


mercator[®]

Intelligent Business Integration

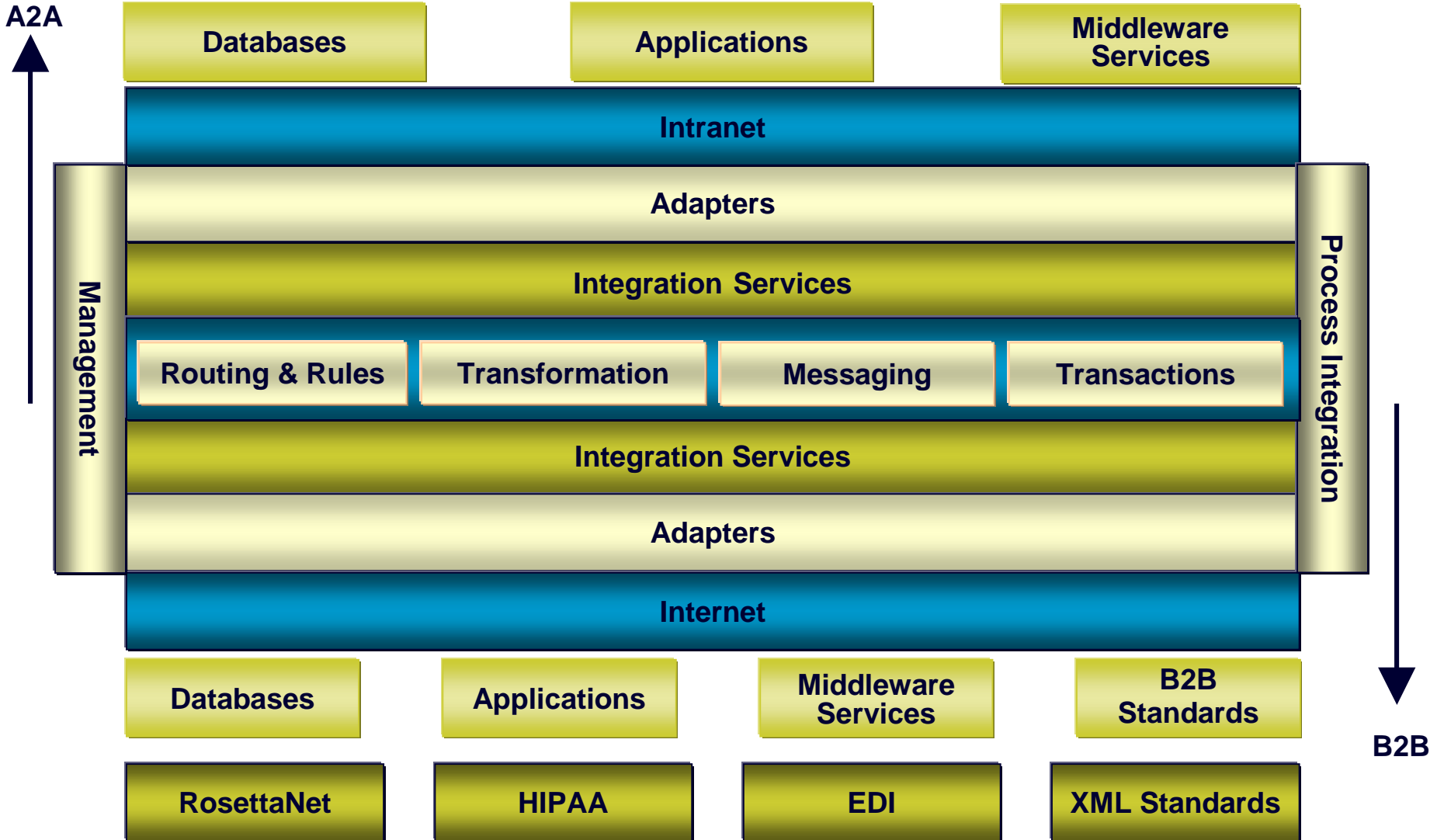


Leveraging Web Services Application Integration

David S. Linthicum

CTO Mercator

Master Technology Stack



Application Integration Types

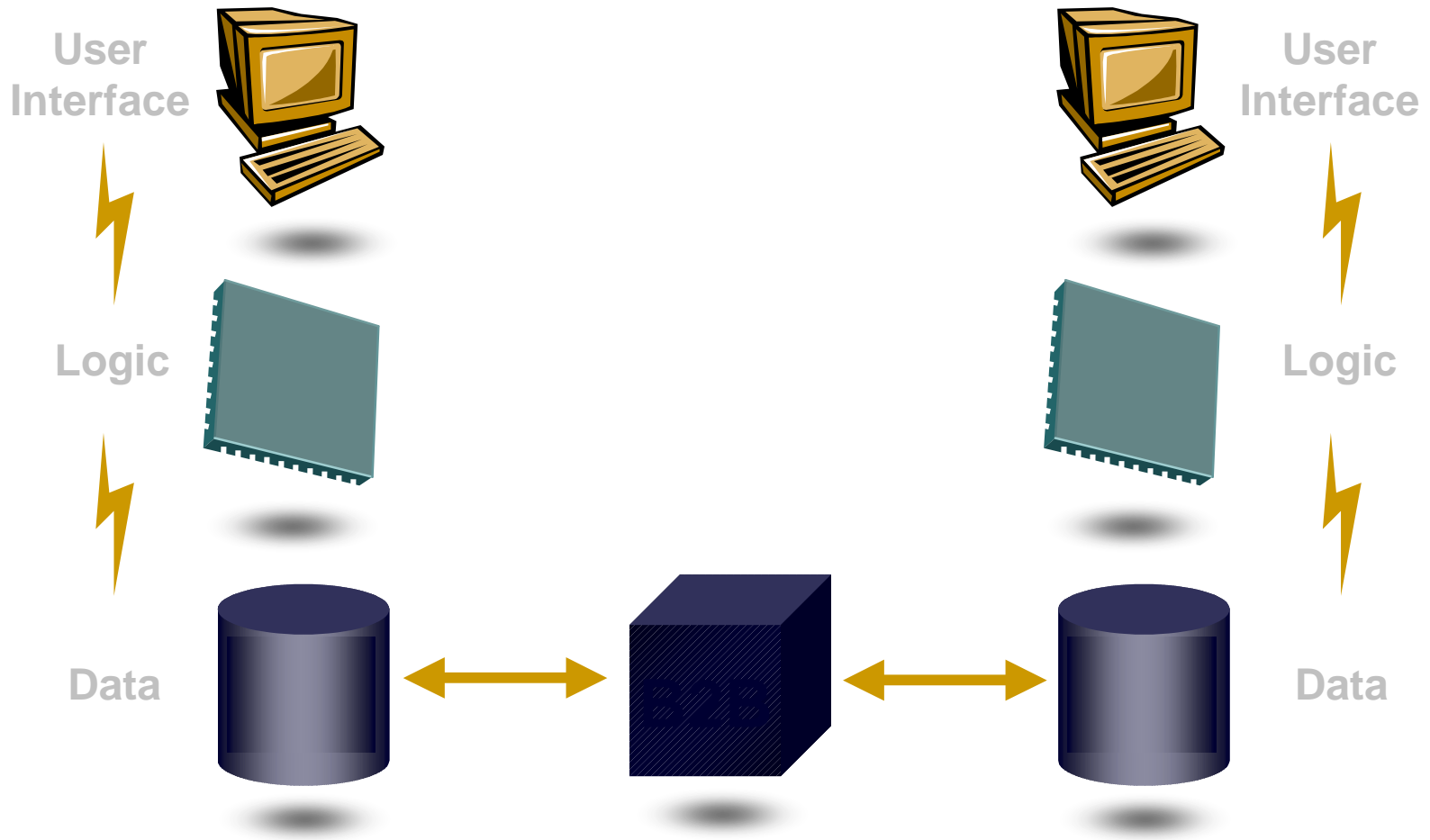
Portal -Oriented

Process Integration -Oriented

Application Service -Oriented

Information -Oriented

Information-Oriented Application Integration



Transformation Formatting

Application Service-Oriented Application Integration

- ▶ Method integration to create a composite application
- ▶ Sharing business logic
- ▶ Sharing code
- ▶ Sharing processing
 - Shared programs
 - Shared transactions
 - Shared objects

Value Of Application Service-Oriented Application Integration

- ▶ Provides true code reuse infrastructure for many enterprise applications
- ▶ Availability of technology and expertise
- ▶ Ultimate application integration solution for many trading communities and enterprises

Issues With Application Service-Oriented Application Integration

- ▶ Much more complex and expensive than the other approaches
- ▶ Takes a lot of time, architecture, and planning
- ▶ Enabling technology may not scale to enterprise class applications or fall short in other ways
- ▶ Too invasive for most B2B problem domains

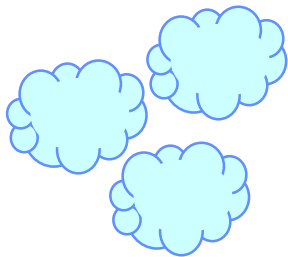
Enabling Technology

- ▶ Application servers
- ▶ TP monitors
- ▶ Distributed objects (CORBA/COM)
- ▶ Traditional development tools
- ▶ Web Services

