Evolution of the Standard Simulation Architecture

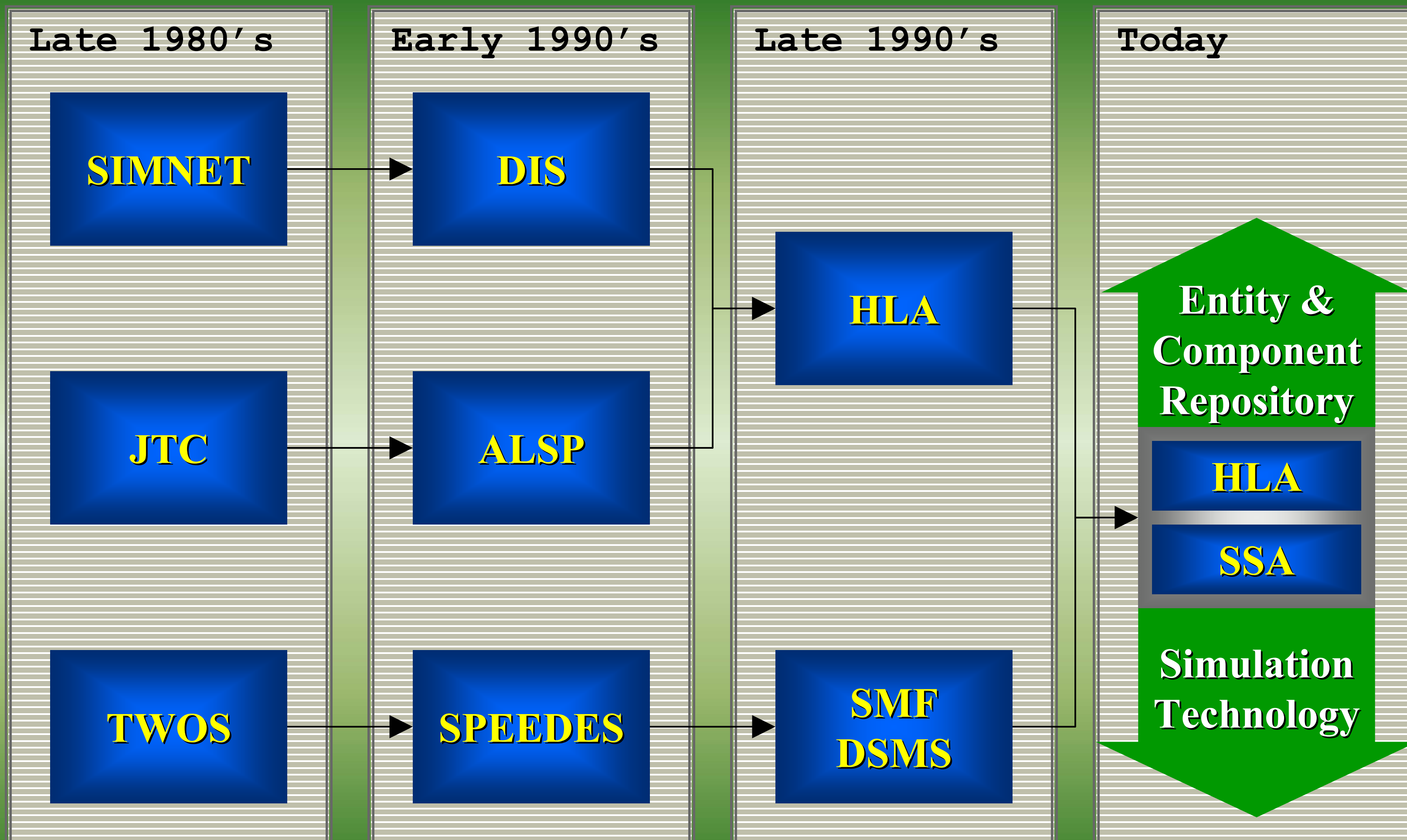
Dr. Jeffrey S. Steinman
Chief Science and Research Officer
RAM Laboratories, Inc.

Douglas R. Hardy
Scientist
SPAWAR Systems Center, San Diego

Benefits of a Standard Simulation Architecture

- **Reduce software development cost**
  - Modeling tools, constructs, and support utilities
  - Improve reliability by reducing code written by developers
- **Facilitate interoperability, composability, and reuse**
  - Entity and component repositories with object composition tools
  - Abstract interfaces decouple software implementations
  - Layered architecture supports technology insertion
- **Provide high performance**
  - Scalable parallel and distributed computing
- **Maximize configuration flexibility**
  - Flexible assignment of models to hardware platforms

Historical Evolution of the Standard Simulation Architecture



Development of the SSA

- Government & industry partnership
  - Investments made by both government and industry
  - SPEEDES, CCSE, WarpIV
- Layered architecture supports inclusive development
  - Academic R&D
  - Research laboratories
  - Multiple industry technology vendors
- Success requires government participation
  - Standards organizations
  - Government programs

