



OneSAF C4I Adapter Overview

**OneSAF User's Conference
August 18, 2004**

High Level View of C4I Adapter

C4I Systems

- MCS
- ASAS
- AFATDS
- AMDWS
- CSSCS
- FBCB2



C4I Adapter

C4I Side Common

C4I System Interfaces:

- Commsserver
- Email
- SocketComms

Functionality:

- CMP
- Plugins

Translation Services

Common:

- Translation Infrastructure
- Mapper Generator
- Convertors

Uncommon:

- Mappers



Sim Side Uncommon

Simulation Interfaces:

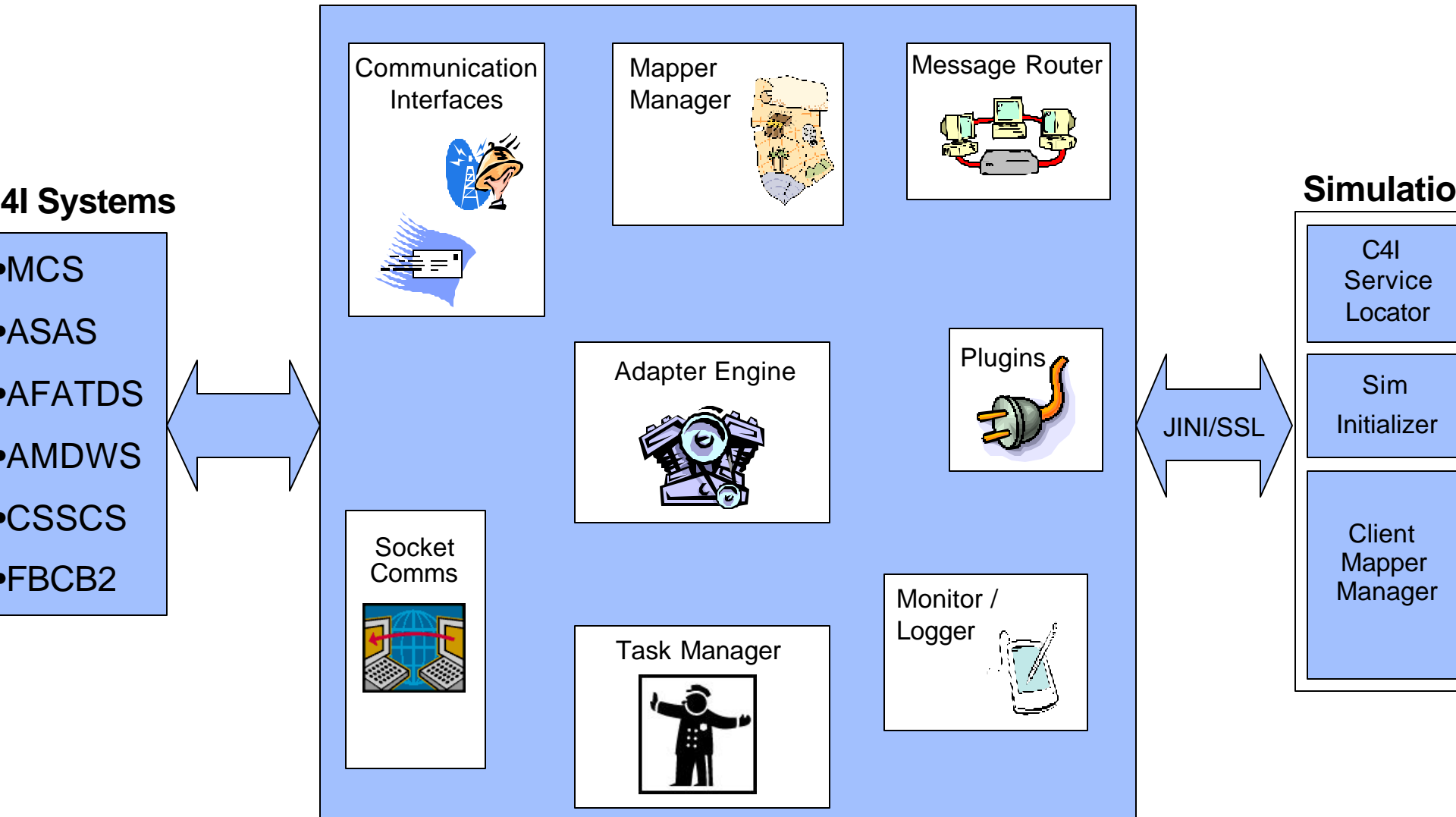
- OneSAF: ODB
- WARSIM: HLA FOM
- Initialization
- Checkpointing