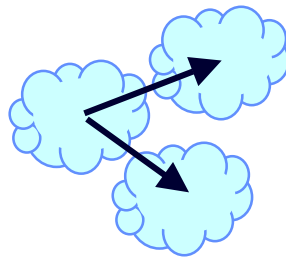
Application Integration is Evolving

Loosely coupled,
independent

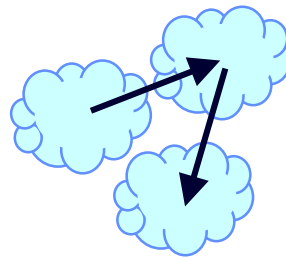
Tightly coupled,
interdependent



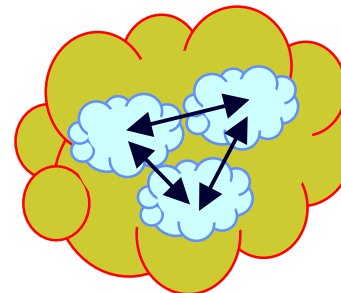
**Autonomous
Monolithic
Applications**



**Data
Consistency**



**Multi-step
process**



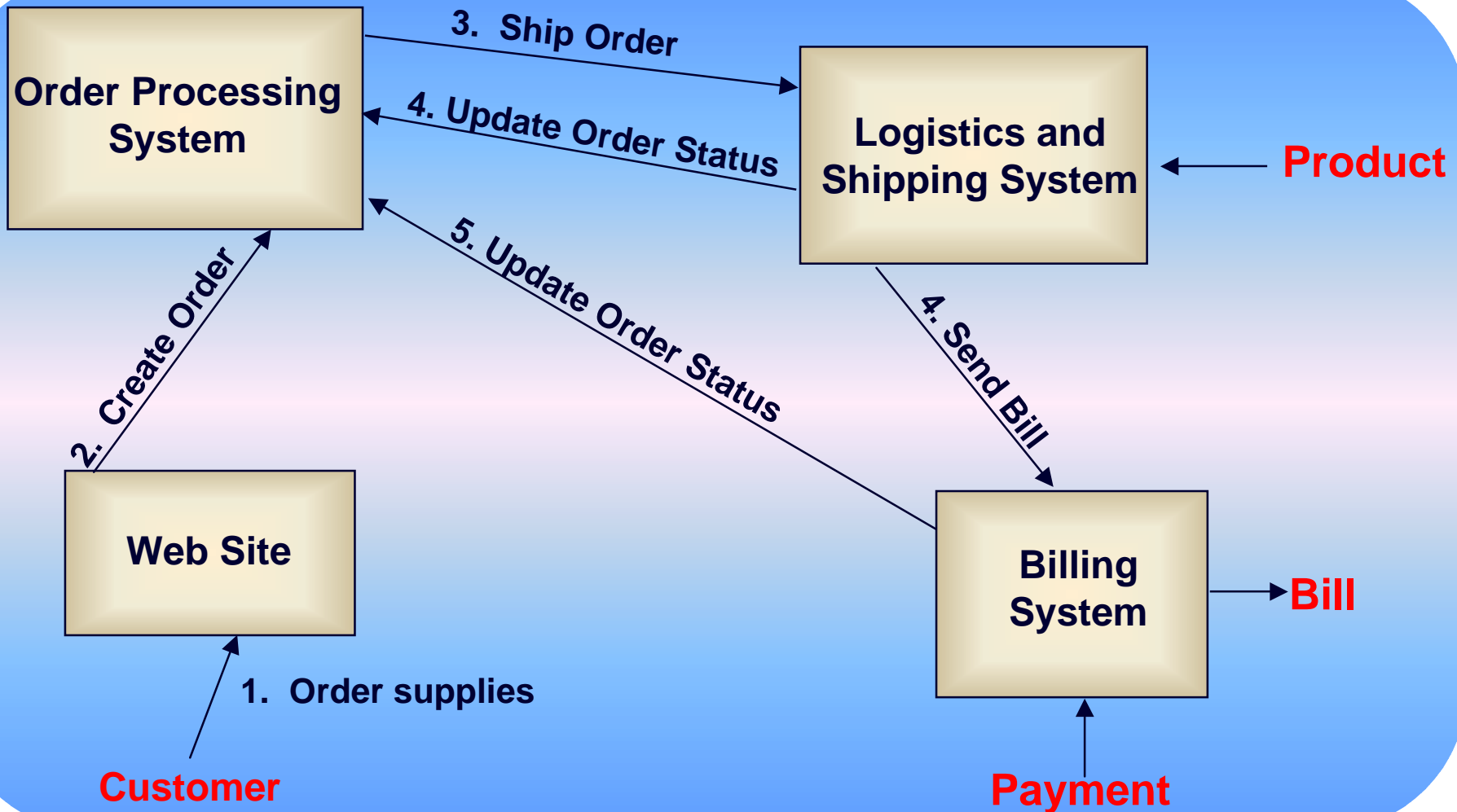
**Composite
Applications
(Web Services)**