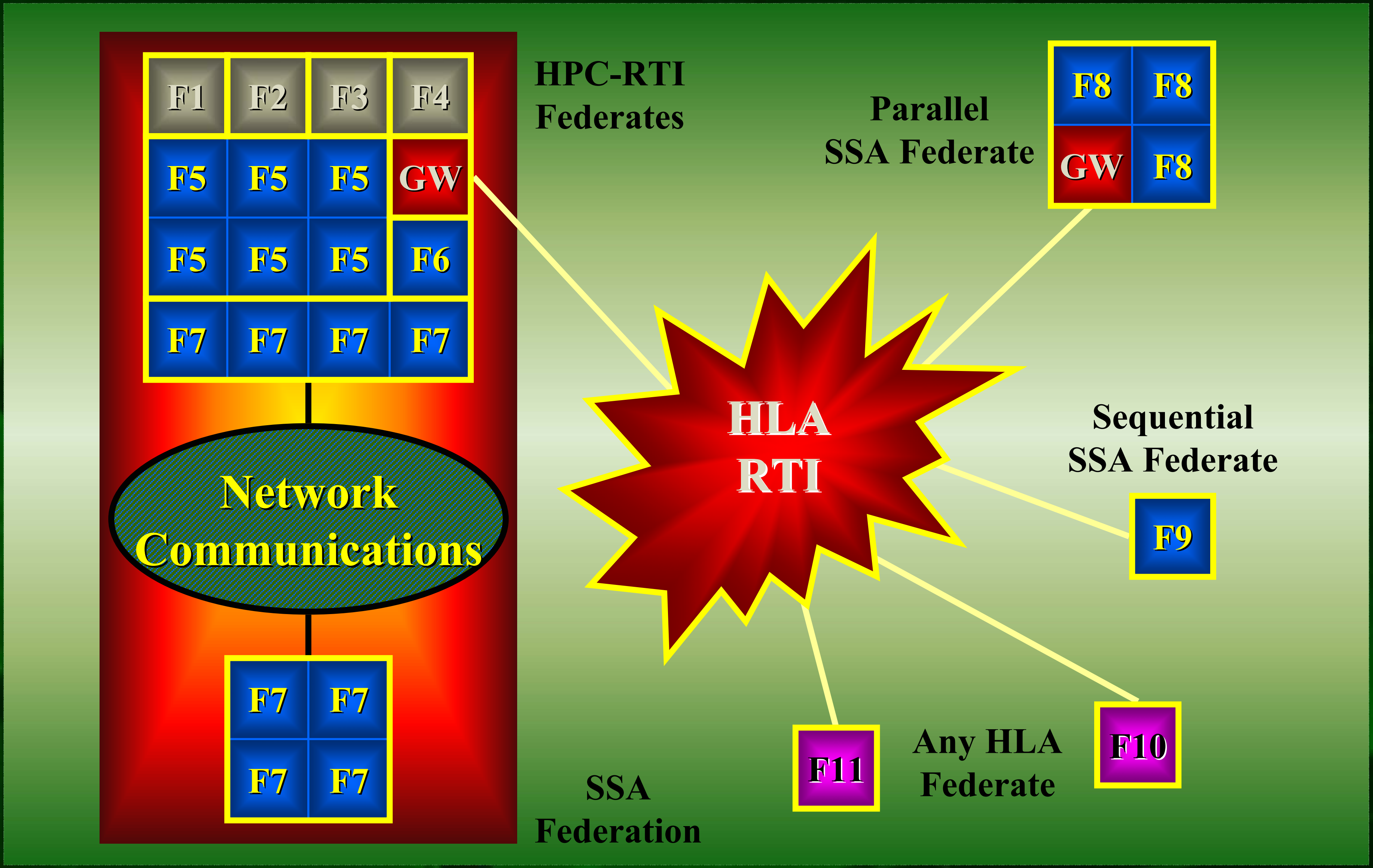
RAM Laboratories, Inc.