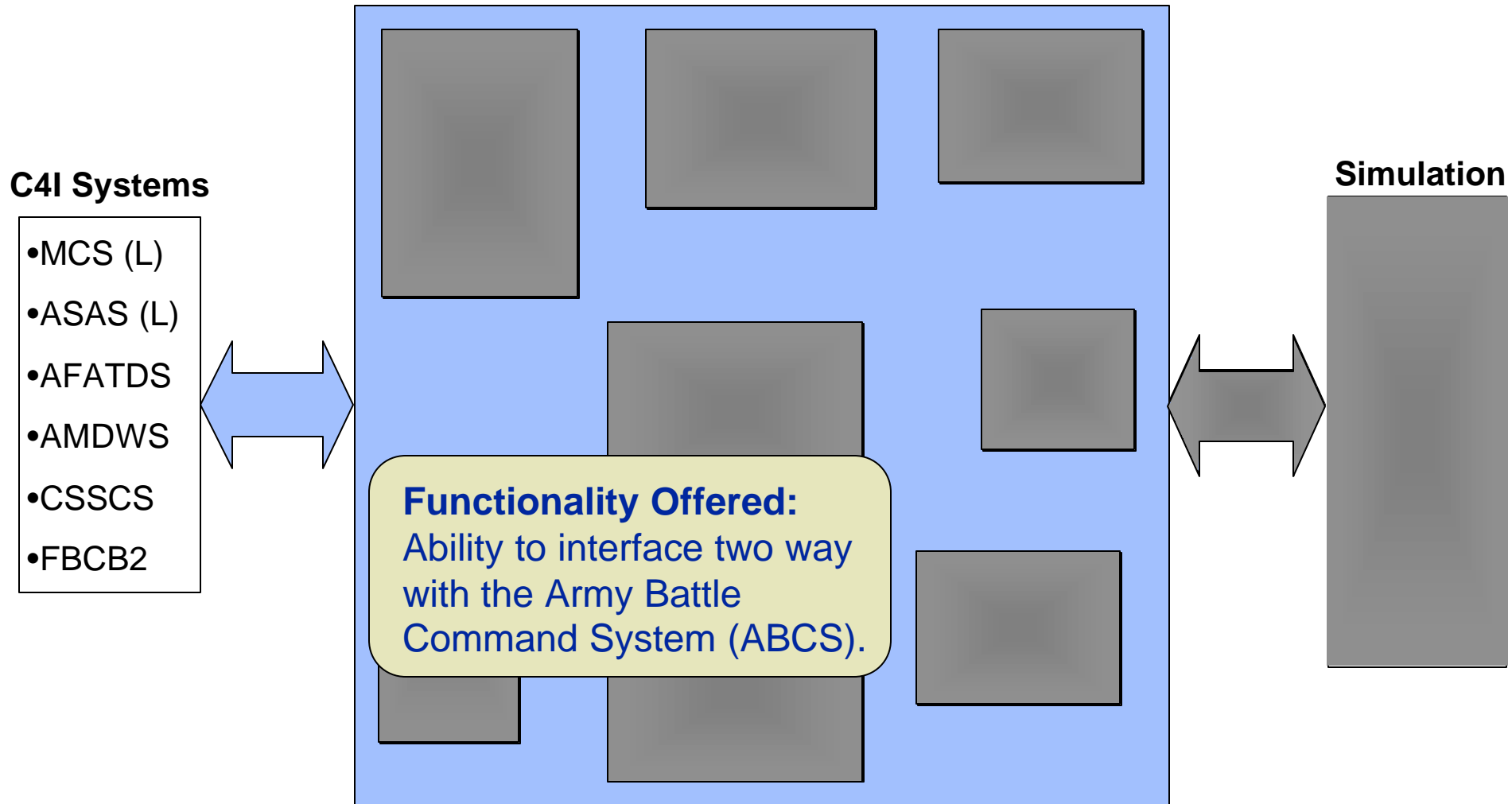
Simulations

- OneSAF
- WARSIM

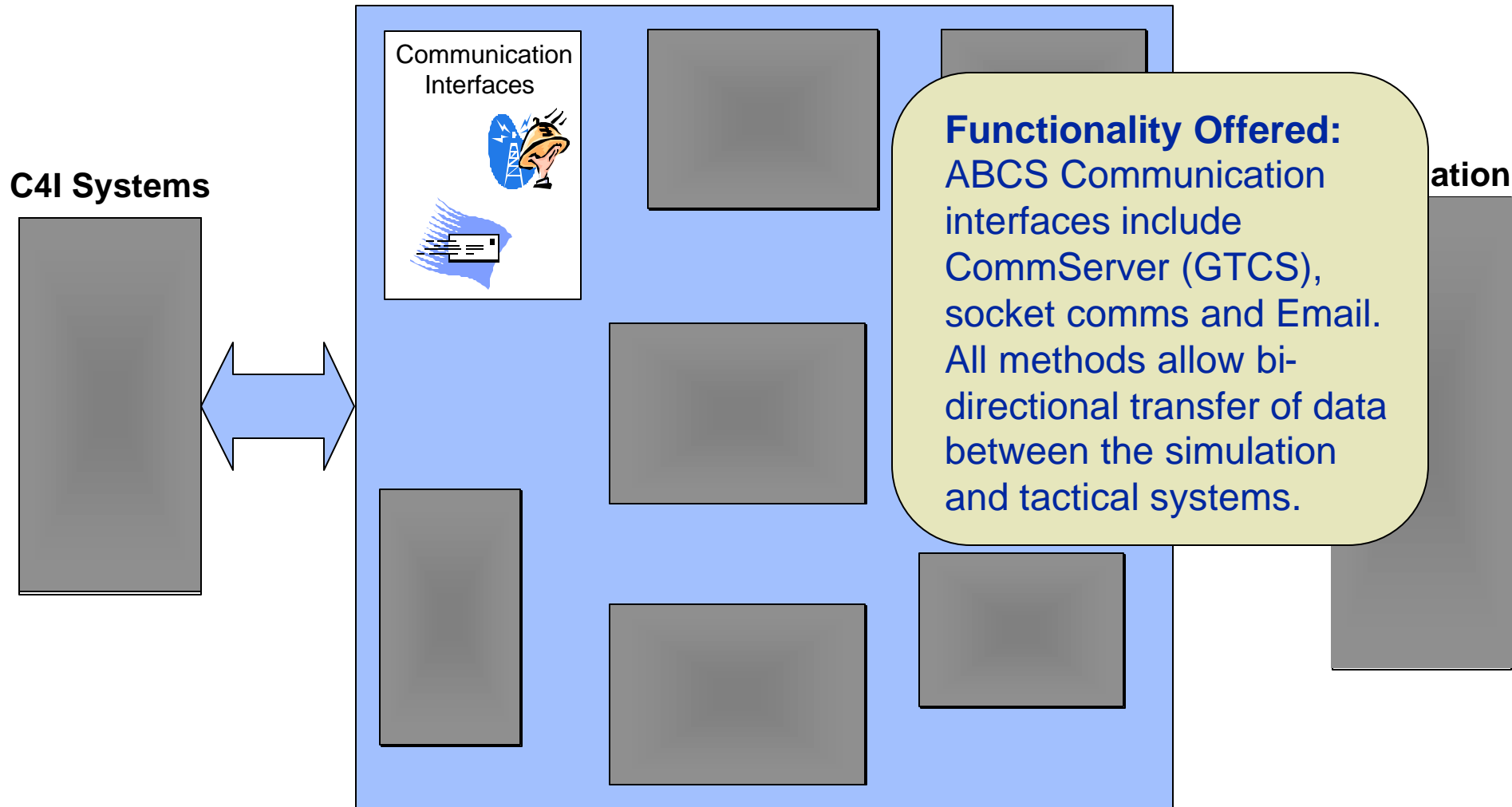
OneSAF C4I Adapter Architecture



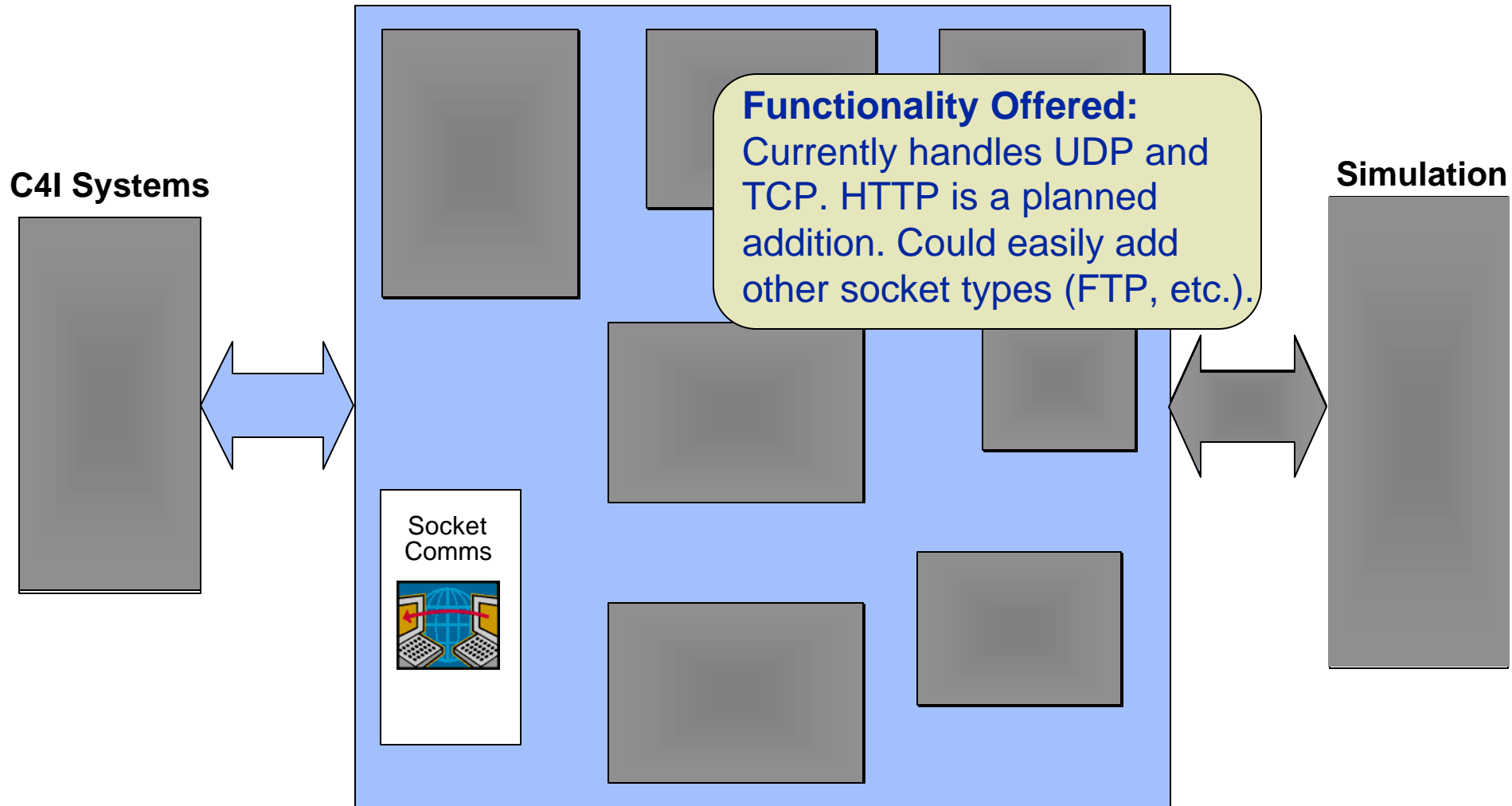
OnesAF C4I Adapter Architecture



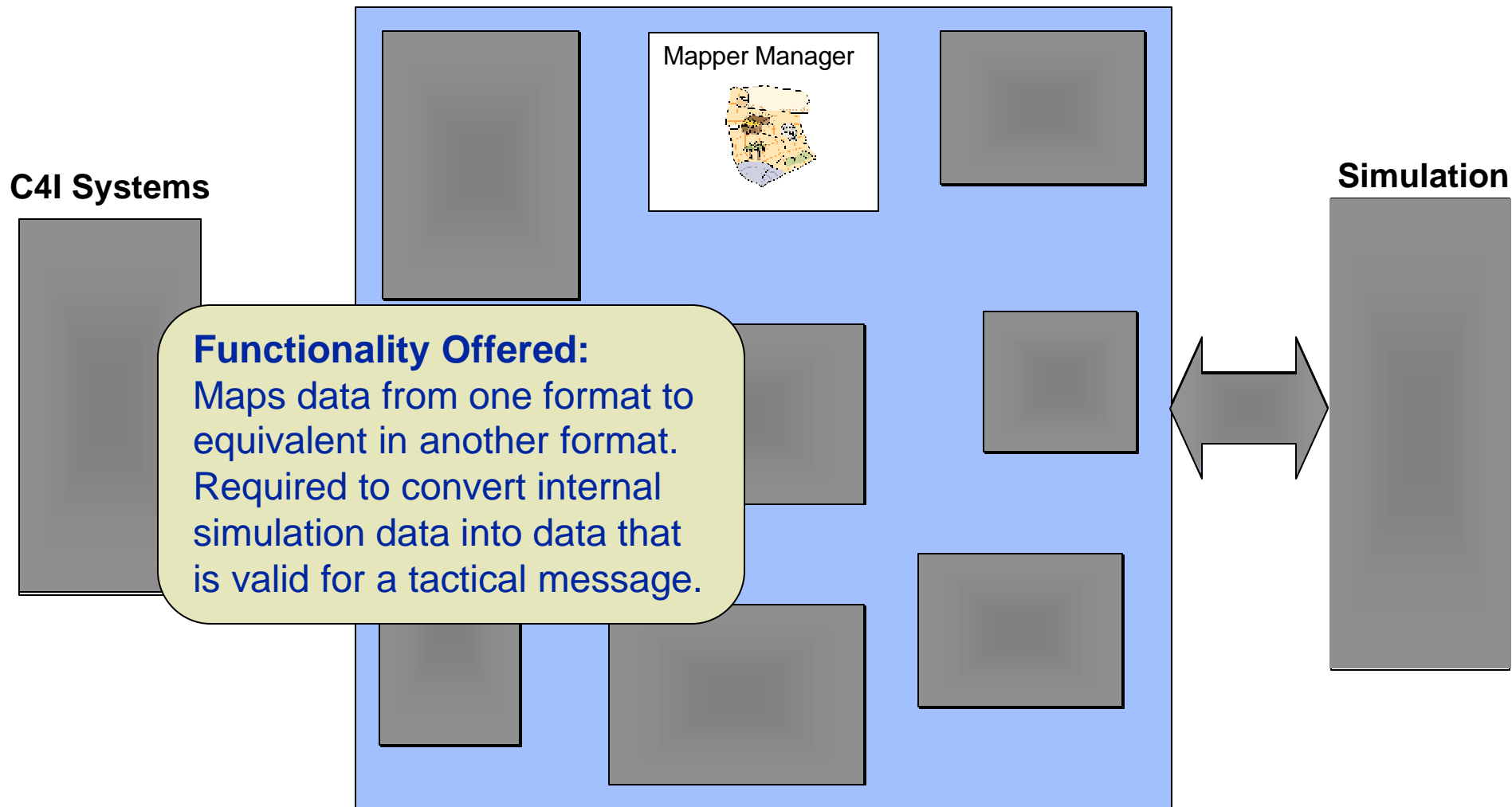
OnesAF C4I Adapter Architecture



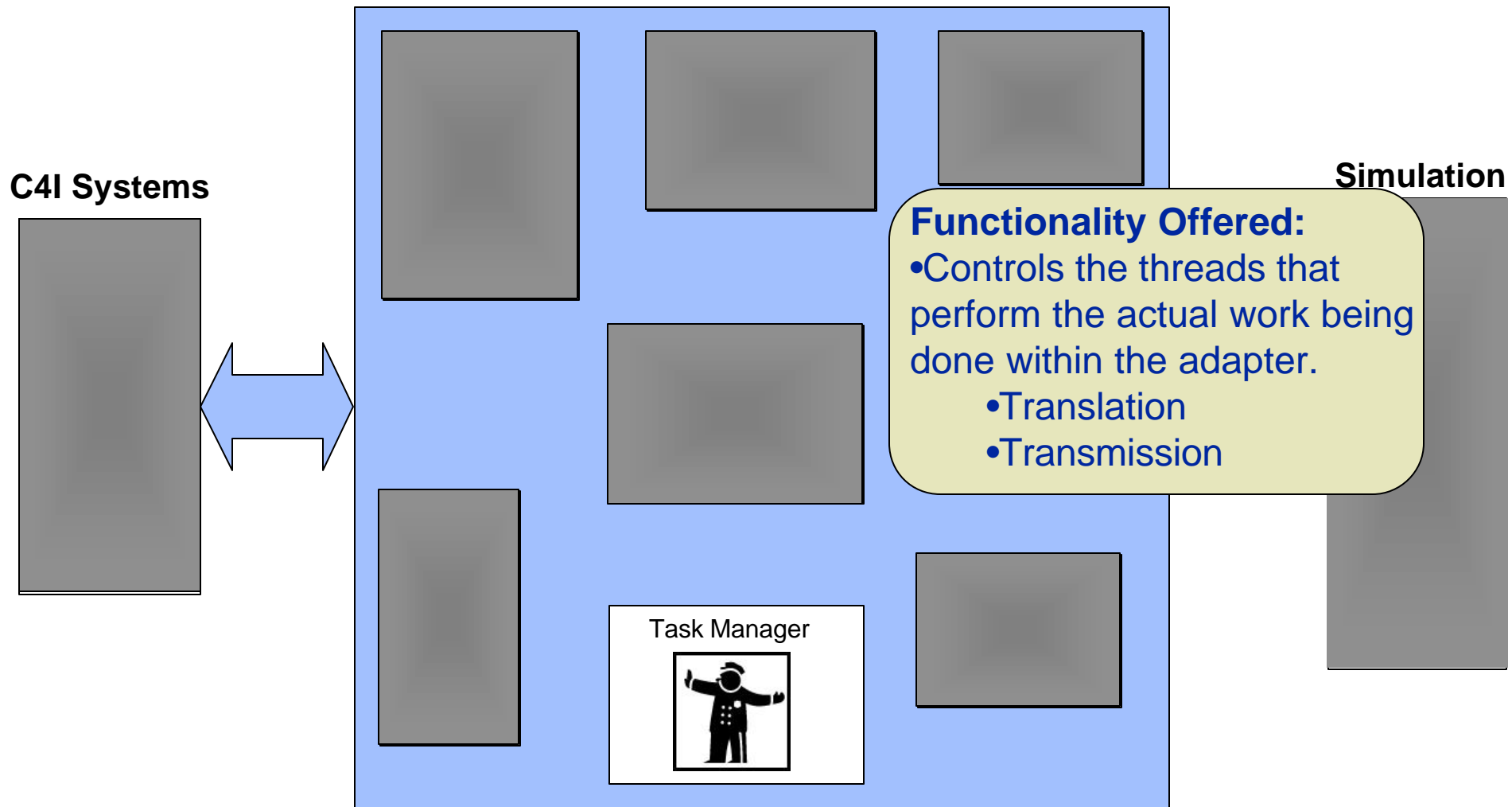
