



Performance from Experience

Evolution of EAI: ***The Future of EAI, Messaging and XML***



Prepared For:
OMG

Telcordia Contact:
George Collier
Mohammad Moghadam

collier@research.telcordia.com
973 829-4762

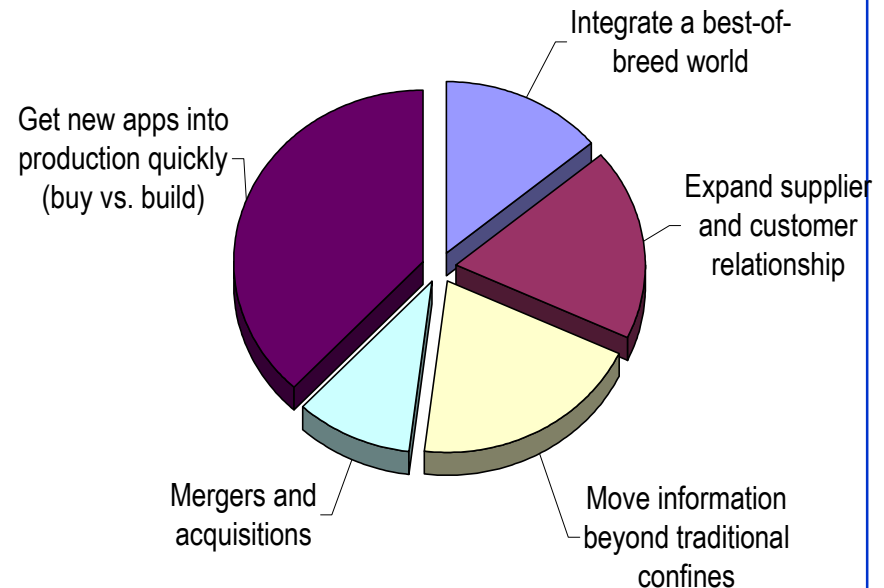
February 8, 2000

The Drivers for EAI will Not GO AWAY!



EAI Drivers

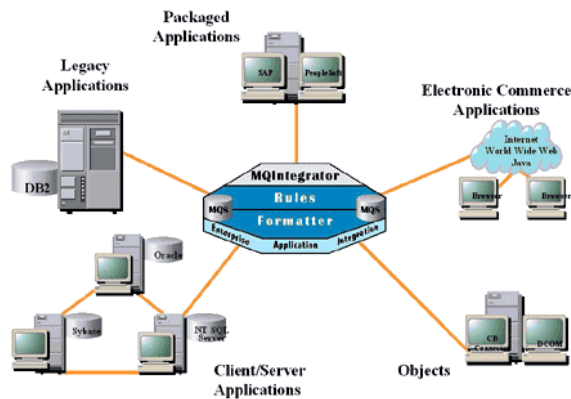
- Semantic Canyon
 - incompatibilities in the conceptualizations underling systems
- History
 - IT infrastructure built up over time
 - Why Y2K? Nobody thought the systems would last this long ...
- Mergers and Acquisitions
 - Combining two existing IT infrastructures
- Electronic commerce
 - Teaching old dogs new tricks



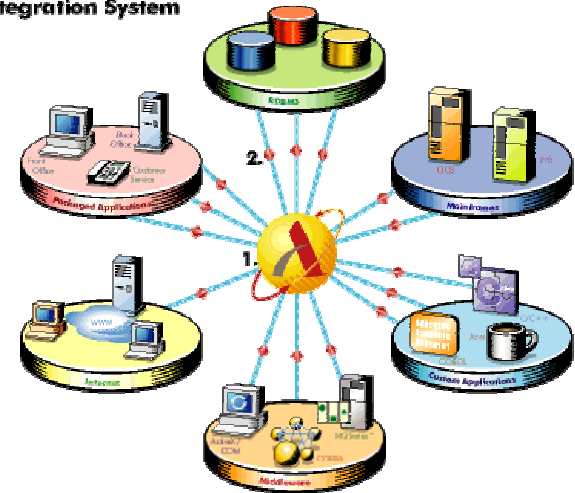
Needs behind application integration (Source: Aberdeen group, 11/98)

