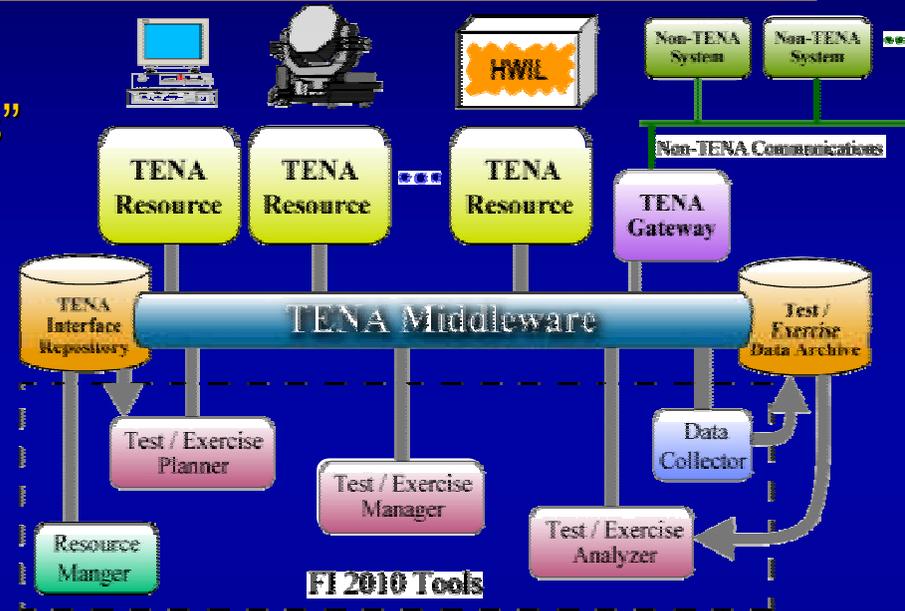




What is IKE 2?

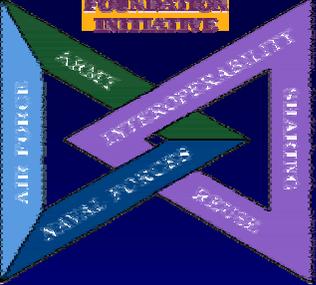
Middleware that Implements SDOs

- IKE 2 is middleware connecting distributed “logical range resources”
- IKE 2 Combines:
 - ◆ Peer-to-peer Paradigm
 - ◆ Publish/Subscribe Paradigm
 - ◆ Distributed Object Paradigm
 - ◆ Observable Distributed Shared Memory Paradigm



■ IKE 2 applications may have:

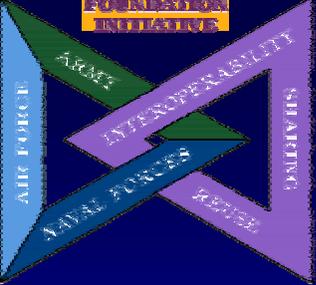
- ◆ Diverse latency & throughput needs
- ◆ Hard real-time needs
- ◆ Quality-of-Service needs
- ◆ Security needs
- ◆ Large geographical separations



SDOs are Described in TDL

What is TDL and Why Use It?

- TDL leverages existing commercial standards as much as possible
 - ◆ OMG IDL provides a solid foundation upon which to build
 - ▣ Stand on the shoulders of giants
 - ◎ OMG has been developing IDL for 12 years
 - ◆ Plan to utilize XML and UML/XMI
 - ▣ Take advantage of large (and growing) number of COTS tools
- TDL introduces constructs not directly supported in IDL
 - ◆ Class vs. interface
 - ▣ Interface has only methods, no publication state
 - ◆ Extends vs. implements
 - ▣ Single inheritance for classes; multiple inheritance for interfaces
 - ◆ Publication state
 - ▣ Attributes that are disseminated to subscribing IKE 2 applications
 - ▣ Client applications notified with publication state is modified
- TDL used to auto-generated C++ and IDL