Layered Architecture: The Standard Simulation Architecture

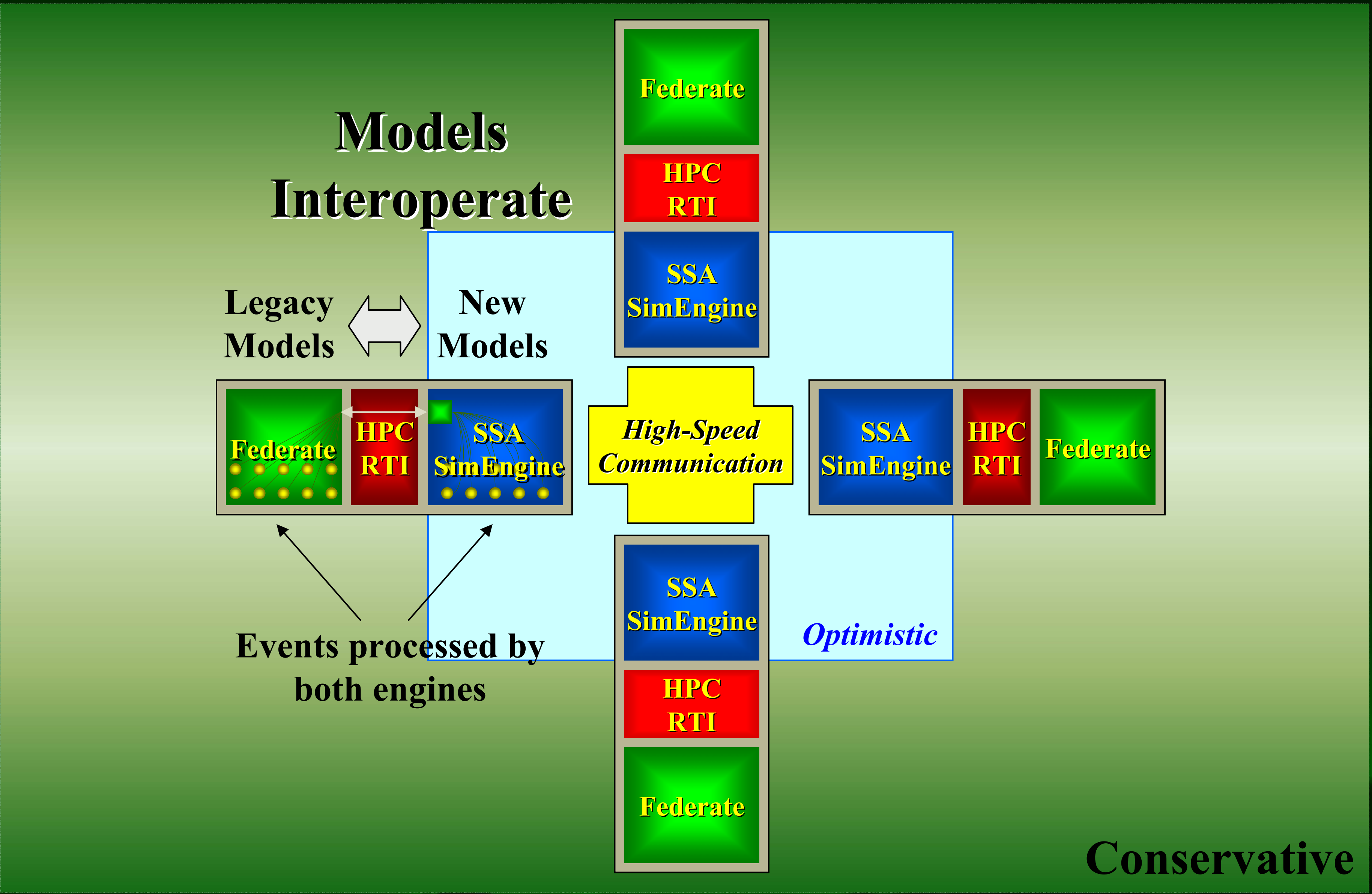
SPAWAR SYSTEMS CENTER SAN DIEGO

<table>
<tr><td colspan="2" rowspan="2">Direct Federate</td><td colspan="2">Abstract Federate</td><td colspan="2">HLA Federate</td><td colspan="2">HLA Federation</td><td colspan="2">External System</td></tr>
<tr><td colspan="2">CASE Tools</td><td colspan="2" rowspan="3">HPC-RTI</td><td colspan="2" rowspan="3">HLA Gateway</td><td colspan="2" rowspan="3">External Modeling Framework</td></tr>
<tr><td colspan="4">Entity Repository</td></tr>
<tr><td colspan="4">Component Repository</td></tr>
<tr><td colspan="10">SOM/FOM Translation Services</td></tr>
<tr><td colspan="10">Distributed Simulation Management Services</td></tr>
<tr><td colspan="10">Standard Modeling Framework</td></tr>
<tr><td colspan="10">Time Management</td></tr>
<tr><td colspan="10">Event Management Services</td></tr>
<tr><td colspan="10">Standard Template Library</td></tr>
<tr><td colspan="10">Persistence</td></tr>
<tr><td colspan="10">Rollback Utilities</td></tr>
<tr><td colspan="10">Rollback Framework</td></tr>
<tr><td colspan="5">Internal High Speed Communications</td><td colspan="5">External Distributed Communications</td></tr>
<tr><td colspan="10">Network Communications</td></tr>
<tr><td colspan="10">Utilities</td></tr>
<tr><td colspan="10">Threads</td></tr>
<tr><td colspan="10">System Services</td></tr>
</table>

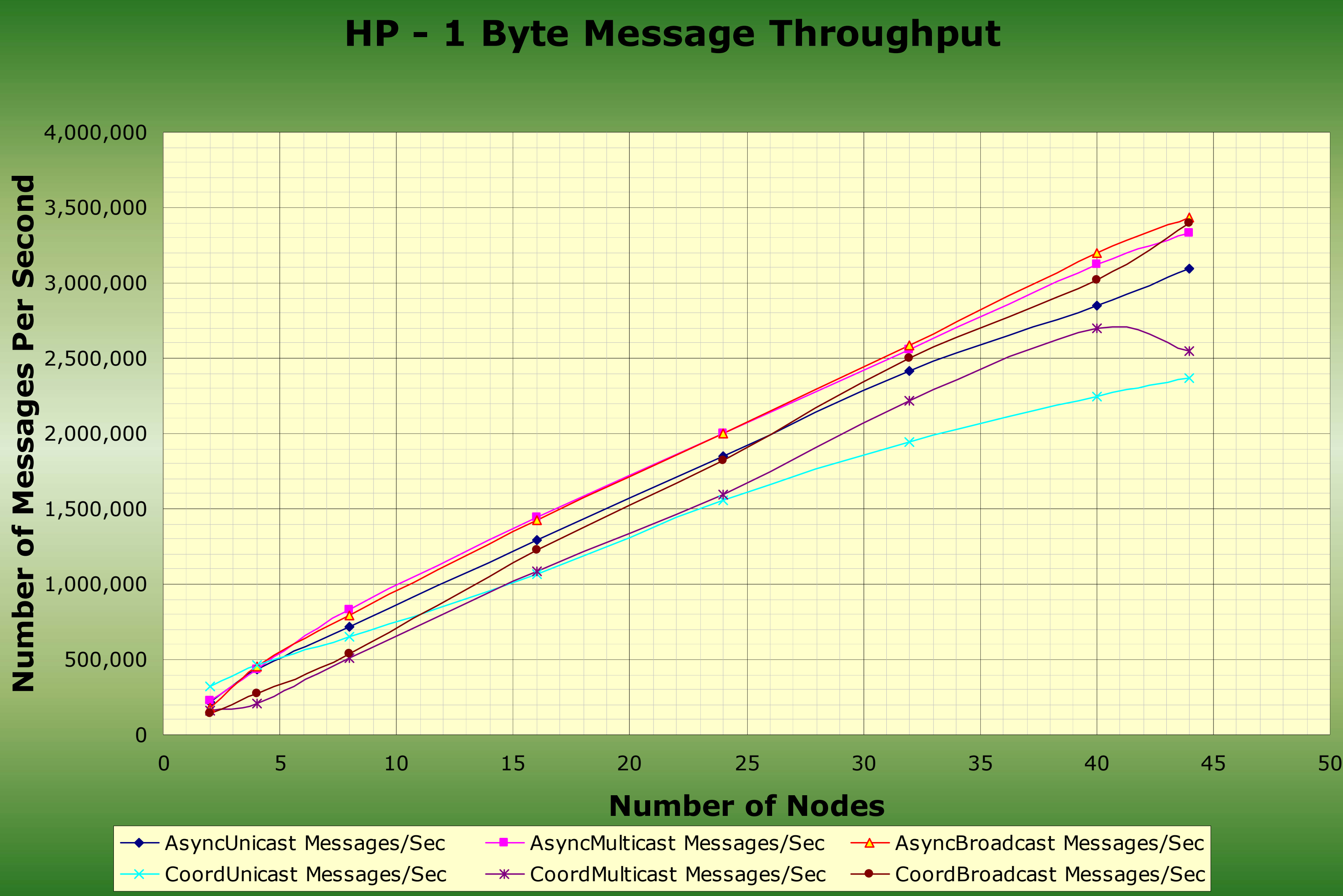
Interoperability and the Standard Simulation Architecture



