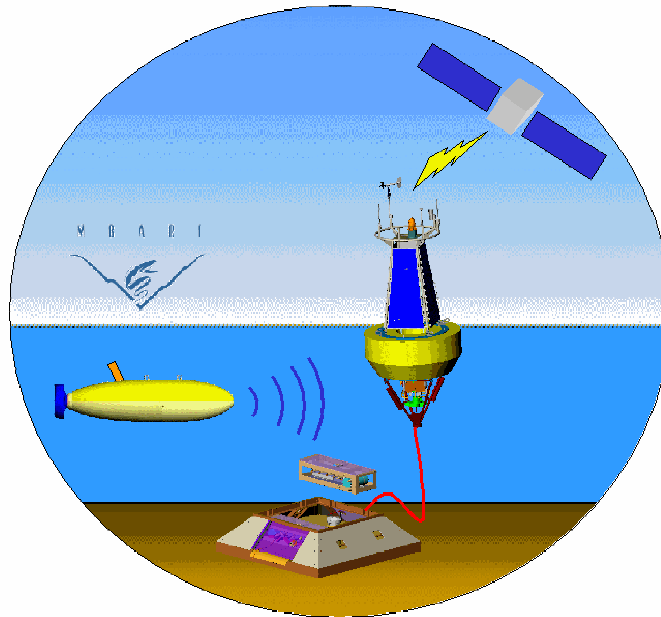
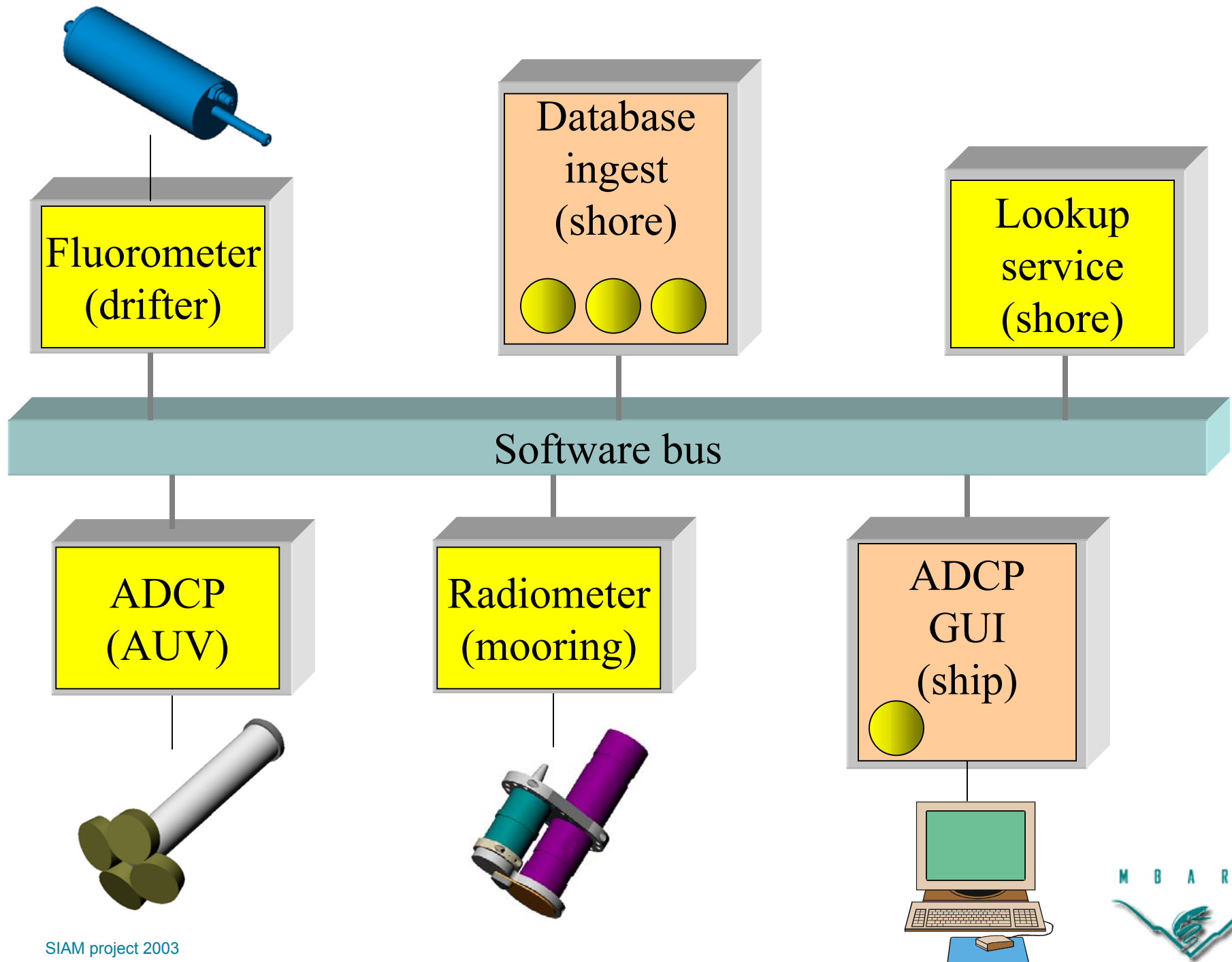