Therefore: Applications Dedicated to Large System Integration = **EAI**

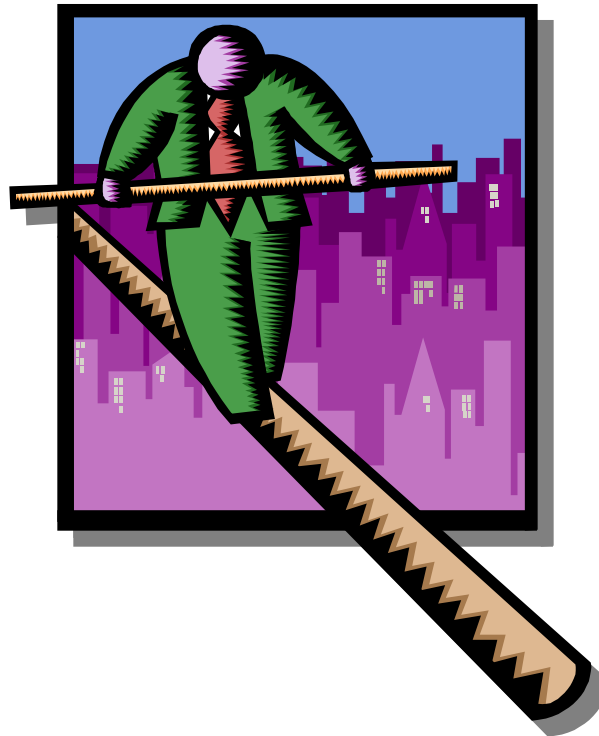
- There will always be a demand a place for a meeting of minds
- This place will supply translations, “trade languages” and other mechanisms to mediate syntactic and semantic gulfs
- It will provide “loose-coupling” between participants



**ActiveWorks
Integration System**



The Desire for De-coupling Drives the Requirements for EAI

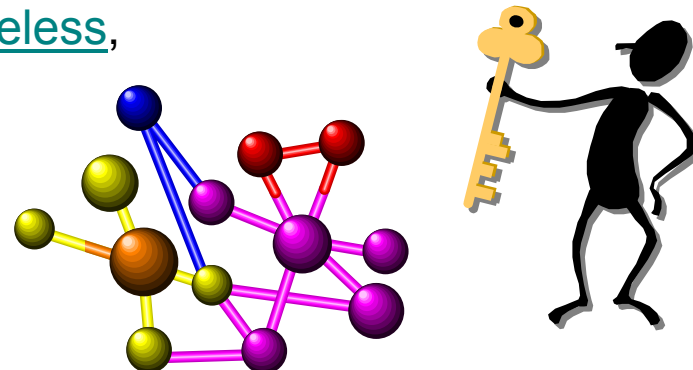


The Desired Architecture Reduces Complexity by Reducing Dependencies (Simon Phipps - IBM)

- Enterprise computing ... IBM: been there, done that
 - Delivery, program, data, network, security
 - Modularity is key ...
 - "clear space between the parts"

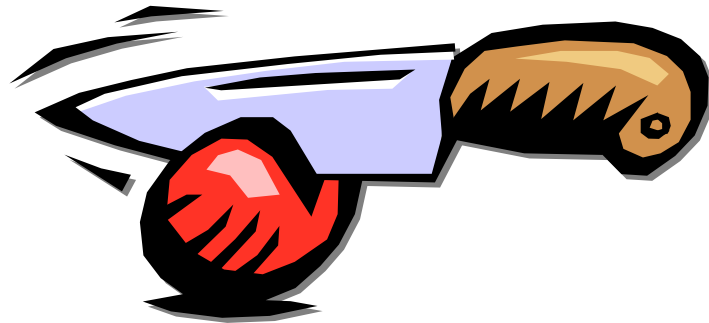


- Emerging neutral solution: Web + Java + XML + TCP/IP
 - Web is platform neutral, stateless,
 - Java is platform neutral,
 - XML is data neutral
 - Security will be public key