OnesAF C4I Adapter Architecture



OneSAF C4I Adapter Architecture



OnESAF C4I Adapter Architecture

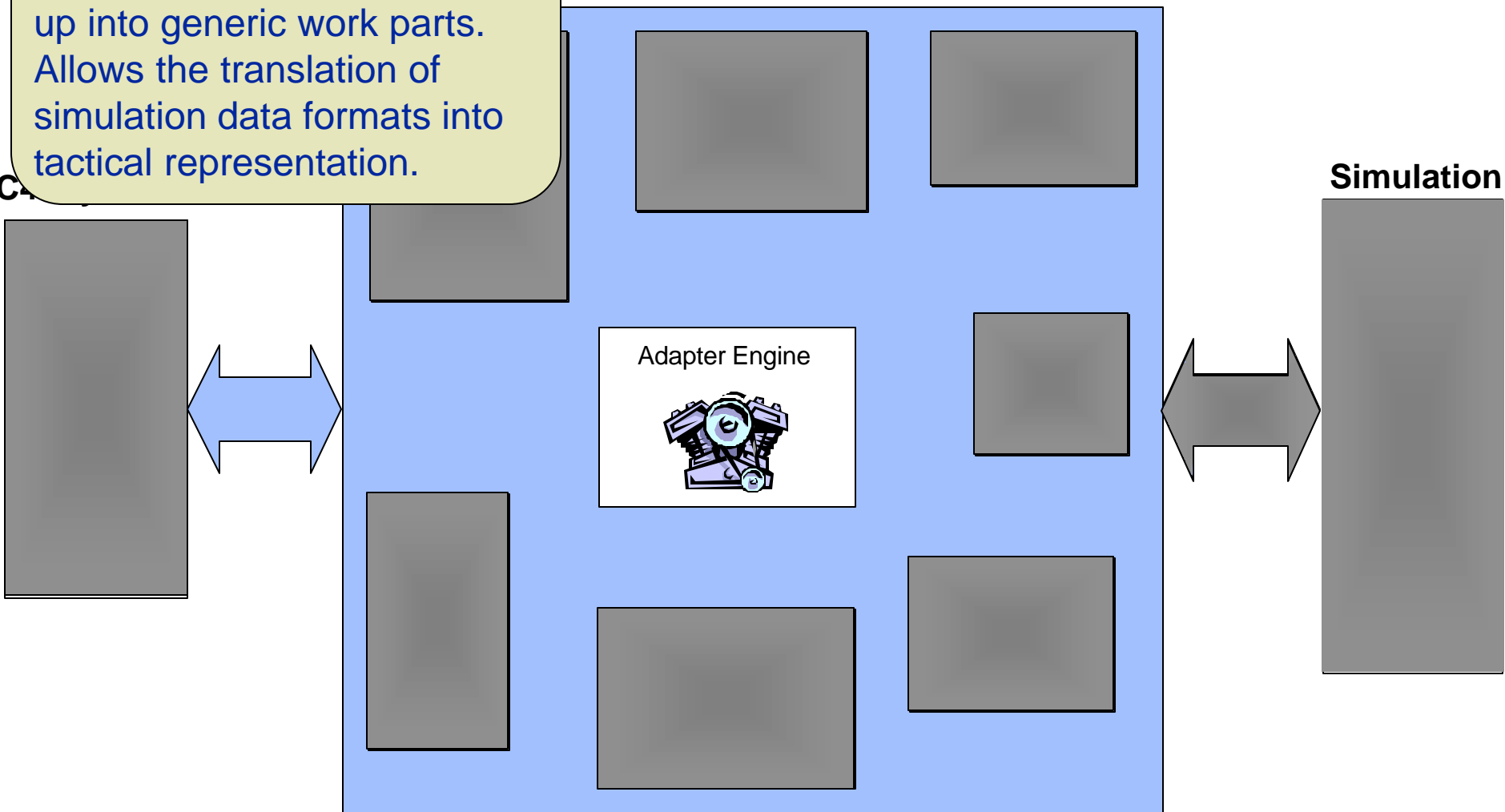


Adapter Architecture

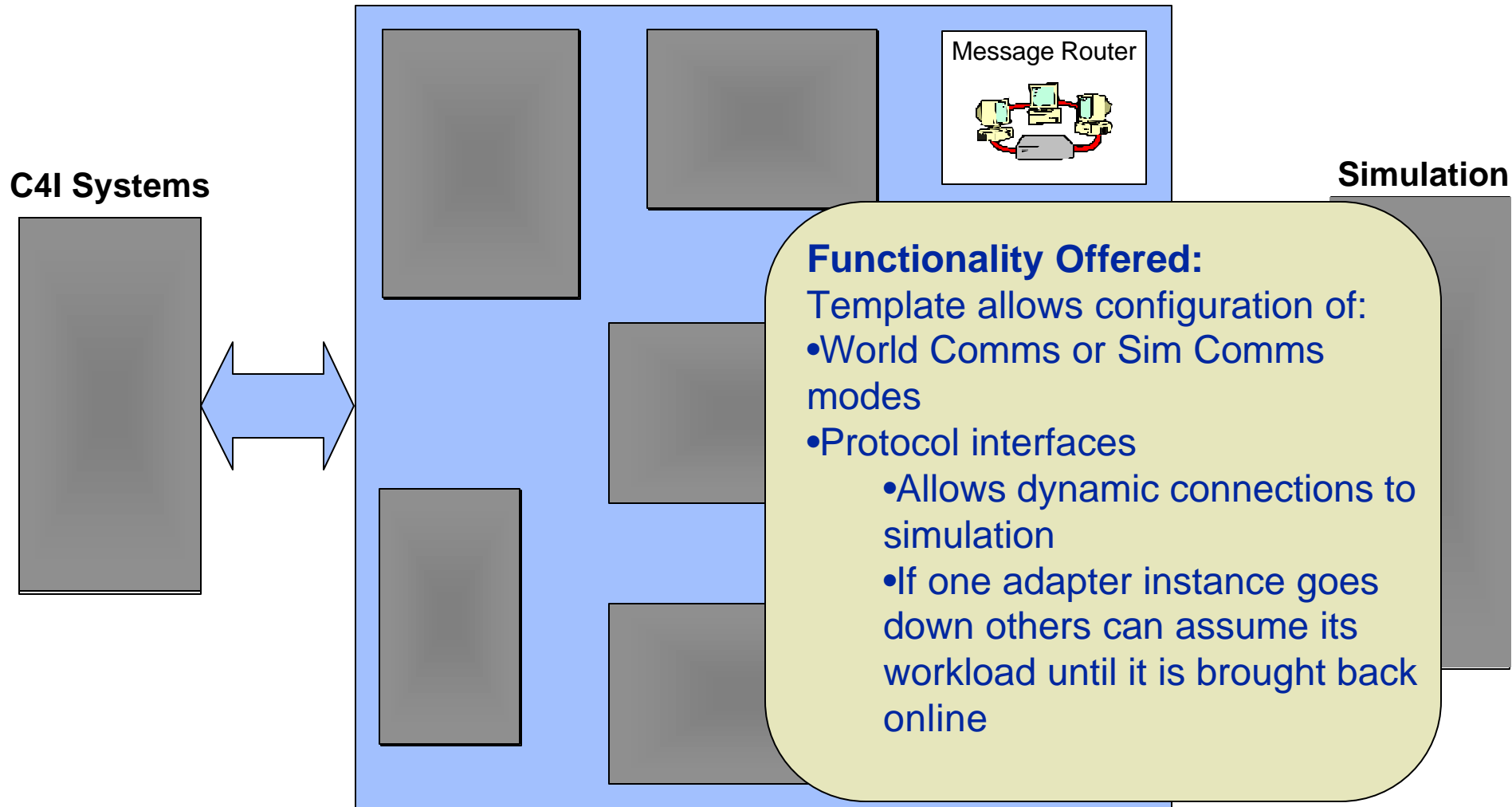
Functionality Offered:

Translation process broken up into generic work parts. Allows the translation of simulation data formats into tactical representation.