Interoperation of devices within the MOOS network



Tom O'Reilly
January 17, 2003



Software bus protocols

- JRMP
 - RMI API
 - Java only
- IIOP
 - RMI or CORBA IDL API
 - Allows interoperation of Java, C++ (C?)

Software bus protocols

- SOAP
 - Can be used for C++, Java, C#...
 - ASCII-based XML protocol
 - High latency (~25-35X JRMP latency)
 - High data overhead (~40X JRMP data volume)

JRMP vs IIOP: Data volumes

	IIOP TCP/IP layer	JRMP TCP/IP layer	IIOP PPP layer	JRMP PPP layer
Proxy download	1194	2782	511	937
emptyTest()	240	377	105	178
primitiveTest()	268	403	109	176
structTest()	1261	1571	151	225
Shutdown	156	416	105	141

IIOP: RMI vs CORBA IDL

	RMI	CORBA IDL
Inheritance	RMI interfaces and any passed objects can be subclassed	IDL interfaces can be subclassed; IDL structs can not be subclassed
Objects-by-value	Yes	Yes, but...
Java - C++/C interoperability	Yes, but...	Yes
Ease of programming	Fairly simple (see example)	Extremely verbose (see example)

Proposed protocol and API

- Java RMI-IIOP will be “standard” way to distribute objects
 - Easy to pass objects-by-value
 - Subclassing of distributed interfaces and passed objects
 - Straightforward application code
 - Non-Java components can participate through CORBA IDL

Implementing a device service

- Service must communicate with device using device protocol through back-end
- Service must implement distributed object interface to network

Non-Java service hosts: options

- Delegate distributed object service to an accessible Java host (*surrogate approach*)
 - OR
- Implement service using CORBA IDL