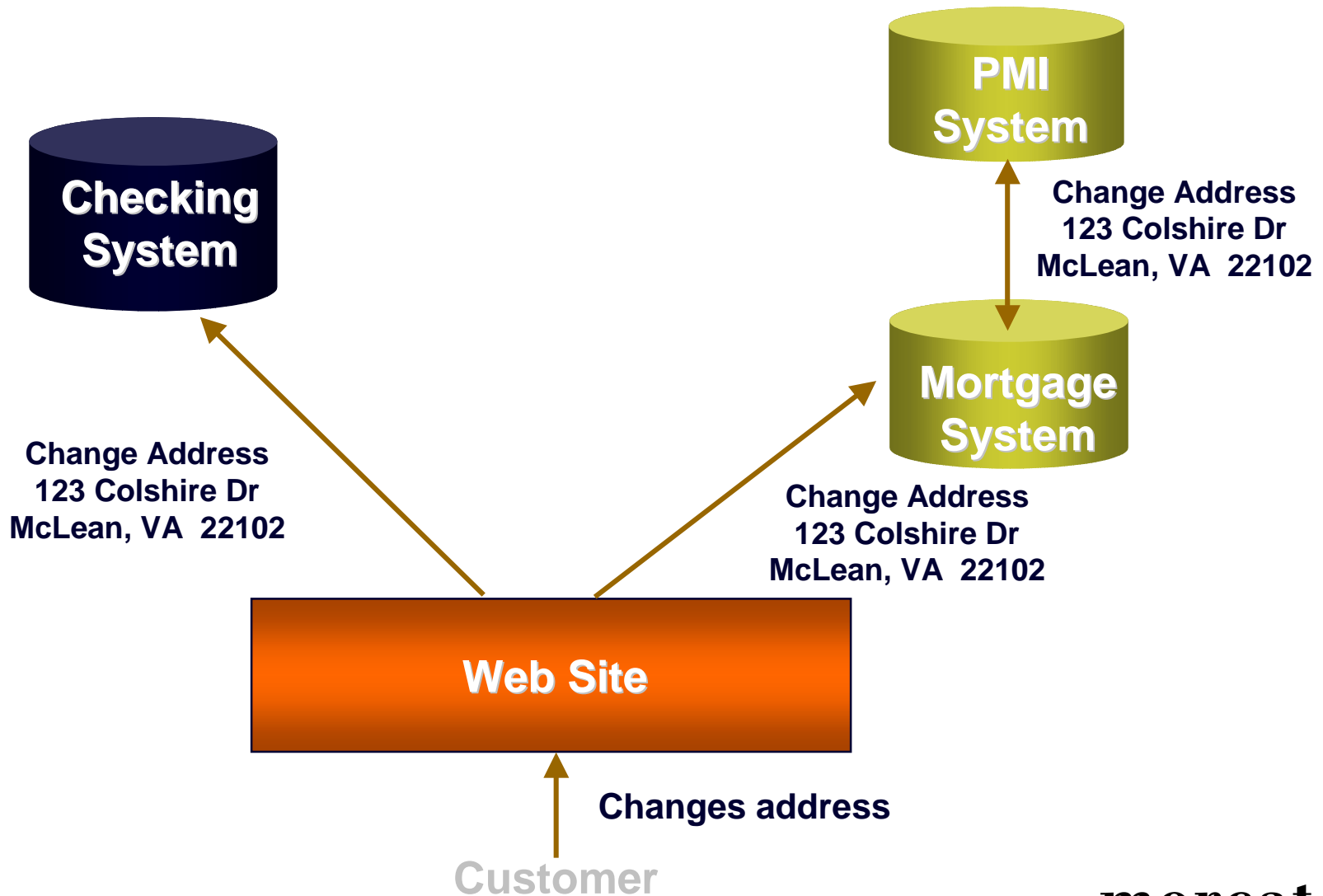
**Autonomous
Distributed
Applications**

Business Process Control