C4I

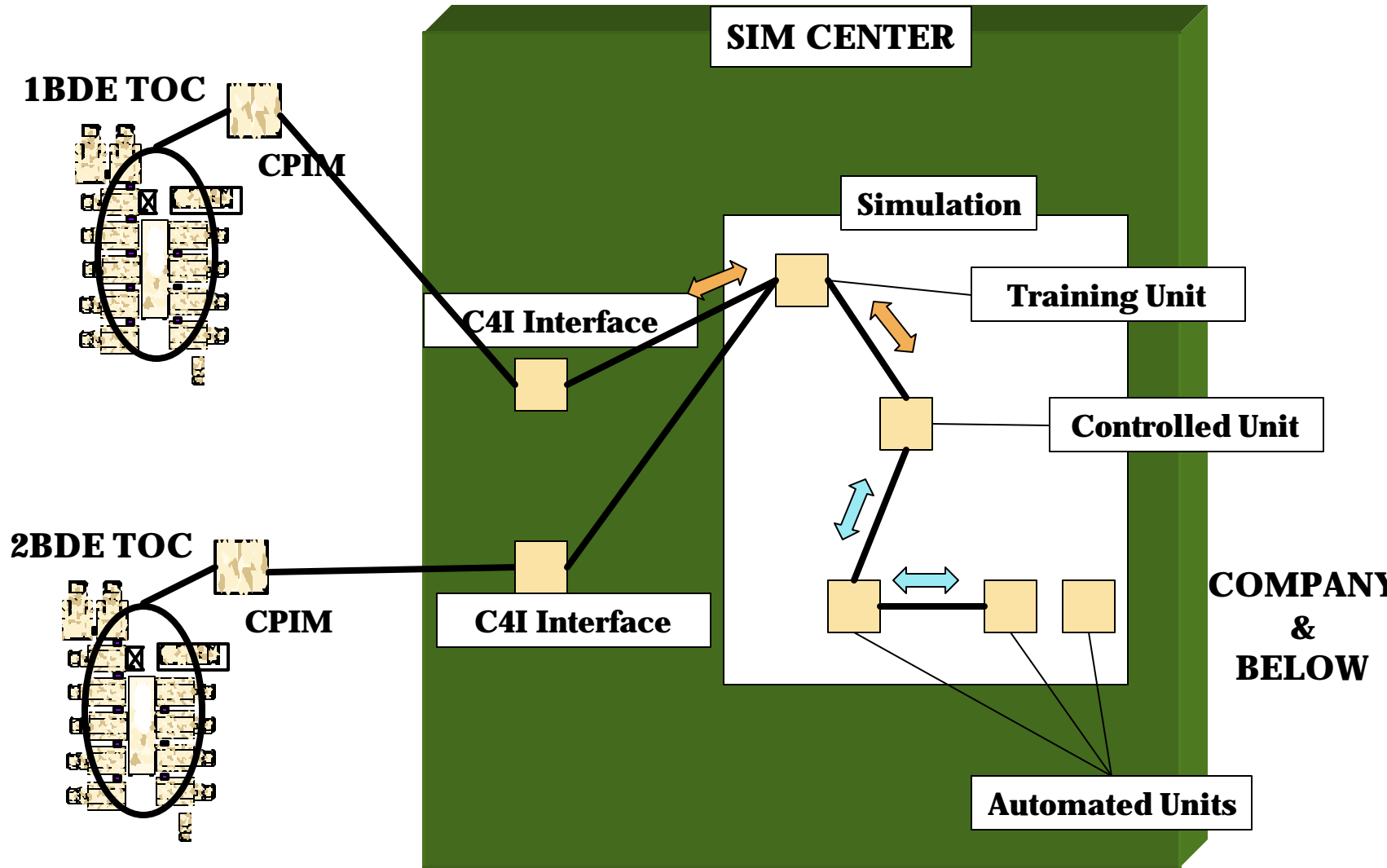


OnesAF C4I Adapter Architecture



Message Routing Schemes

Simulation Comms

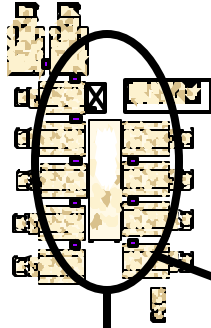


Message Routing Schemes

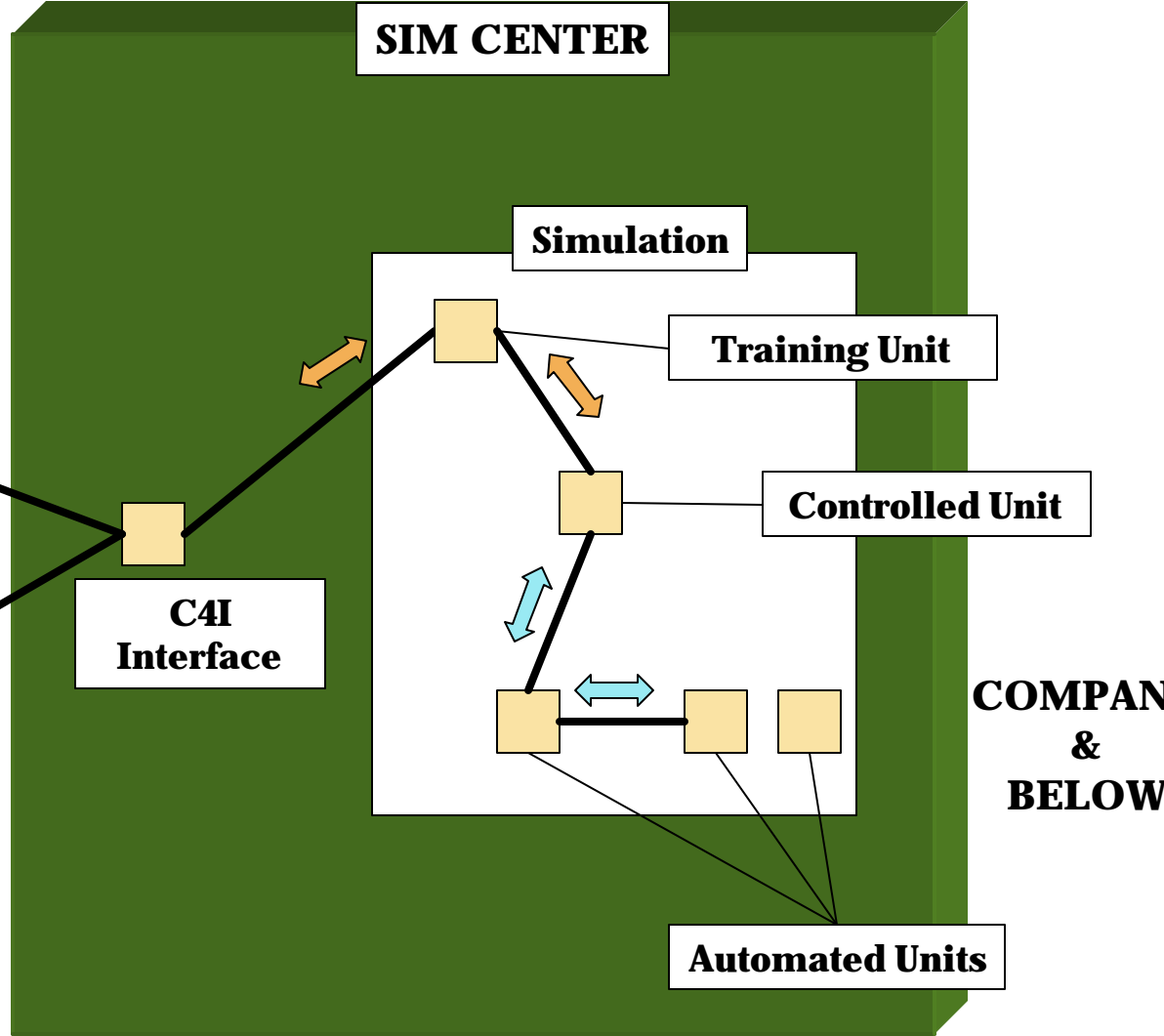
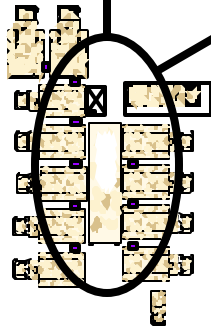
World Comms