Example TDL & IDL

IDL Auto-generated from TDL

■ Example.TDL

```
package Example
{
  interface Controllable
  { string initialize(); };

  class Participant
    : implements Controllable
  { string name;
    long ID; };

  class Sensor
    : extends Participant
  { string state;
    string point(in double azimuth)
  };

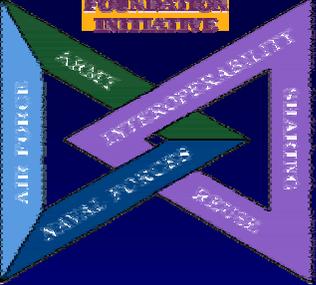
  class Platform
    : extends Participant
  { double fuel;
    Sensor longRangeSensor;
    void move(in double x);
  };
};
```

■ Example/Platform.IDL

```
#include <Participant/Participant.idl>
#include <Sensor/Sensor.idl>

Module Example
{
  module PlatformIDL
  {
    valuetype PublicationState
      : ParticipantIDL::PublicationState
    {
      public double fuel;
      public SensorIDL::PublicationState
        longRangeSensor;
    };

    interface Servant
      : Participant::Servant
    {
      void move(in double x);
    };
  };
};
```



IKE 2 Implementation

C++ Implementation in Testing Today

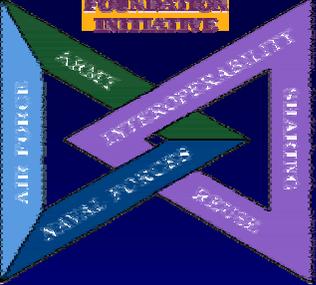
- SDOs support inheritance from other SDOs
 - ◆ Subscribers to a base class will receive (sliced) data from derived classes
- SDOs support **containment** of other SDOs
 - ◆ Subscribers to SDOs contained in other SDOs received the contained data
- IKE 2 is implemented using ACE+TAO
 - ◆ IKE 2 uses DIME to provide data dissemination
 - ◆ DIME uses TAO's Real-Time Event Channel
- IKE 2's 2nd beta release has been made
- IKE 2 **may** be available for your (government) use
 - ◆ Contact me
 - ◆ Not sure about commercial use at this time



IKE 2 Uses DIME

What is DIME?

- DIME = Distributed Interest-based Message Exchange
- Originally developed for the RTI-NG (DoD simulation middleware) program
 - ◆ Wanted a system to facilitate experimenting with various message routing technique
- DIME is an extensible framework for Publish/Subscribe message passing systems
- Publish/Subscribe applications need a means to describe who gets what messages
- Most existing systems have only primitive means to perform this description, e.g., an integer
 - ◆ This burdens the application programmer with the hard problem of mapping integers to higher-level semantically rich descriptions
 - ▣ E.g., Tanks in Bosnia = 7, Tanks in Serbia = 8, etc.
- DIME allows the application programmer to describe who gets what messages at high level
 - ◆ The hard problems such as mapping descriptions to integers is separated from the application program

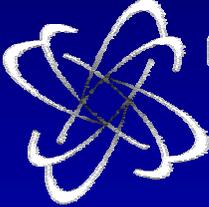


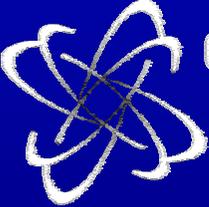
Future Work

What Future Needs are Envisioned?

- **Where Can We Get More Help from TAO?**
 - ◆ **Real-Time Support** — Many envisioned applications have hardware in the loop
 - ▣ CORBA Real-Time capabilities will be directly re-used
 - ◆ **Fault Tolerance** — ability to recover from application, computer, and network failures
 - ▣ Many envisioned applications are *mission critical*
 - ◆ **Security** — Always important to the DoD community
 - ◆ **Scalability** — Some applications may be involved hundreds of nodes
 - ◆ **Heterogeneous Communication** — support simultaneous use of different communication mechanisms (e.g., IP, shared memory) and associated routing and hierarchy issues

Acknowledgements

-  Object Sciences Corporation is grateful to  SAIC, An Employee-Owned Company, the prime contractor for the work presented here

-  Object Sciences Corporation is also grateful to the FI 2010 program office for making it possible to attend this conference