Multistep Process Dominates For Now

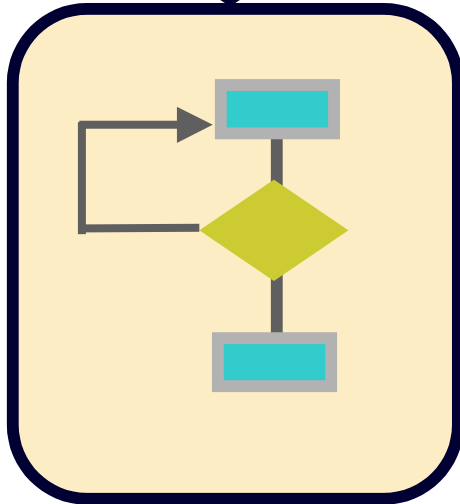
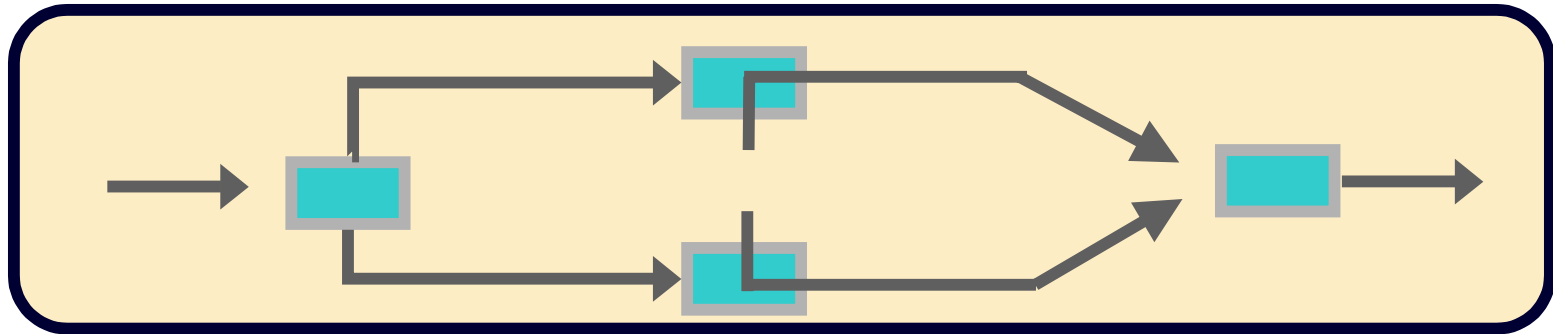