ONE
SEM-
AUTOMATED
FORCES
FORCES

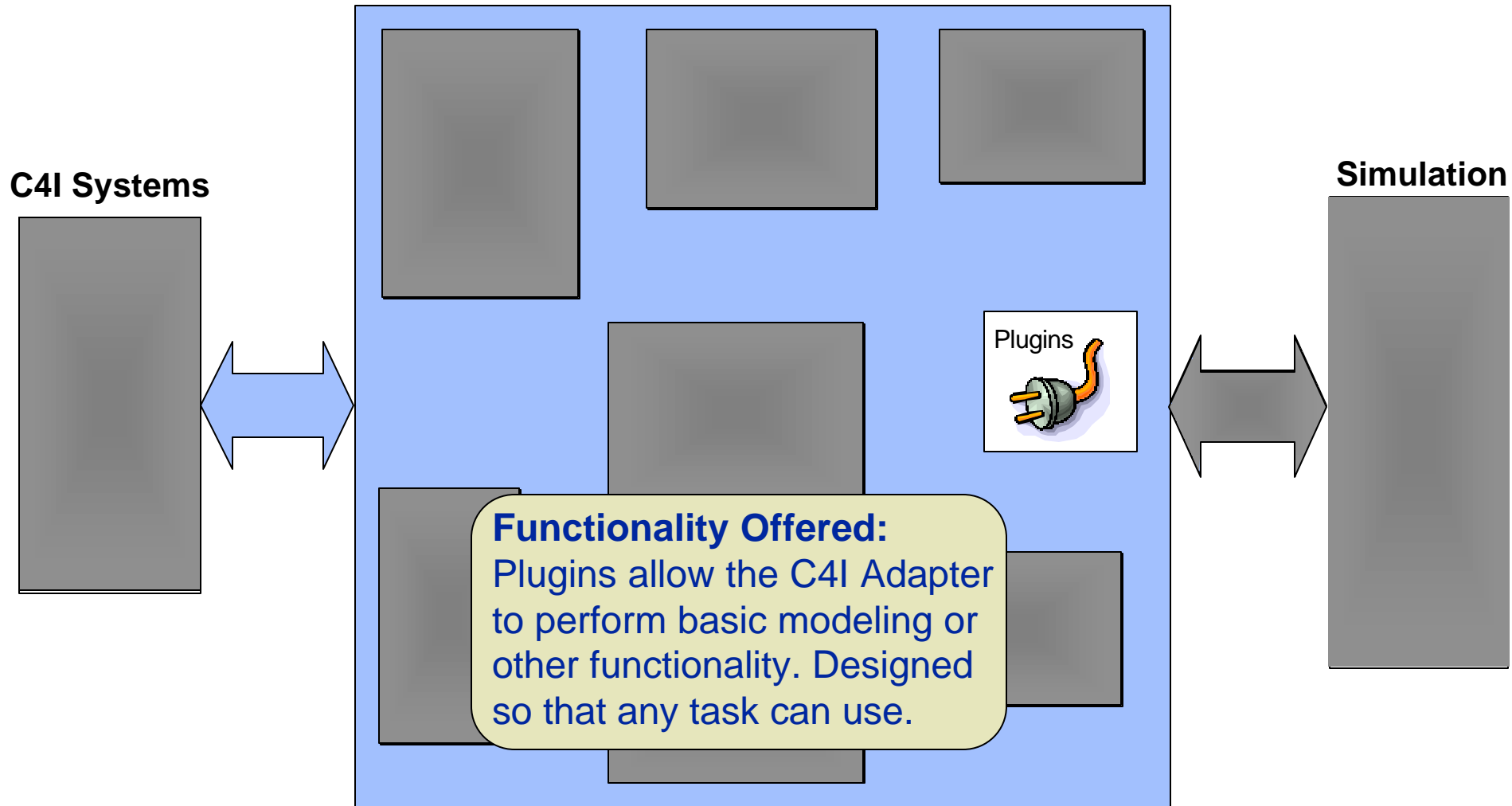
1BDE TOC



2BDE TOC



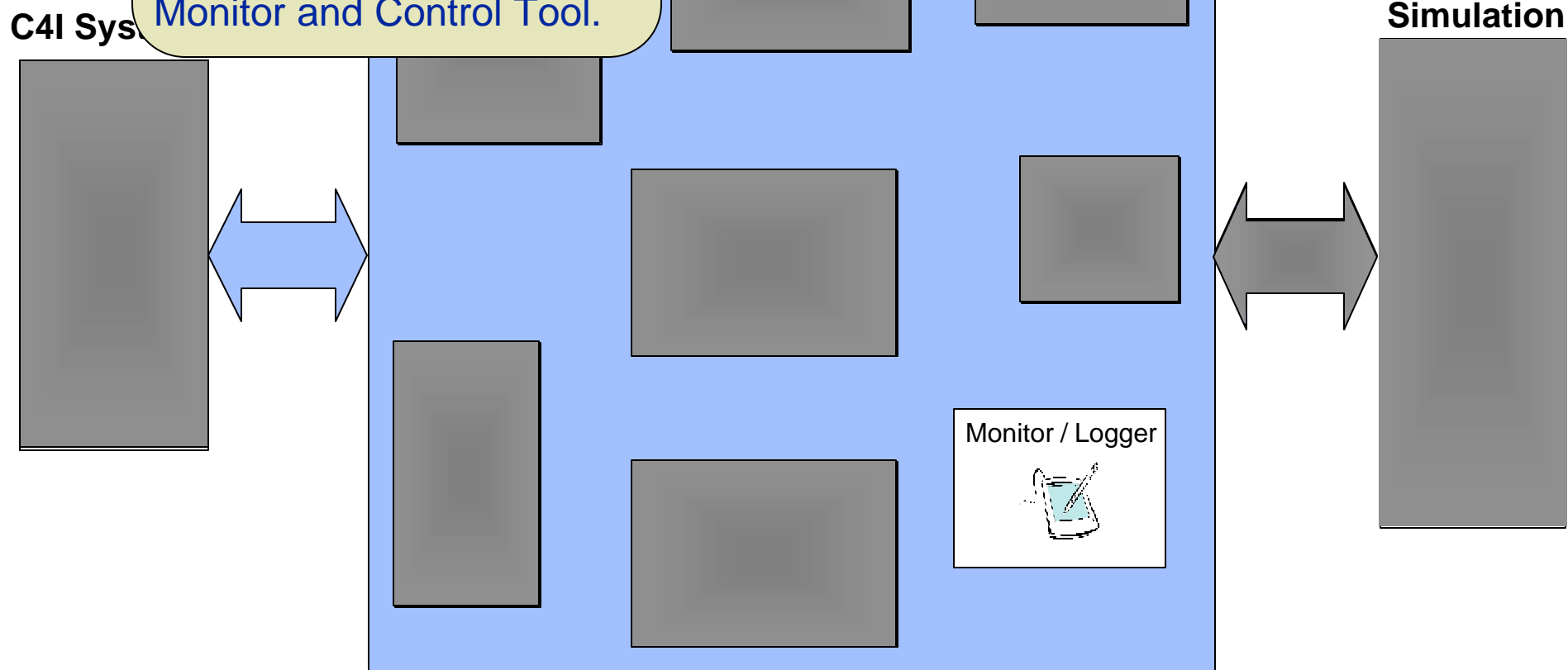
OneSAF C4I Adapter Architecture



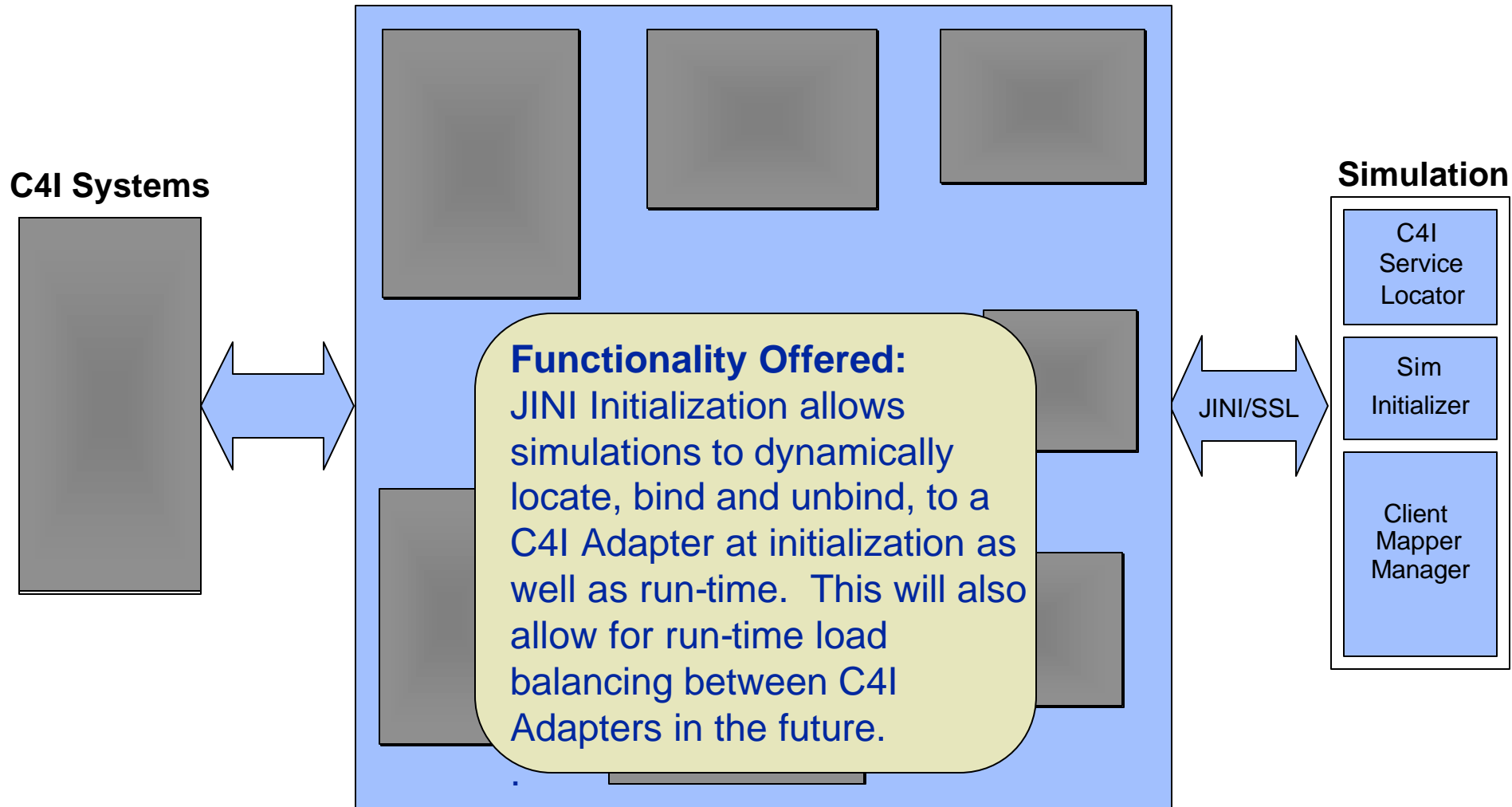
OnesAP C4I Adapter Architecture

Functionality Offered:

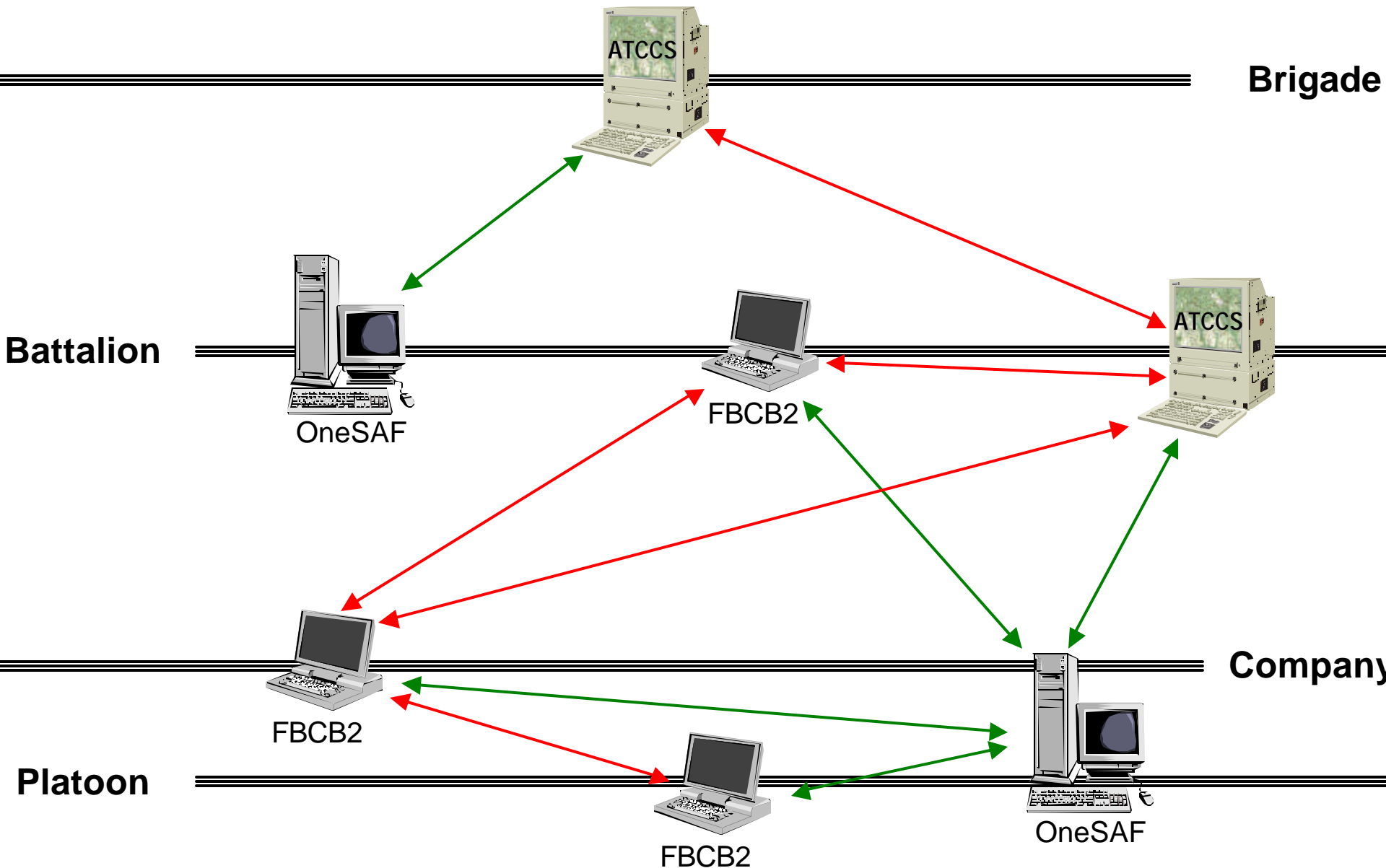
Performs performance monitoring and logging of the adapter. Allows remote control of the adapter via the Monitor and Control Tool.