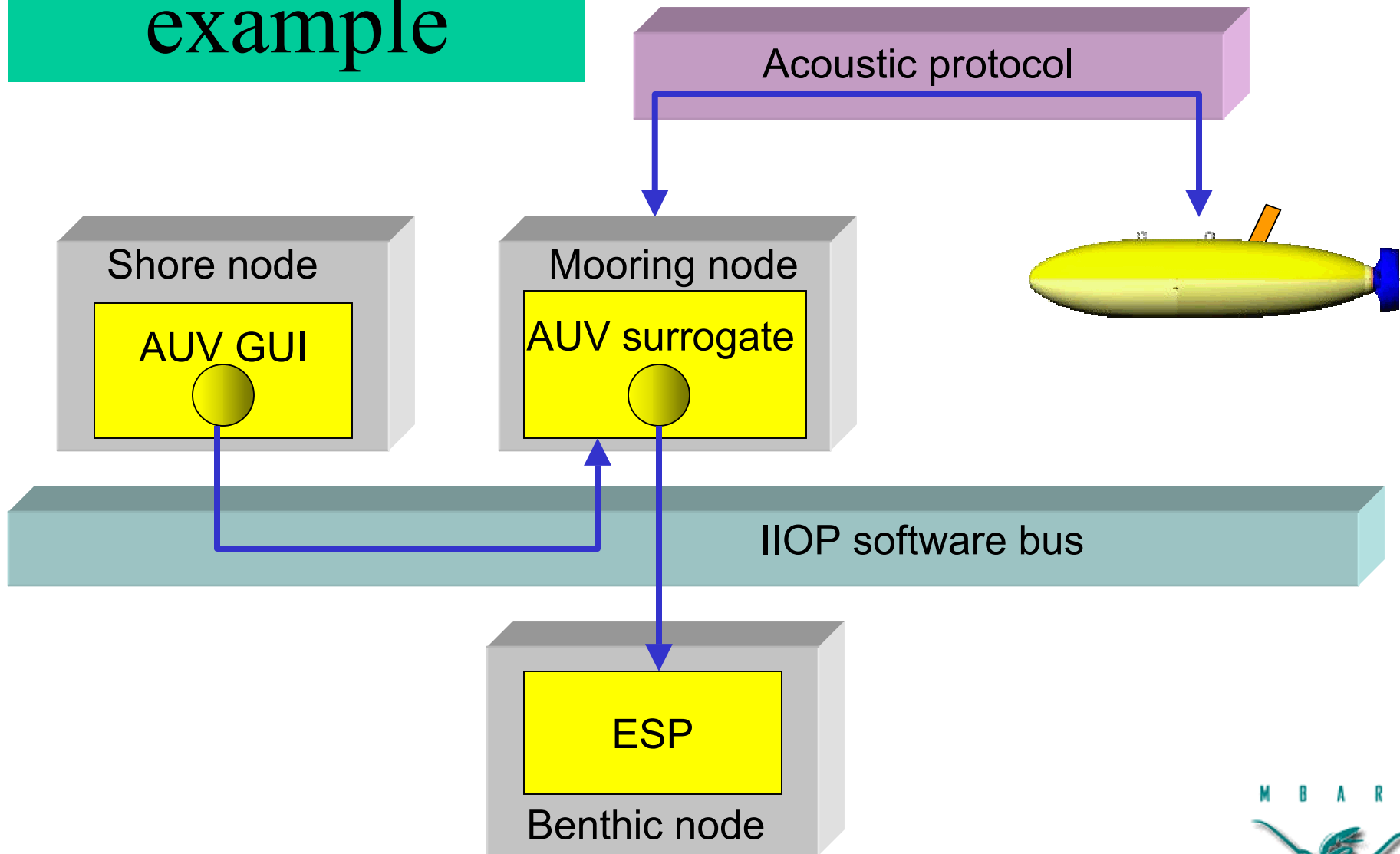
Surrogate approach

- Surrogate runs on protocol-enabled host, interacts with network on behalf of device
- Surrogate uses two connections/protocols:
 - Connection to network, software bus protocol
 - Connection to device, device-specific protocol

When is surrogate useful?

- Device host cannot run Java
- Communication link cannot support IIOP
- Legacy device integration
 - No need to change existing device code!

Surrogate example



CORBA Service

- CORBA clients can be created using IDL generated by rmic
- Java clients can use CORBA services using Java IDL

CORBA Client Example

Create a RMI Service

```
// Step 1: Instantiate the CTD Service
```

```
CtdImpl ctdImpl = new CtdImpl();
```

```
// Step 2: Publish the reference in
```

```
// the Naming Service using JNDI API
```

```
Context initialNamingContext = new InitialContext();
```

```
initialNamingContext.rebind("CTDService", ctdImpl);
```

CORBA Client Example

Compile Java RMI service

```
javac -d . -classpath . CtdImpl.java
```

Generate IDL using rmic

```
rmic -classpath . -iiop -idl CtdImpl
```

CORBA Client Example

- Use IDL generated with rmic to implement CORBA client
- The client can be implemented in any language supported by the ORB