Data Consistency

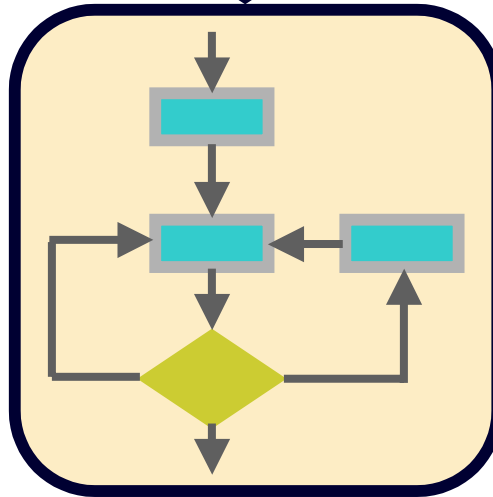


Process Integration Will Define Application Integration

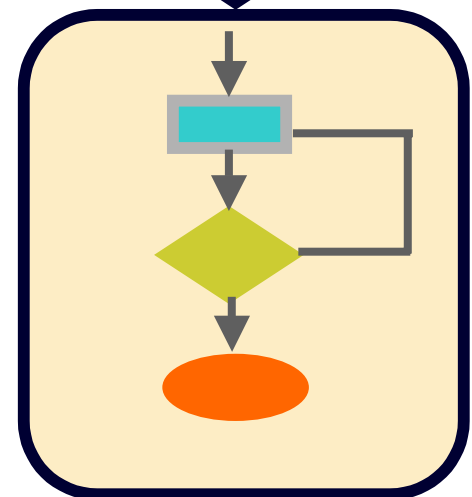
Process Model



System A

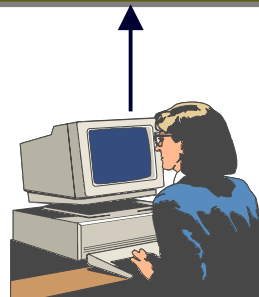
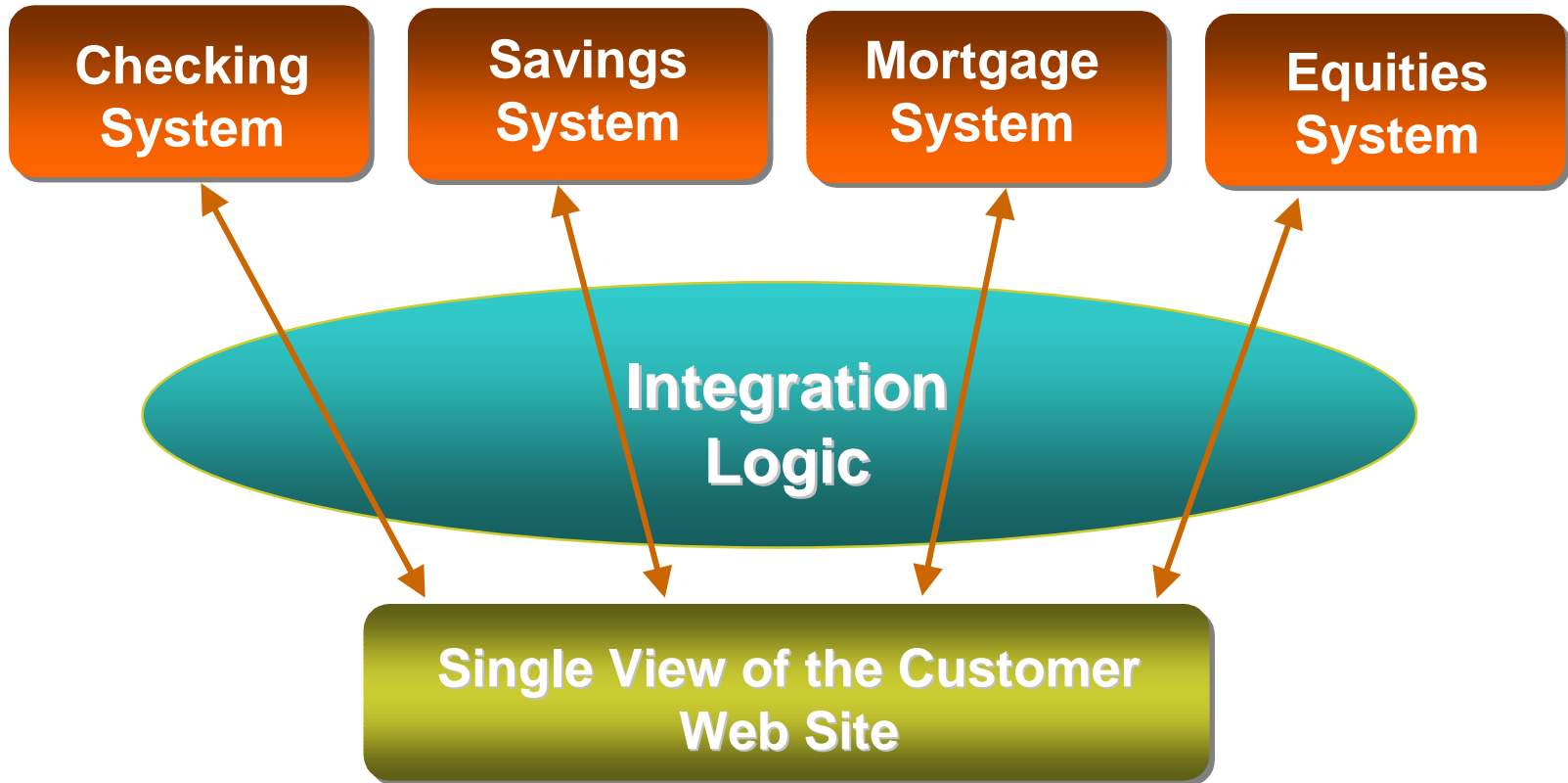


System B



Company C

Composite Application



**Customer
checking on account**

Contrasting Three Forms of Integration

Data Consistency

- Loosely coupled integration
- Multiple applications
- Asynchronous dominates
- One-way
- Simple abstraction of business rules