OnesAF C4I Adapter Architecture



OneSAF C4I Interoperability Operational View



→ OneSAF Stimulation
→ Live Stimulation

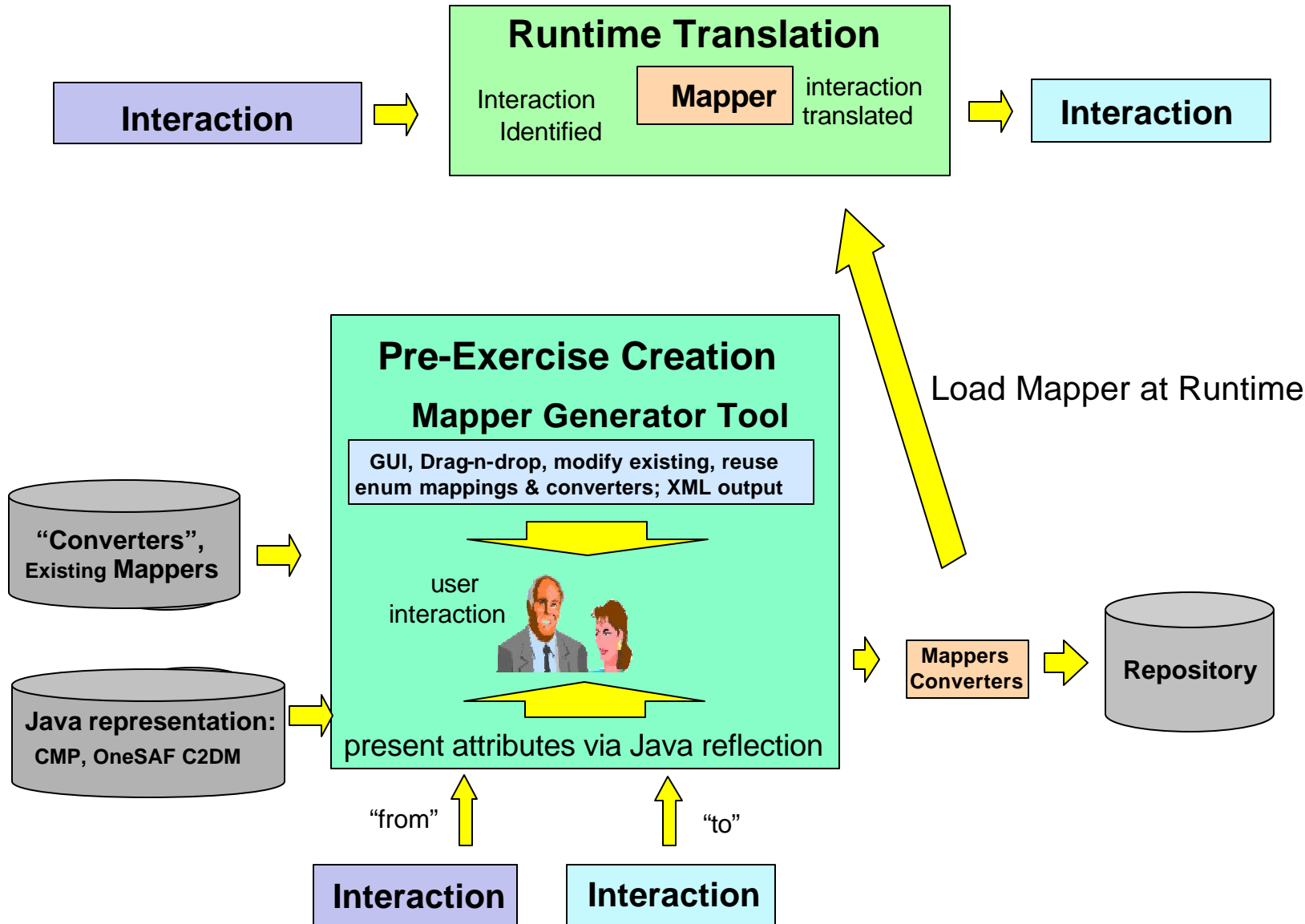
OneSAF FOC Target ABCS 6.3.6

System	Connection	Protocol	Message Format
MCS	Email	SMTP, POP3	Planner Attachments (XML files)
	CommServer	TCP/IP	USMTF2000, JVMF (FBCB2 DCX2)
MCS-Light	CommServer	TCP/IP	USMTF2000, JVMF (FBCB2 DCX2)
CSSCS	Email	SMTP, POP3	Planner Attachments (xml files)
	CommServer	TCP/IP	USMTF2000, JVMF (FBCB2 DCX2), CSSCS stovepipe USMTF-like
ASAS-RWS	Email	SMTP, POP3	Planner Attachments (xml files)
	CommServer	TCP/IP	USMTF2000, JVMF (FBCB2 DCX2)
ASAS-Light	CommServer	TCP/IP	USMTF2000, JVMF (FBCB2 DCX2)
AFATDS	Email	SMTP, POP3	Planner Attachments (xml files)
	CommServer	TCP/IP	USMTF2000, JVMF (FBCB2 DCX2), AFATDS stovepipe VMF Package 11
FAADEO	Socket	UDP/IP	FDL
FBCB2	Socket	UDP/IP, TCP/IP	JVMF (FBCB2 DCX2)

ONE
SEM-
AUTOMATED
FORCES

0
1
2
3
4
5
6
7
8
9

Mappers: Development and Execution



OneSAF C4I Mapping Tool

- Graphical Interface that allows non-programmers to visualize and “map” between disparate types of data
 - Easy to use drag and drop capabilities
- Supports reusable converters that can be shared and strung together to normalize
- Enumeration mappings can be performed once and reused by different mappers
- Provides the capability to open and edit existing mappers
- All mapping are stored in XML files that can be hand edited if needed.
- Has already proven value by allowing OneSAF KA/KE personnel to create enumeration mappings that can be used directly by the Adapter

ONE
SEM-
AUTOMATED
FORCES



OneSAF C4I Mapper Tool

Wednesday, March 10, 2004
MapperTool ver 0.9

Supports the following formats:

CMP Messages
Java Class
FOM
XML



CMP Messages
Java Class
FOM
XML



The easily navigable design allows for KA/KE individuals to map data with the required domain knowledge.

FROM HEADER

- net.c4i.mapper.header.T47001B_Header
- java.util.Date: getDate()
- java.util.Collection: getRecipients()
- java.lang.String: getClassification()
- java.lang.String: getPrecedence()
- java.lang.String: getOriginator()
- java.lang.String: getMsgNumber()
- java.lang.String: getBaseline()

TO HEADER

- net.c4i.mapper.header.BaseHeader
- void: setDate(java.util.Date)
- void: setRecipients(java.util.Collection)
- void: setOriginator(java.lang.String)
- void: setMsgNumber(java.lang.String)
- void: setClassification(java.lang.String)
- void: setPrecedence(java.lang.String)
- void: setBaseline(java.lang.String)
- void: addRecipient(java.lang.String)

FROM BODY

- TARGET NUMBER : 4.2.1
- TARGET DESCRIPTOR : 4.3.1
- TARGET LOCATION DATA : 4.4
 - IMPACT POINT LOCATION DATA : 4.4
 - GEOGRAPHIC LOCATION DATA : 4.4
 - TARGET LATITUDE : 4.4.2.1
 - TARGET LONGITUDE : 4.4.2.2
 - TARGET ELEVATION : 4.4.3.1
- TARGET TYPE DATA : 4.5
 - TARGET TYPE DATA RECURRENCE
 - TARGET GENERIC TYPE : 4.5.2
 - TARGET SUBTYPE DATA : 4.5.3
 - TARGET SUBTYPE DATA RECURRENCE
 - TARGET SUBTYPE : 4.5.3.1
 - DEGREE OF PROTECTION
 - NUMBER OF TARGET ELEMENTS
 - TARGET SUBTYPE ELEMENTS
- MOVING TARGET TIME DATA : 4.6
- TARGET MOVEMENT DATA : 4.7
- SHELL REPORT DATA : 4.8
- TARGET SIZE DATA : 4.9
 - RECTANGULAR SIZE DATA : 4.9.1
 - LENGTH : 4.9.1.1
 - WIDTH : 4.9.1.2
 - ATTITUDE : 4.9.1.3
 - RADIUS : 4.9.2.1
- TARGET LOCATION ERROR DATA : 4.10
- TARGET COMMENTS DATA : 4.11
- TARGET AIR DEFENSES : 4.12.1
- OBSERVER DATA : 4.13
- TERMINAL HOMING MUNITION DATA : 4.14
- ACCURACY EVALUATION : 4.15.1
- TARGET ACTIVITY DATA : 4.16
- SPECIAL INDICATOR DATA : 4.17
 - EFFECTIVE TIME DATA : 4.18
 - EFFECTIVE DAY : 4.18.1
 - EFFECTIVE HOUR : 4.18.2
 - EFFECTIVE MINUTE : 4.18.3

TO BODY

- net.onesaf.services.odm.c2.message.EndOfMission
- void: setTargetLocation(net.onesaf.services.data.dm.objects.ecs.coordinate.Coordinate)
- void: setTargetLocation(net.onesaf.services.data.dm.objects.ecs.coordinate.OCC)
- void: setTargetNumber(java.lang.String)
- void: setTargetEntityType(net.onesaf.services.sys.bso.EntityTypeEnum)
- void: setTargetEntity(net.onesaf.services.sys.bso.EntityTypeEnum)
- void: setEndOfMissionType(net.onesaf.services.data.dm.rdm.c2.EndOfMissionType)
- void: setEndOfMissionDesignator(net.onesaf.services.data.dm.rdm.c2.EndOfMissionDesignator)
- void: setFuseQuantity(int)
- void: setFireForEffectProjectile(net.onesaf.services.data.dm.rdm.phys.Munition)
- void: setFireForEffectFuse(net.onesaf.services.data.dm.rdm.phys.FuzeTypeEnum)
- void: setProjectileCount(int)
- void: setTargetElevation(int)
- void: setTargetLength(int)
- void: setTargetWidth(int)
- void: setNumberOfTargetElements(int)
- void: populateODMMessage(net.onesaf.services.data.dm.rdm.c2.IntrMessage)
- void: setRecipient(net.onesaf.services.sys.id.UniqueID)
- net.onesaf.services.odm.c2.AbstractMessage: createMessage(net.onesaf.services.odm.c2.AbstractMessage)
- void: setTimeStamp(float)
- void: setSender(net.onesaf.services.sys.id.UniqueID)
- void: setTransmission(net.onesaf.services.odm.entity.AbstractTransmission)
- void: setNetworkType(net.onesaf.services.data.dm.rdm.phys.NetworkTypeEnum)

Data Modifiers:

- Geotransform Lat/Long/Elevation to Double:** Double x, Double y, Double lon, Double lat, Double elev, N/A
- Adjust Longitude 100:** Double lon, Double adjust
- Pkg11_K02.09_4.5.3.2_TO_EndOfMission_EntityTypeEnum:** TARGET SUBTYPE: 4.5.3.2, net.onesaf.services.sys.bso.EntityTy...
- Convert Gregorian Time Unit to float:** Year, number of seconds since semats; Month, Date object with time associated; Day, N/A; Hour, N/A; Minute, N/A; Second, N/A

Annotations:

- Mapped data can easily be deleted by clicking on the line and selecting delete.
- The MapperTool supports mapping from any number of FROM objects to any number of TO objects.
- All mappings are stored in XML files.
- Each internal window may be minimized to reduce clutter on busy mappings.
- Enumerations are stored in their own xml files so they can be reused by other mappings.
- Data modifiers come in many flavors: constants, converters, enumerations, constructors, aliases, and methods.
- Data modifiers can be developed and incorporated into the tool by modifying a configuration file.
- Users may add additional information by attaching notes to mappings.



Questions?

Please Contact:

Steve Lopez-Couto

stephen.lopezcouto@peostri.army.mil



Backup...

OneSAF C4I Adapter Overview

<i>Functionality</i>	<i>Description</i>
Task Manager	One task manager with a configurable number of worker threads in a pool to execute tasks.
Tasks	Tasks are isolated into logical work groups. Work groups can be easily extended or new work groups created, when needed.
Plugins	Allows the C4I Adapter to perform basic modeling or other functionality; interface mechanism built into the adapter for all tasks to utilize, if needed.
JINI initialization service	Allows simulations to dynamically locate, bind and unbind, to a C4I Adapter at initialization as well as run-time. This will also allow for run-time load balancing between C4I Adapters in the future.
Initialization	Reduction of data needed per device, common data put in templates, processed at initialization of the Adapter vs. runtime initialization. Cleaner interface facilitating easier reuse and extensibility.
Singleton applications vs. each interface manager having own copy	The implementation of the Task Manager and Tasks enabled further reduction in the number of threads and processes competing for system resources. Implemented singletons for the following: Protocol Interfaces, the Translator, CMP Interface, Task Manager, Interaction Manager, and the Message Router
Testing Capability	Test drivers and stubs developed to test the C4I Adapter

OneSAF C4I Adapter Overview, continued

<i>Functionality</i>	<i>Description</i>
Decoupling of implementation details from the C4I Adapter framework	The Protocol Interfaces are specified in a Template and may be easily added, removed, or modified. Message router class also configurable via a Template which specifies both a World Comms and Sim Comms class. Simulation C4I Device class implemented for each unique simulation, binds to the correct C4I Adapter via the JINI Initialization Service.
Genericizing of the socket types comms	Socket Comms implemented to handle any socket type communications such as UDP, TCP, unicast and multicast. Easily extendable for http or other socket type communications. Maintenance and extensibility greatly enhanced.
Performance Improvements	Reduced number of threads, elimination of all polling, class pooling.
Reusability Improvements	Cleaner initialization, additional decoupling from the simulation interface, dynamic discovery of services vs. defined in data files, decoupling of the publisher / subscriber interface, plug-in capabilities
Performing Checkpoint without pausing the C4I Adapter	Only components that have checkpoint data register with the checkpoint service. The initiation of checkpoint does not have to traverse the entire C4I Adapter for the checkpoint to occur. Those components that registered