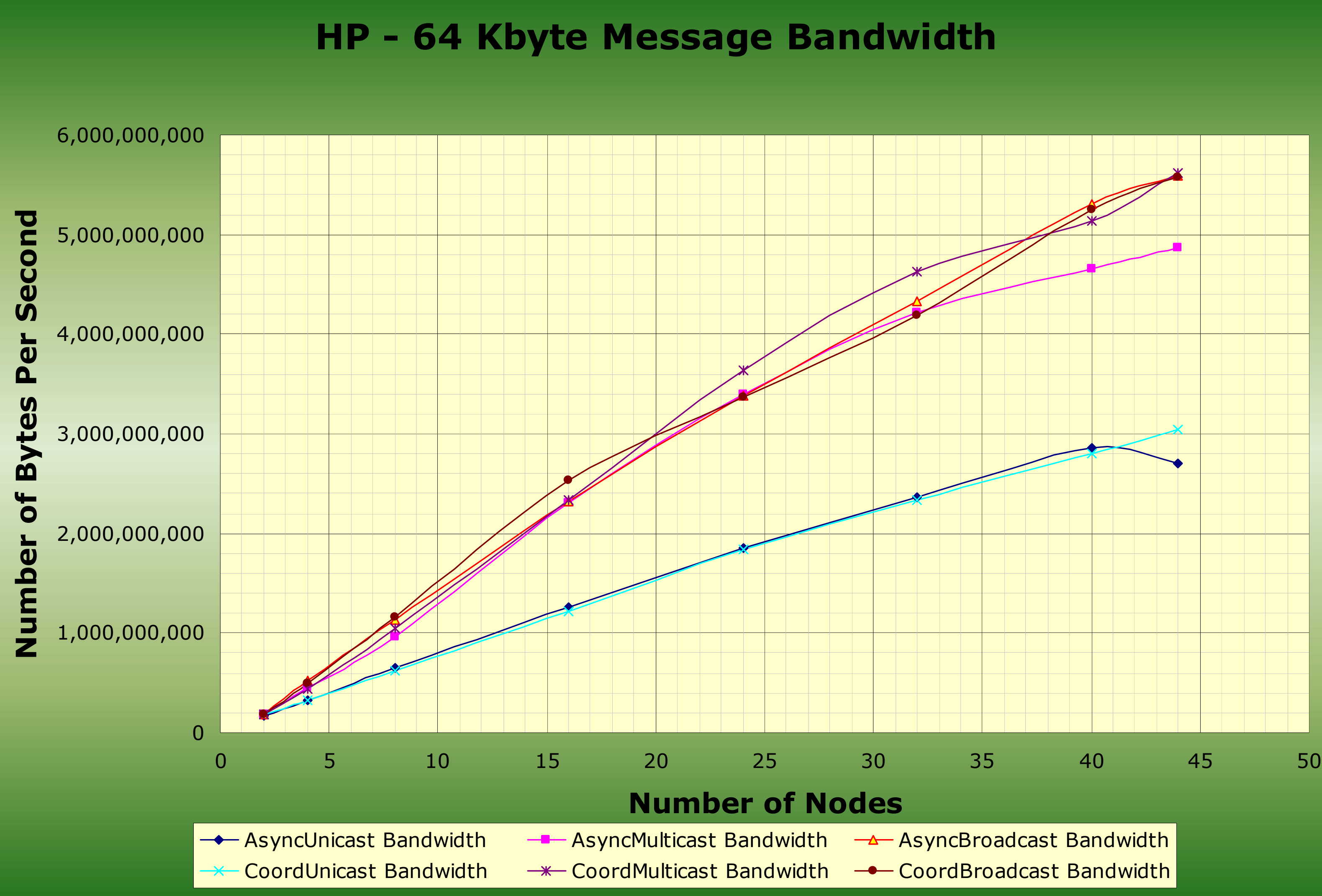
High Performance Interoperability with Legacy Systems