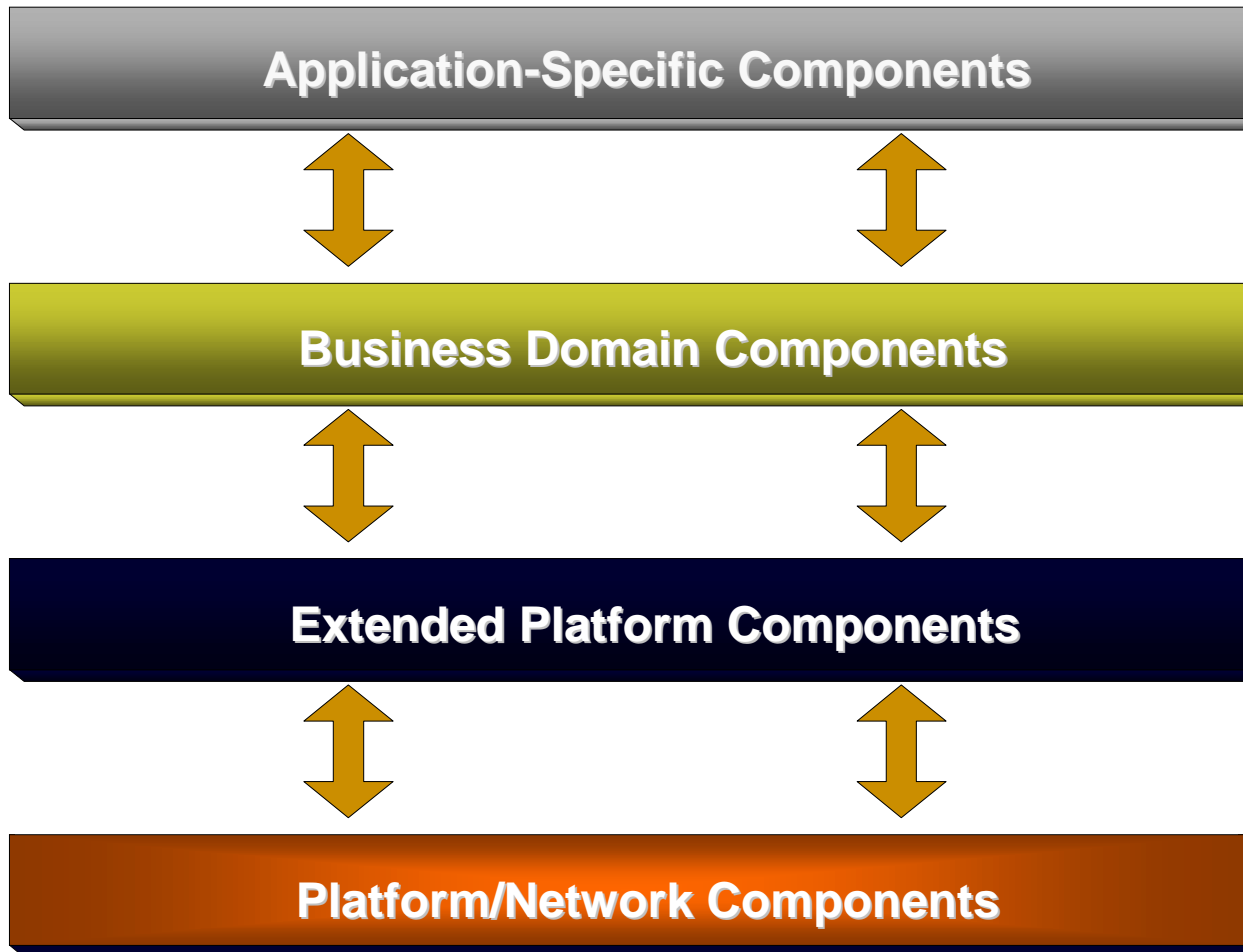
Multi-step Process

- Loosely coupled integration
- Multiple applications
- Can be either Asynchronous or Synchronous based
- One-way and request/reply
- Limited complexity in abstraction of business rules

Composite Application

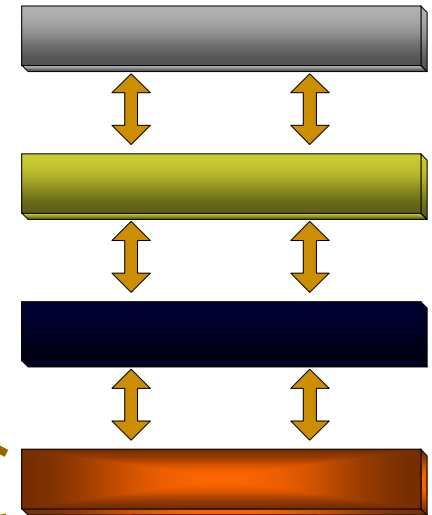
- Tightly coupled integration
- Single application
- Synchronous dominates
- Request/reply
- Complex abstraction of business rules

Enterprise Architecture Framework Layers



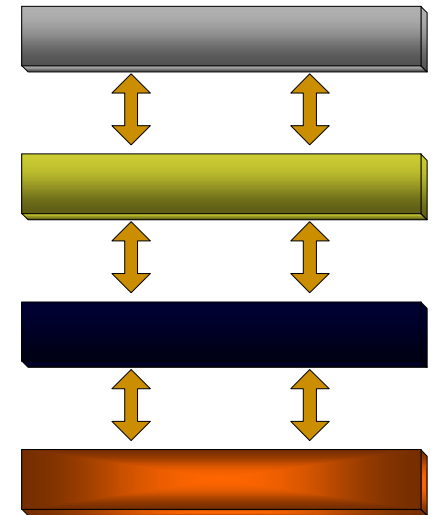
Platform/Network Components

- ▶ Networks
 - Routers
 - Switches
 - Communication Links
 - Firewalls (assumes functionality limited)
- ▶ Processors
- ▶ Operating Systems