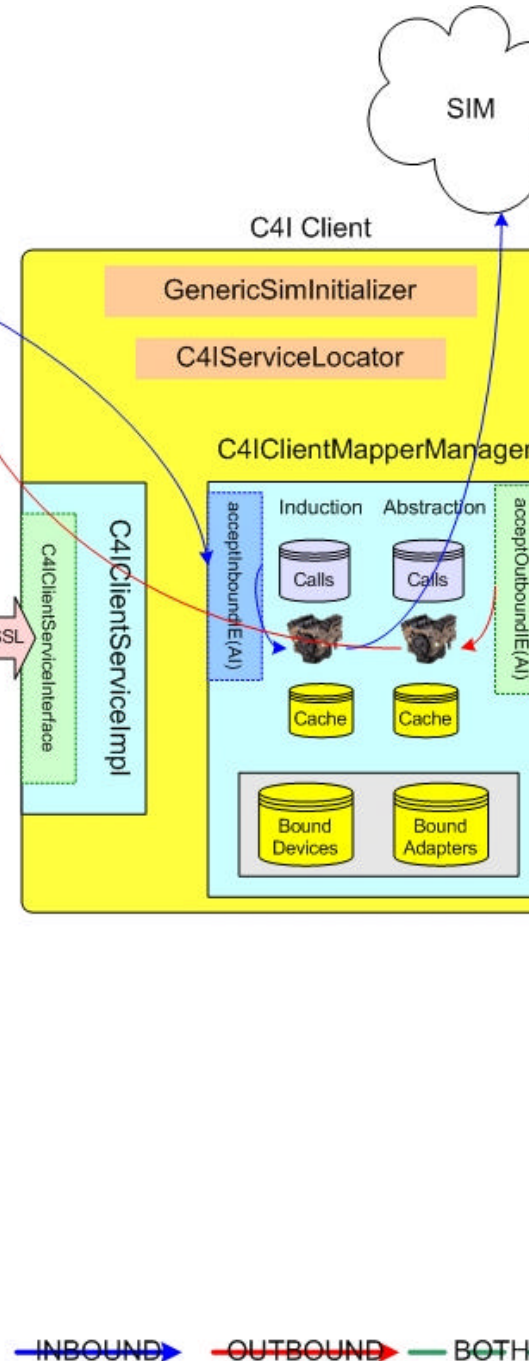
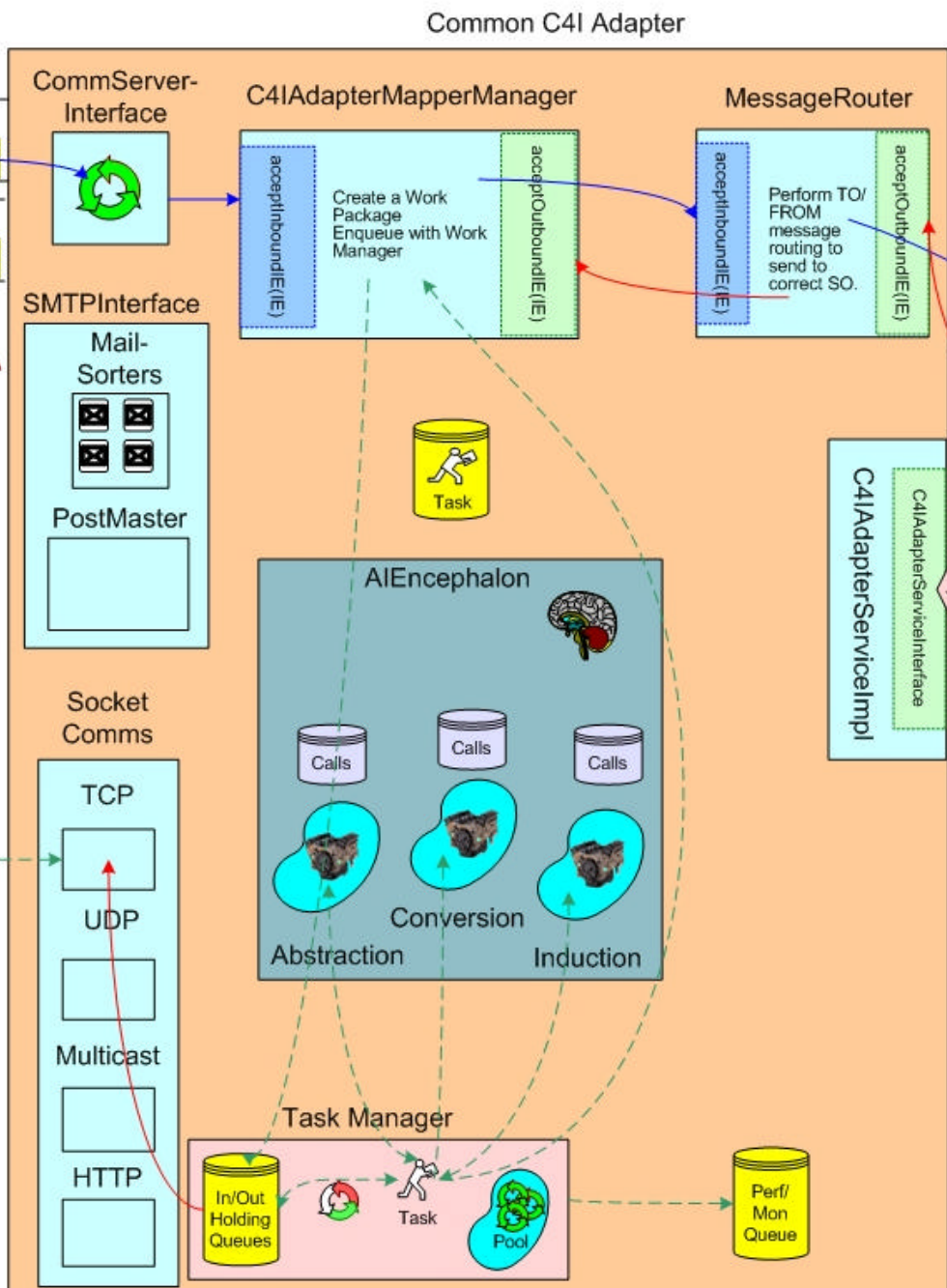
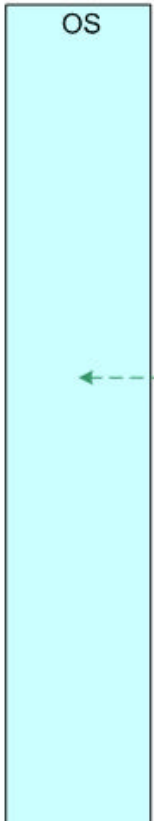
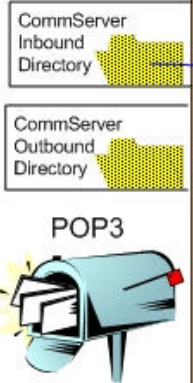
OneSAF C4I Adapter Overview, continued

<i>Functionality</i>	<i>Description</i>
Protocols supported	CommServer (GTCS), email, socket comms (TCP, UDP, Multicast, future HTTP)
Generic Translation capability	XML based Mappers and breaking up the translation process into generic work parts (Abstraction, Conversion, and Induction) allow translation of anything to anything
Lightweight Simulation Interface	Clearly defined API interface, connection to C4I Adapter Main via JINI or SSL
Built in Performance and Monitoring Information availability	Ability to turn on / turn off performance monitoring collecting and logging
Monitor and Control functionality	Allows the monitor and controlling of any C4I Adapter machine from any other machine on the network. Provides monitoring of the following information: <ul style="list-style-type: none">•C4I Devices connected•number of messages being processed•number of messages waiting to be processed•thread status (idle, or task name working on)•C4I Adapter name•host name•IP address of the machine the C4I Adapter is obtaining tactical services from (TOC BS)•JINI service ID

OneSAF C4I Adapter Overview

ONE
SEM-
AUTOMATED
FORCES

0 1 2 3 4 5 6 7 8 9



Task Manager

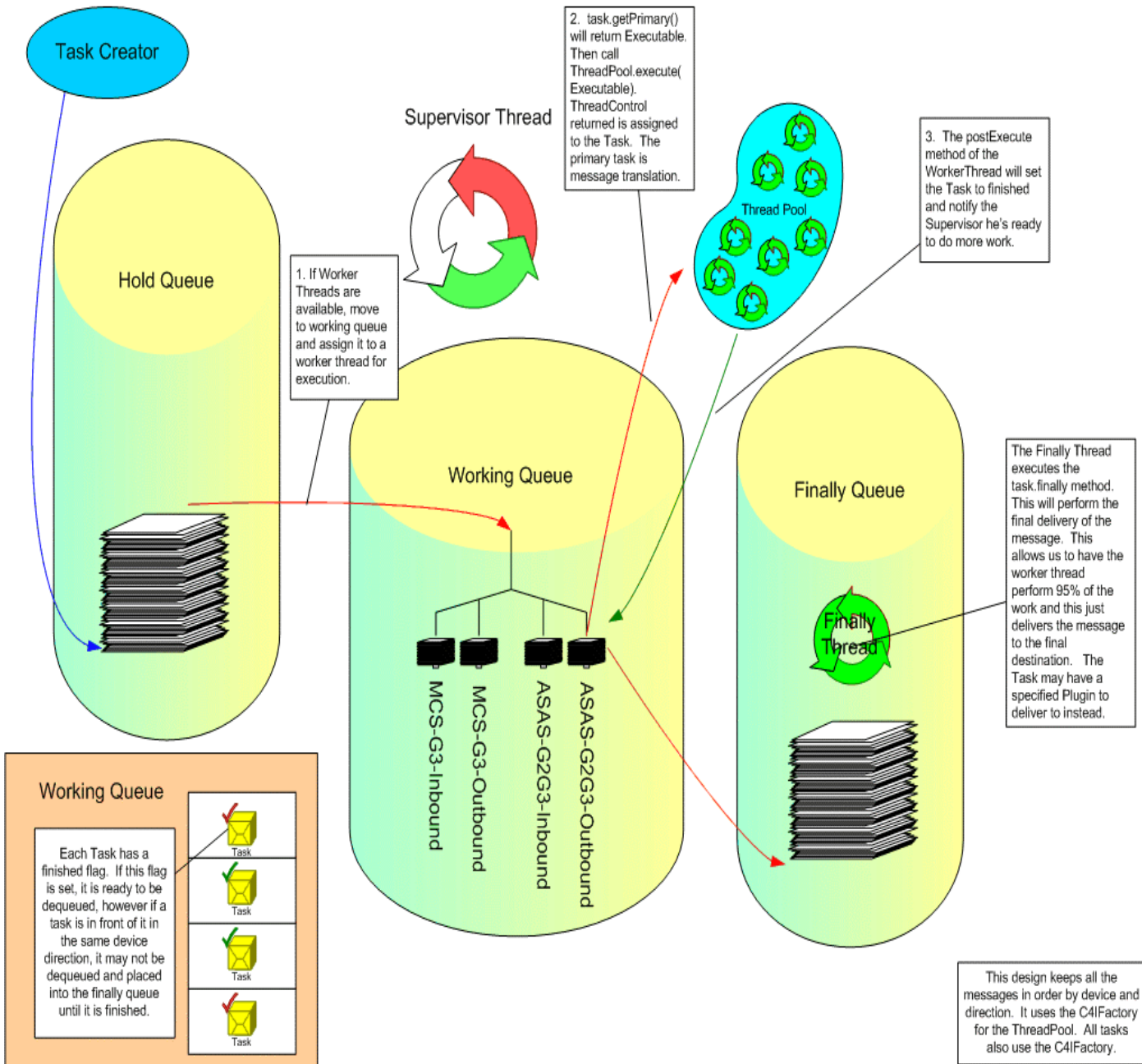
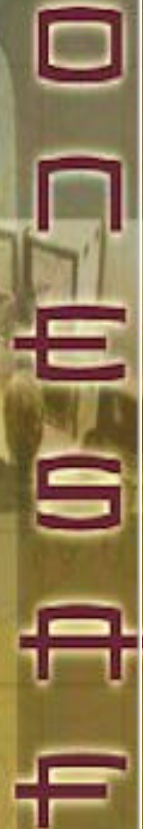
- Configurable number of threads available in the Thread Pool
- The different queues and flags keep all the messages in order by order received, device and direction

Step through of the tasking process:

1. Task Creator creates a task, goes to the Hold Queue
2. If worker threads available, move to working queue, assign worker thread for execution
3. The worker threads perform whatever action is assigned to the Task, primarily translation
4. When the worker thread has finished his task, will notify the Supervisor Thread he's ready to do more work, goes back to the thread pool
5. The task remains in the Working Queue until his "finally flag" is set. The flag cannot be set if a task is in front of it for the same device and direction (I.e. AFATDS Inbound). Preserves the order of messages received.
6. Once put in the Finally Queue, the task is delivered to it's final destination which could be a Plugin application.