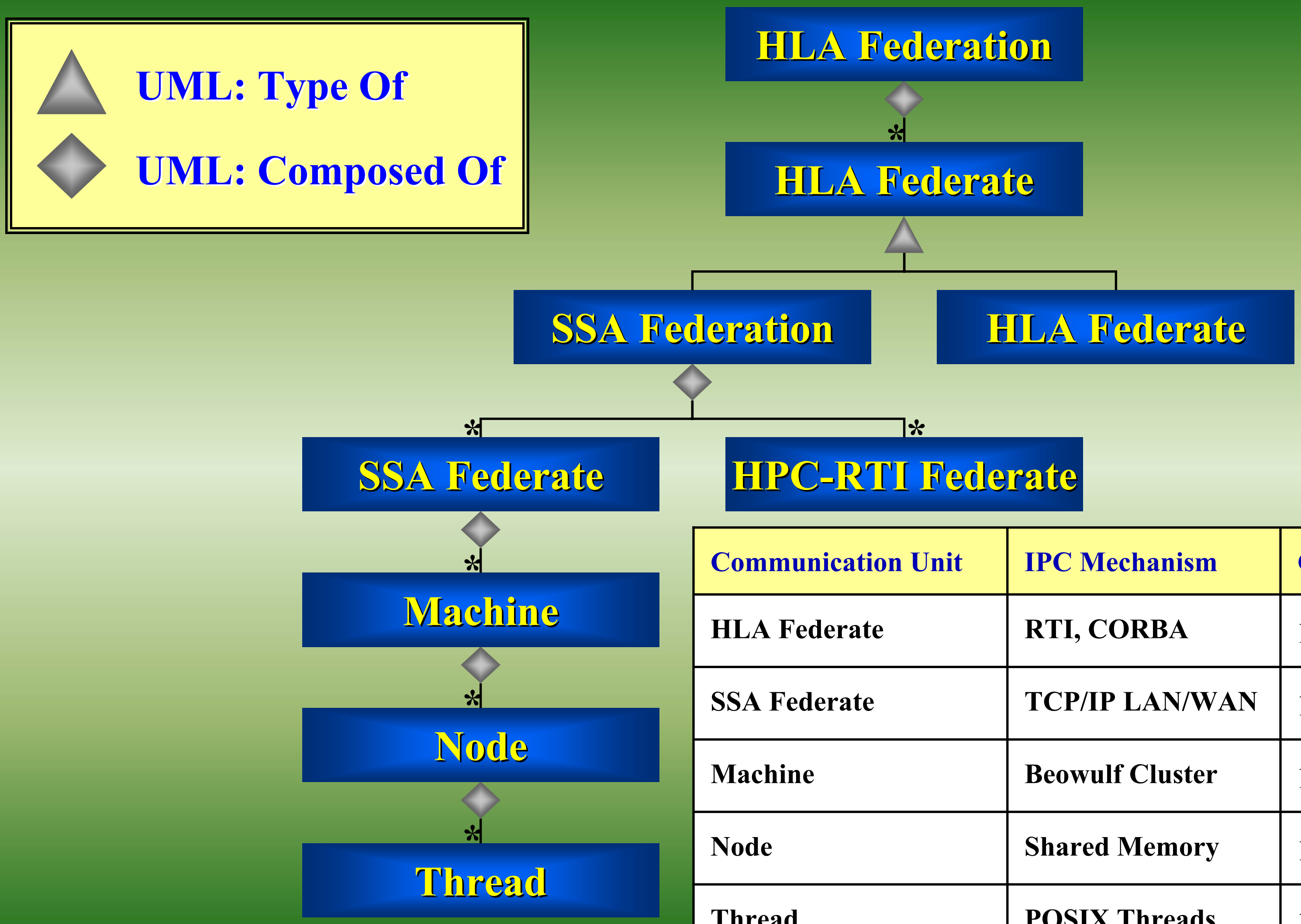
Internal High Speed Communication is Critical for SSA Performance