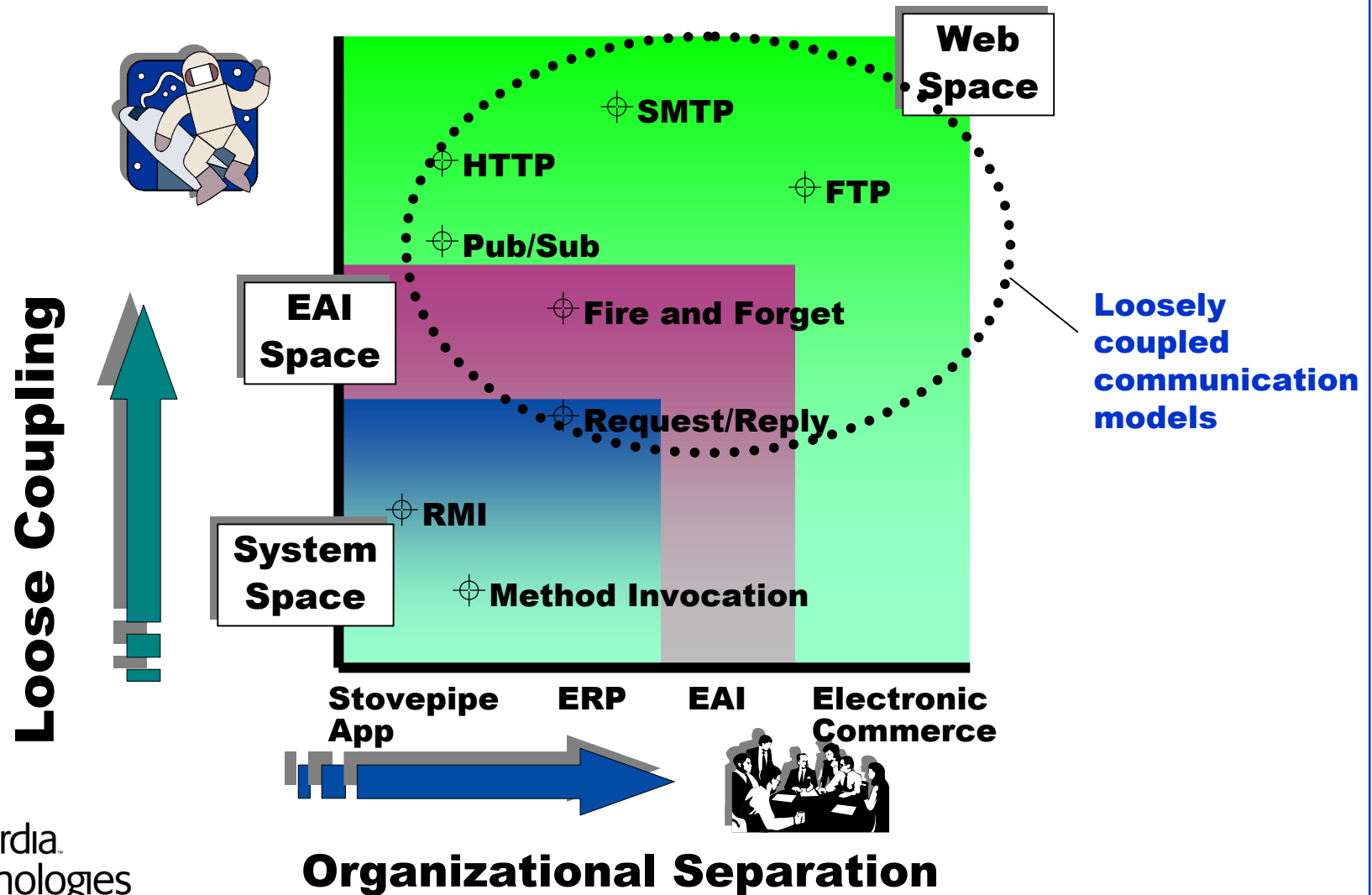
EAI “Separations” which Reduce *Dependencies*, Create *Modularity* and Reduce *Complexity*

- Data from transport
- Publisher from subscriber process management
- Internal event from public event
- Internal data structure from external data structure
- Request from reply
- Connectivity from delivery (guaranteed delivery)



Multi-Organizational Computing Force

Multiple Communication Models



The **P**unch **L**ine



- The drivers for EAI will not disappear
- EAI represents a new systemization and methodology and supporting tools for solving integration problems
- Middleware will grow to encompass this paradigm to some degree, esp., transport, pub/sub, filtering
- Transport will converge to a short list of features supported by all major vendors
- Adaptation, translation and process management will converge much later or not at all
- Separation of transport layer and message contents will be the norm
- XML will rule the message contents



Advantages of XML/Events over IDL Data for Loosely Coupled Applications:

- *Late binding/interpretation*
 - Allows for graceful degradation in the face of mismatches
- *Human readability*
 - Allows for ease of debugging
 - Allows for easy mapping into presentation formats
- *Multiple presentations*
 - Technology supports rule-based “multiple” presentations
- *Processing via simple parsers vs. heavyweight ORBs*
 - Simple programs can be used to process the data
 - KISS me because stupid is good!
- *Web infrastructure*
 - Immense intellectual/monetary capital being invested in the Internet
 - Huge infrastructure of primitive data and transaction servers