Task Manager

ONE
SEM-
AUTOMATED
FORCES



OneSAF C4I Adapter Initialization

Characteristics of the C4I Devices connected to stored in Templates, read in at Initialization

Configuration Options include:

- Sim Comms or World Comms
- Number of Worker Threads
- Logging Levels (including logging of performance data)

Initialization activities started at bootup include:

- CPI creation of the C4I Adapter Infrastructure: Supervisor Thread, Worker Thread creation, Message Router, Interaction Manager
- If Sim Comms (I.e. all communications must go through the simulation), call registerAllExternals C2R API (make the C4I Adapter the recipient for all external TOC traffic)
- Initialization of GTCS, CMPInterface
- Loading of all Template data

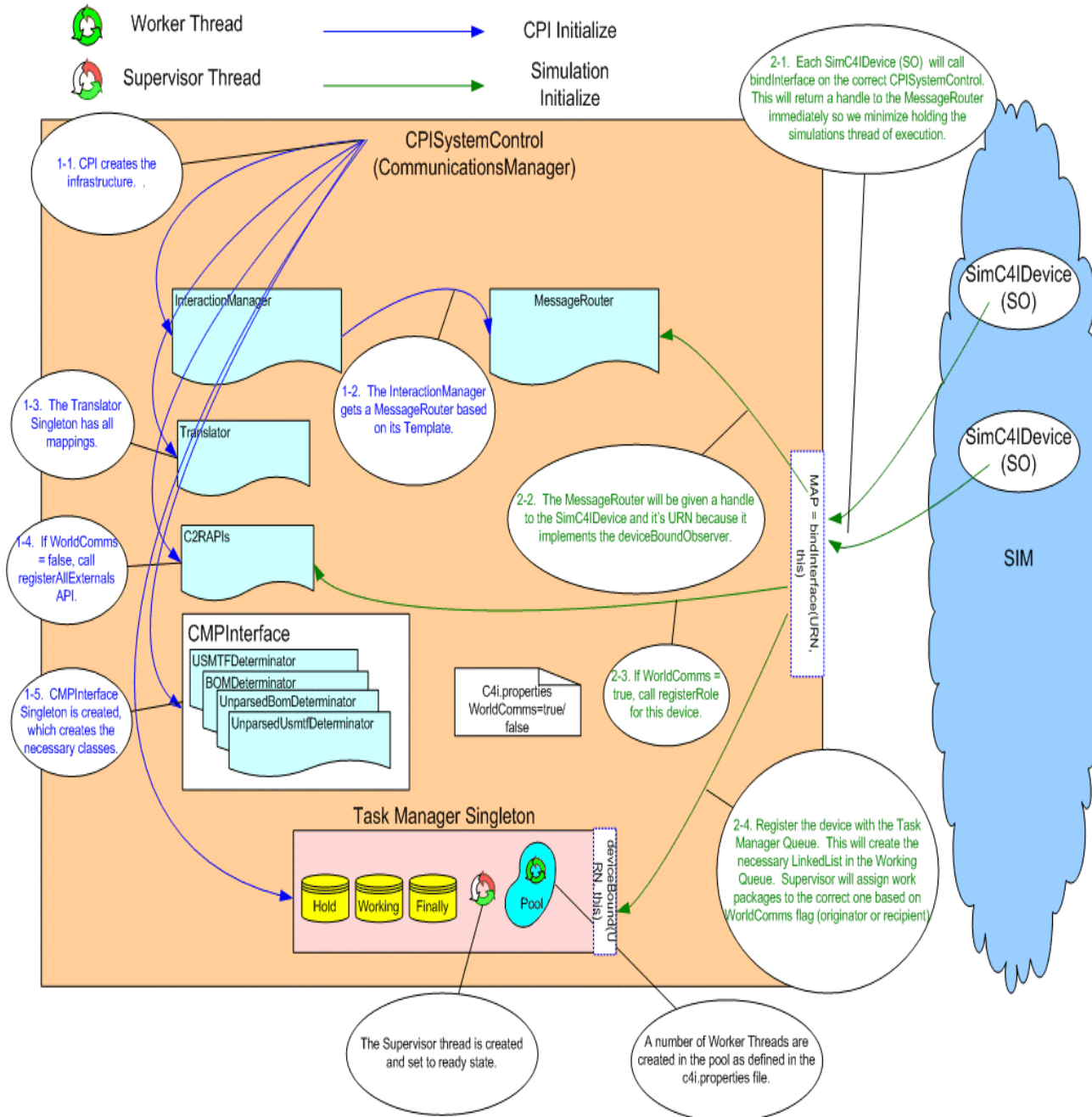
When the simulation is initialized, each SimC4IDevice (simulation representation of C4I device) will bind itself to the correct C4I Adapter via initialization data that specifies URN, CPI ID

- Returned handle to the Message Router immediately so do not hold the simulation thread of control
- If World Comms is true (allow live tactical comms, only route to simulation if entity is simulated), call RegisterRole C2R API for this device
- Register the device with the Task Manager class

C4I Adapter Initialization

ONE
SEM-
AUTOMATED
FORCES

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z



Distributed Services via JINI

Once the scenario file is loaded, the simulation creates the SimC4IDevices
Each CPI (C4I Adapter) is advertising it's ID (for example, 1BDE, 1CAV, CPI1 or whatever).

Each SimC4IDevice has CPI_ID as initialization data. This ID could be changed at runtime (for load balancing) or could add another C4I Device at runtime.

Each SimC4IDevice uses JINI to find the CPI_ID that it is supposed to connect to. It then binds to it's CPI and is immediately ready to send and receive events.

The Monitor and Control Application also uses JINI services. It can monitor and control any of the CPIs that are advertising their ID's on the network.

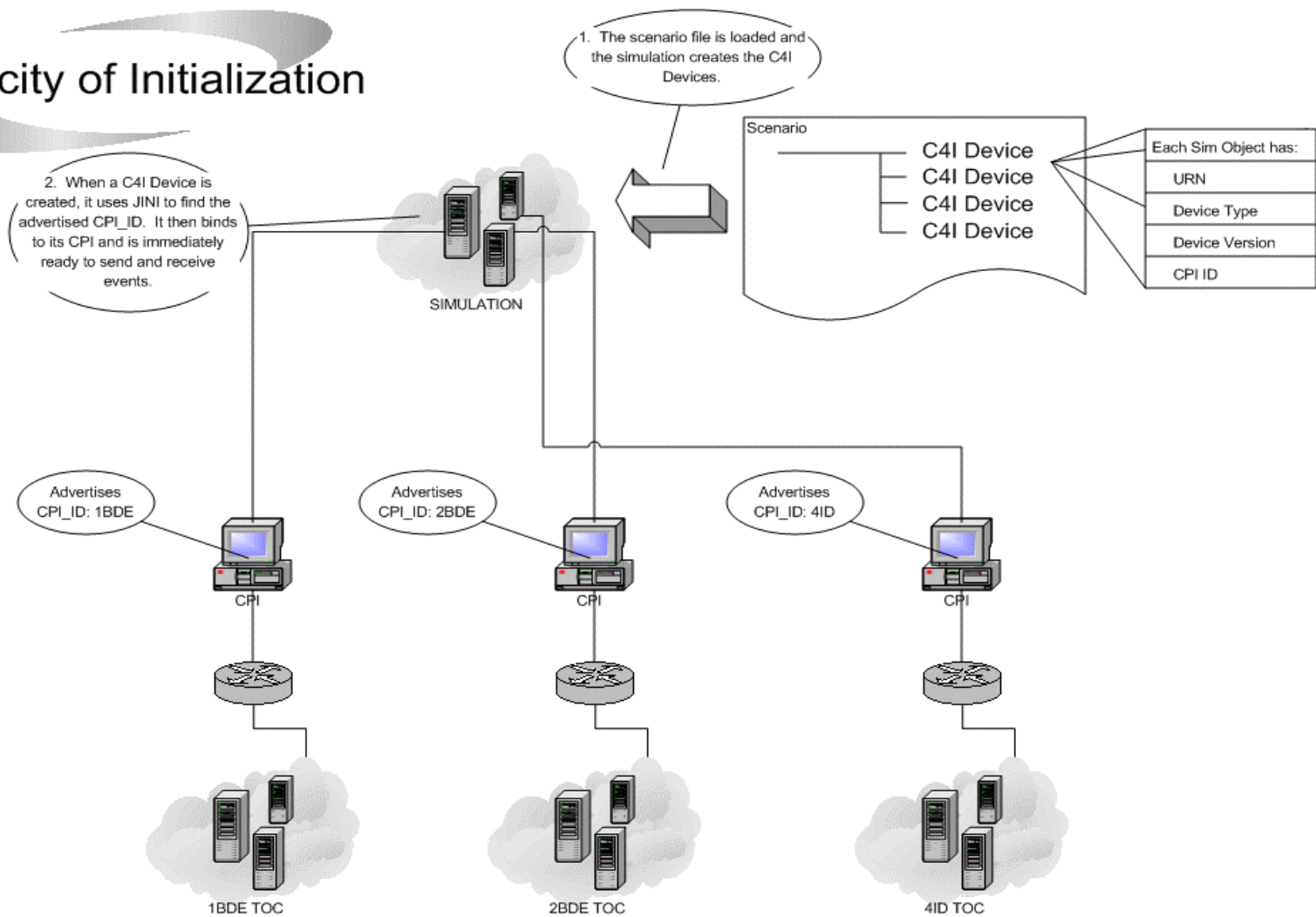
ONE
SEM-
TOMATED
FORCES



Distributed Initialization via JINI

ONE
SEM-
AUTOMATED
FORCES

Simplicity of Initialization



Note:
Network drawn is for World
Comms Off mode.

OneSAF C4I Mapper Tool

<i>Functionality</i>	<i>Description</i>
Mapper Generator Application and Mapper Redesign	<ul style="list-style-type: none">➤ Completing the original intent of a easy to use gui application that allows non-programmmers to create / modify “mappers” that are used at run-time to perform the translation between tactical and simulation formats (and vice versa)➤ Eliminates (or severely limit the amount of) hand creation of ma ppers and / or sections of mappers➤ Maximize reusability of Mappers by decoupling to maximum extent possible, the simulation representation of the tactical message, in the resultant Mapper➤ Automated support for the following:<ul style="list-style-type: none">➤ Enumeration mapping➤ Repeating set mapping➤ Additional conversion routines➤ Eliminates the need for the “CMP generated code”; the “rules” that this code contained are now included in the Mapper Generator Application Reflection engine. At runtime, do not have to execute this CMP generated code.➤ Mappers are XML➤ Code needed to perform a translation greatly reduced:<ul style="list-style-type: none">➤ Location report originally took approx. 5000 Java SLOC (206 for Mapper, 4800 for generated code); now takes 102 XML SLOC➤ Reduction of translations needed:<ul style="list-style-type: none">➤ Original inbound mapping: Byte array -> CMP -> CMP-