Internal High Speed Communication is Critical for SSA Performance

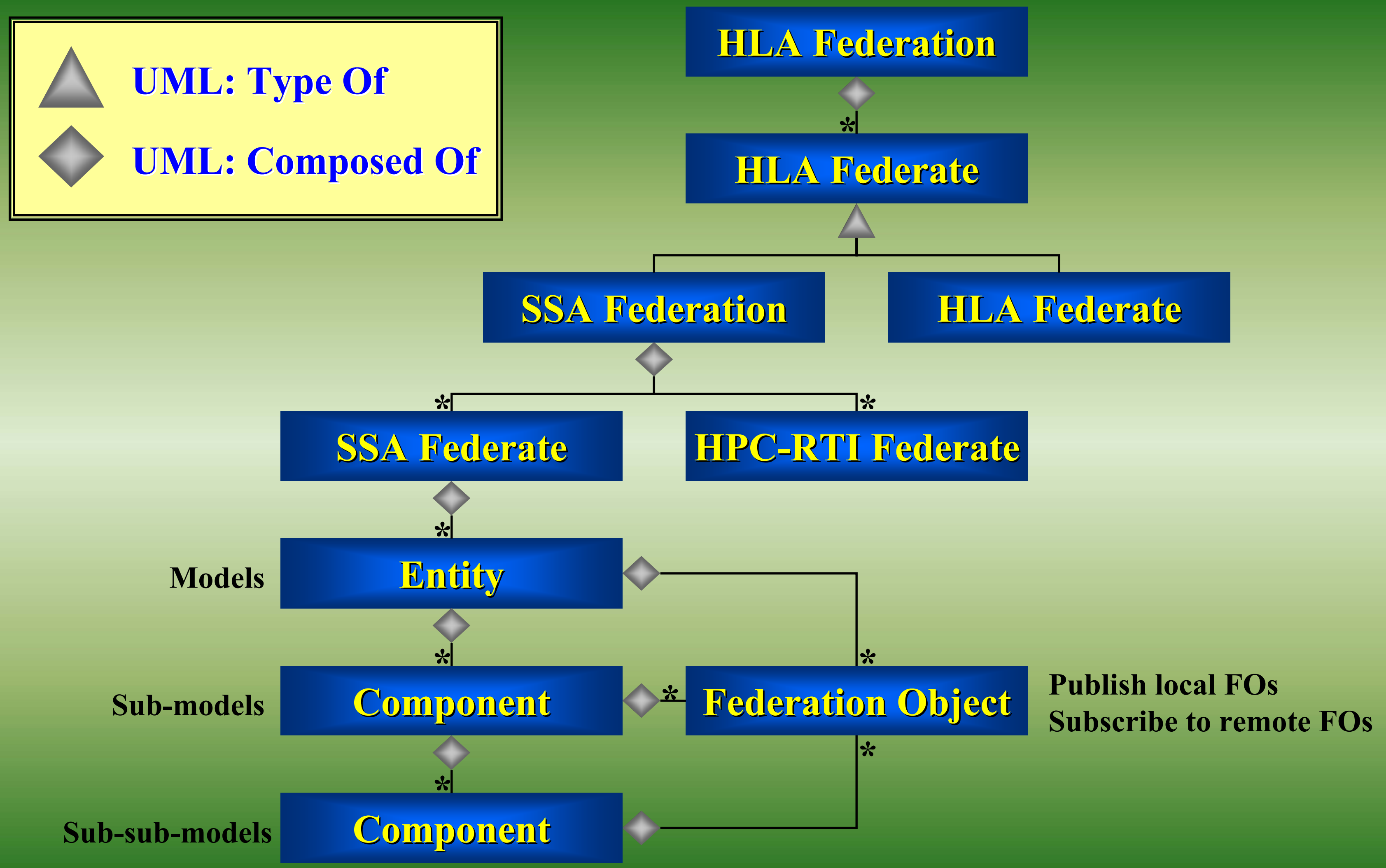


System Composability and IPC Overheads



Communication Unit	IPC Mechanism	Granularity (sec.)
HLA Federate	RTI, CORBA	10^{-3}
SSA Federate	TCP/IP LAN/WAN	10^{-4}
Machine	Beowulf Cluster	10^{-5}
Node	Shared Memory	10^{-6}
Thread	POSIX Threads	10^{-8}

Model Composability



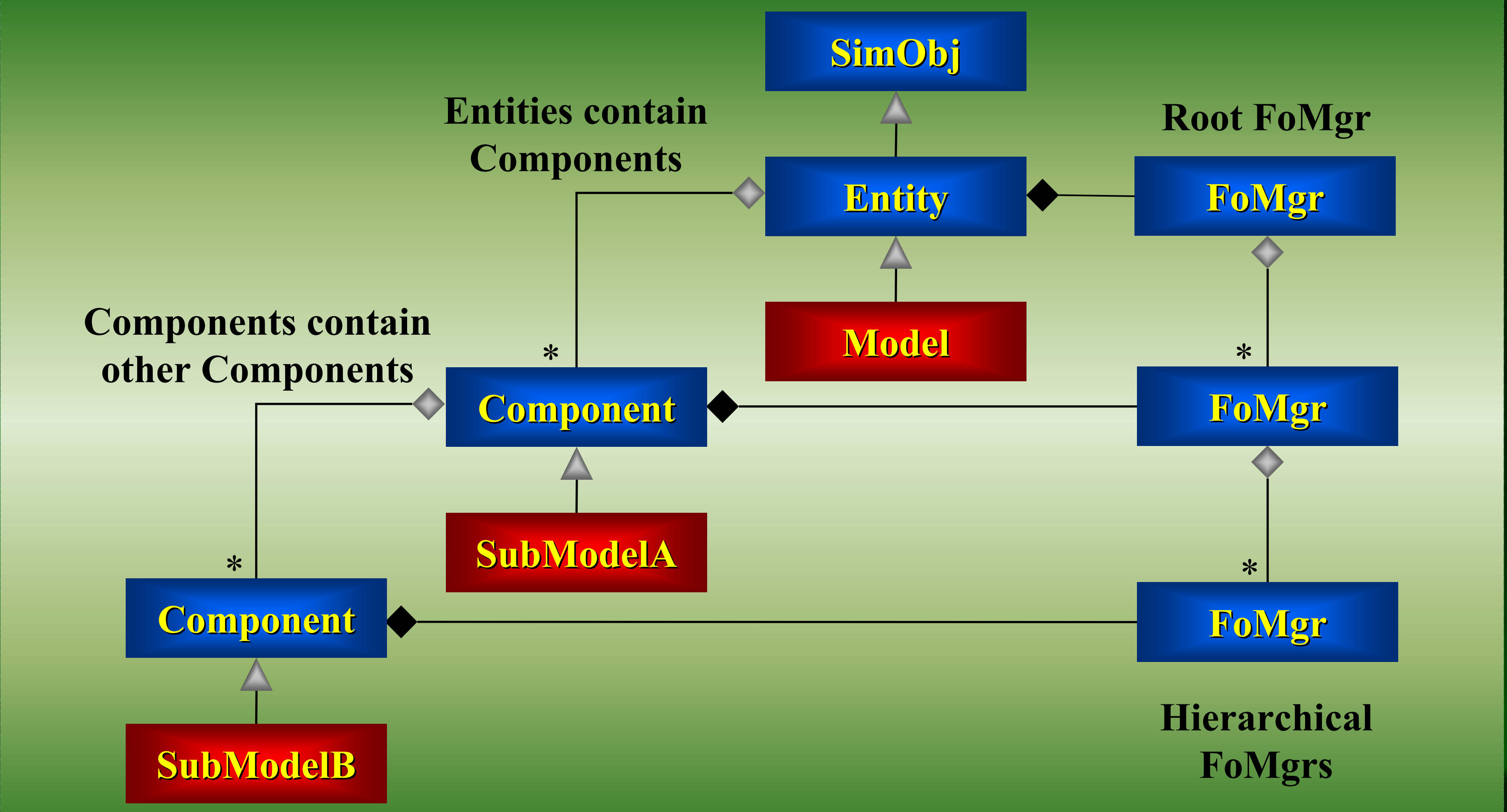
Interoperability

- **HLA provides standards for interoperability between multiple simulations**
  - Coarse grained interoperability
- **The Standard Simulation Architecture provides standards for interoperability for models at three levels**
  - Between **Federates** within an HLA federation
  - Between **Entities** within a (parallel or sequential) simulation
  - Between **Components** within an entity
- **The Standard Simulation Architecture also provides standards for technology insertion**
  - Layered architecture compartmentalizes functionality

Architecture Rules for Model Interoperability

- **Must preserve the abstraction that an entity may reside on any node when running in parallel, or within any federate when executing in an HLA federation**
  - Entity state exchanged with other entities must be provided exclusively through **Federation Objects**
  - Entities interact with other entities exclusively through HLA-style **Interactions**
- **Entities behave like miniature federates…**
  - DSMS Layer provides HLA functionality between entities
  - Operator overloading in C++ automates distribution of attributes
  - Interest management automatically operates on attributes

Hierarchical Composability

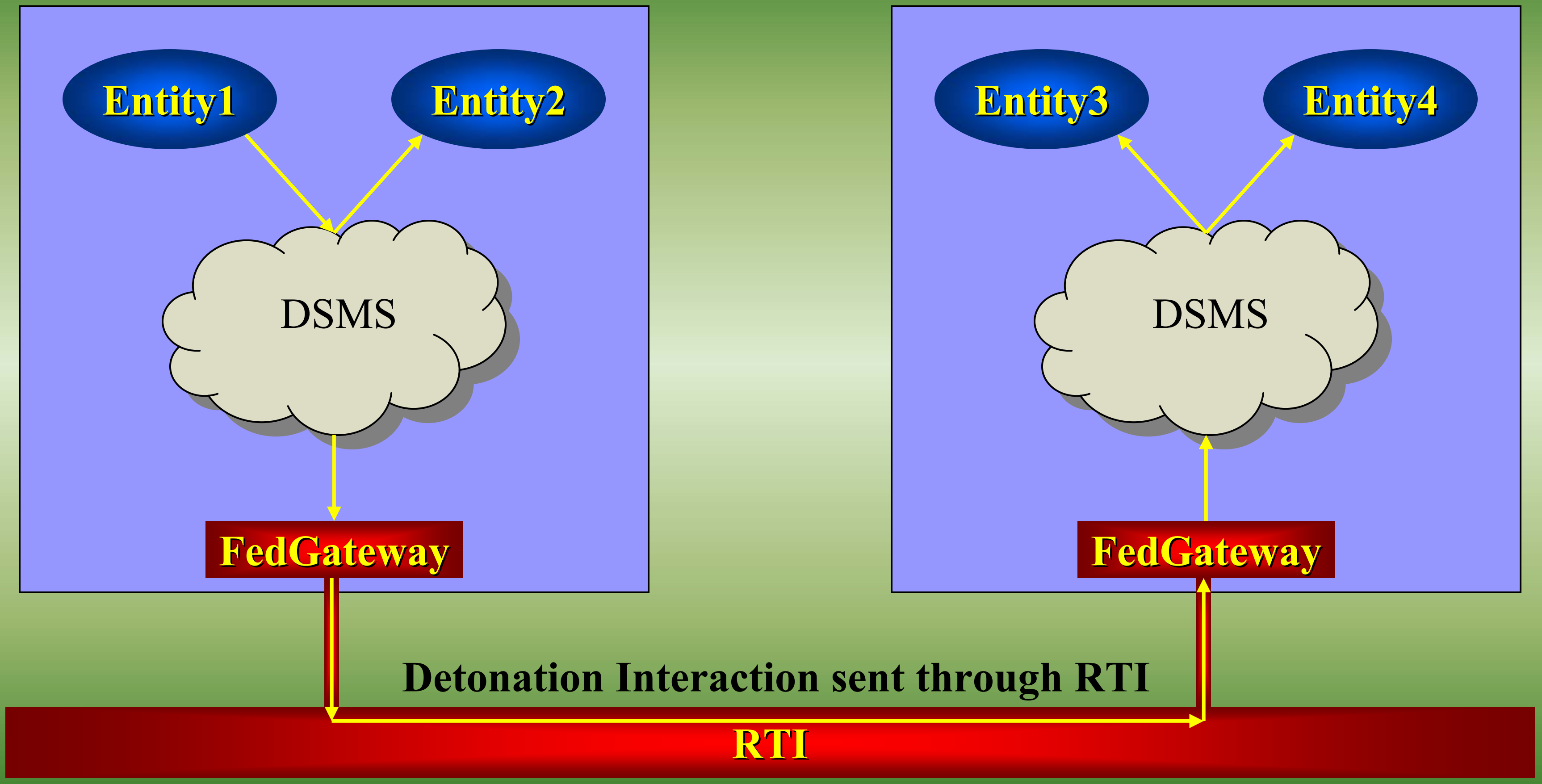


Federation Object Management

- **Interest Management is automatically provided between Entities**
  - Filtering is automatically performed on attributes as they change
  - Hierarchical grids supports multi-resolution scalability in parallel
- **Interest Management is automatically provided between Components**
  - FoMgrs filter FOs based on Component subscriptions
  - Special component for range-based filtering

Interactions Through the DSMS Layer and the HLA Gateway

Entity1 sends a Detonation Interaction



Double Abstraction Barrier Principle

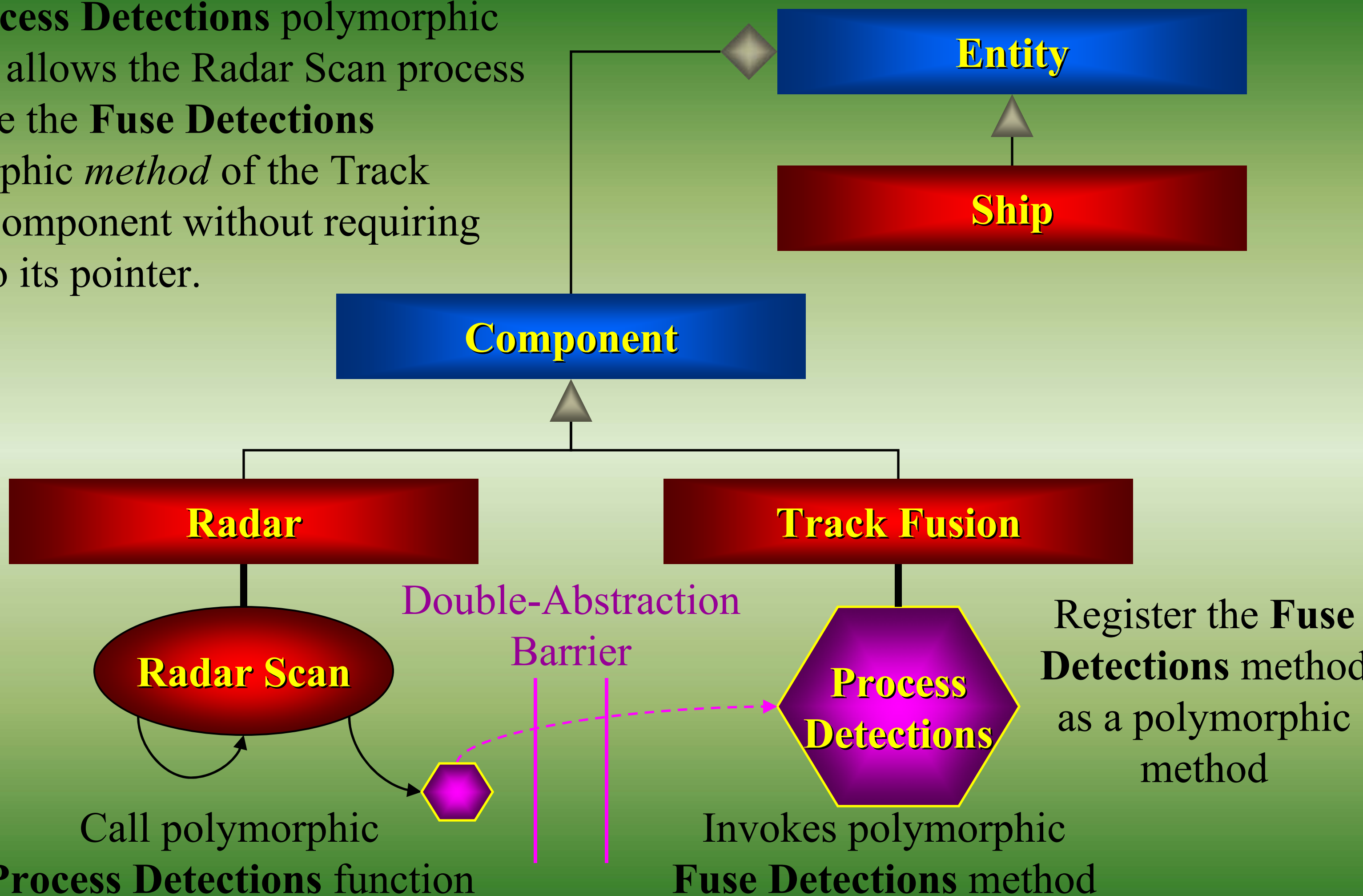
- **Interactions between HLA Federates (~milliseconds)**
  - Federates do not know which other federates have subscribed
  - Federates do not know how interaction is processed
- **Interactions between Entities within an SSA federate (~microseconds)**
  - Entities do not know which other entities have subscribed
  - Entities do not know how interaction is processed
- **Polymorphic methods between Components within an SSA entity (~nanoseconds)**
  - Components do not know which classes have registered
  - Components do not know which methods are registered

RAM Laboratories, Inc.

SPAWAR SYSTEMS CENTER SAN DIEGO

Components and Polymorphic Methods

The **Process Detections** polymorphic *function* allows the Radar Scan process to invoke the **Fuse Detections** polymorphic *method* of the Track Fusion component without requiring access to its pointer.



Standardization Encourages Three Business Models

- **Government Off The Shelf (GOTS)**
  - Development by government laboratories
  - Government provides life-cycle maintenance
- **Open Source**
  - R&D by research institutions and universities
  - Successful R&D feeds into real programs
- **Commercial Off The Shelf (COTS)**
  - Development by industry
  - Users buy software licenses with support contracts

Standardization Steps

- **Government sponsorship and oversight**
  - Establish SISO working group to study standards issues
  - Management of the standardization process
  - Requires appropriate level of funding and commitment
- **Architecture Participants**
  - **Engineering Team** comprised of proven simulation technologists to refine standards
    - Industry, Government, and University
    - Prototype standardized interfaces and services
    - Joint development of unit and system test suites
  - **Technology Panel** of specialists review individual layers
  - **User Group** generates feedback on services

Summary and Conclusions

- **The Standard Simulation Architecture addresses critical needs of DoD simulation community**
  - Interoperability between federates, entities, and components
  - Facilitates object composability
  - Layered architecture promotes technology infusion
  - High performance computing
  - Portability and flexibility
  - Reduces software development costs while improving reliability
- **Requires government sponsorship and oversight**
  - Commitment to standardize and implement the SSA layers
  - Programs must focus on model and component reuse
  - COTS, GOTS, and Open Source business models for technology insertion

WarpIV Simulation Kernel

- **WarpIV provides prototype development of the SSA**
- **RAM Laboratories is currently offering WarpIV to:**
  - Universities
  - Research Laboratories
  - Government Programs
  - Industry
- **For more information about WarpIV, see our website:**
  - http://www.ramlabs.com