► Information Systems Technology Services

- Object Request Brokers (ORBs)
- Database Management Systems
- Web browsers, servers
- Transaction Monitors
- Legacy system bridges
- Directory services
- Application Servers
- Integration servers
- System management
- Custom-written business-independent services



Business Domain

- **Shared business services and information**

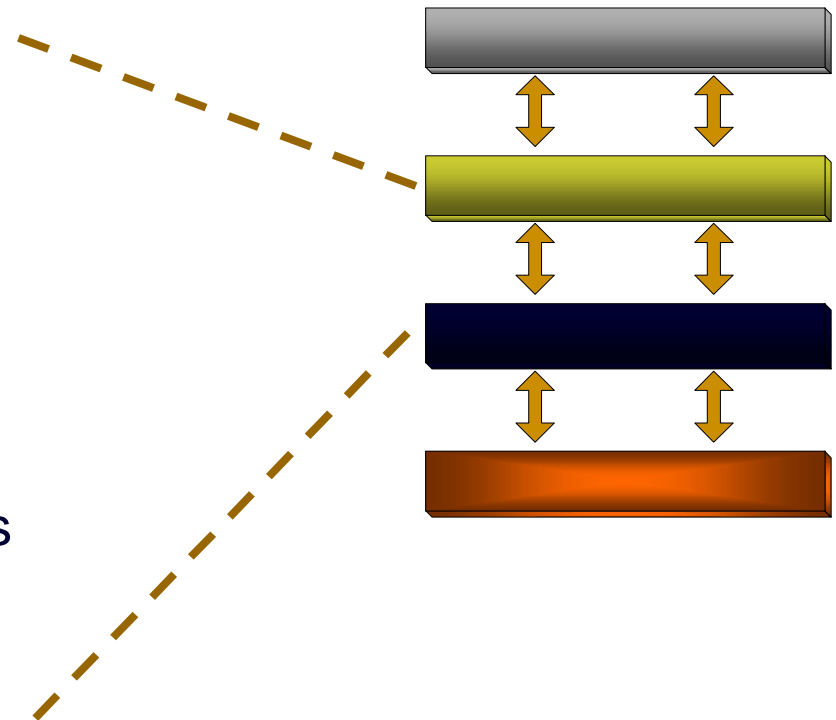
- ▶ Including general-purpose services

- General ledger
 - Office automation

- ▶ Shared stateful run-time services

- ▶ Reusable code components

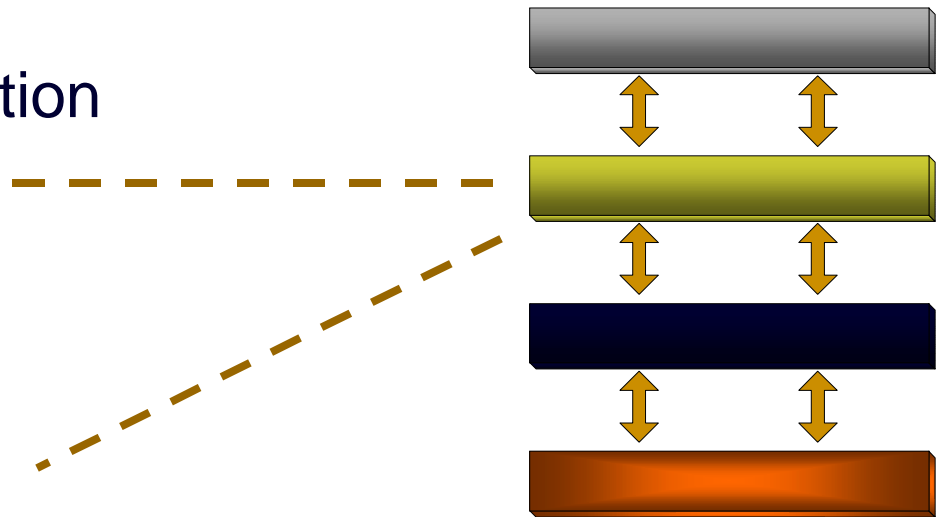
- **Key question: business functionality vs. technical support**



Application Specific

► Dedicated to one application

- GUI
- Server-side code
- Data (not shared)



Components Are Designed for Reuse Across Applications

Home Banking

Call Center

Branch

Application Specific Components

Bill Payment

Open Account

Transfer

Change Address

Customer

Business Domain Components

Data Access to Checking Acct

Data Access to Savings Acct

Distributed Services

Extended Platform Components

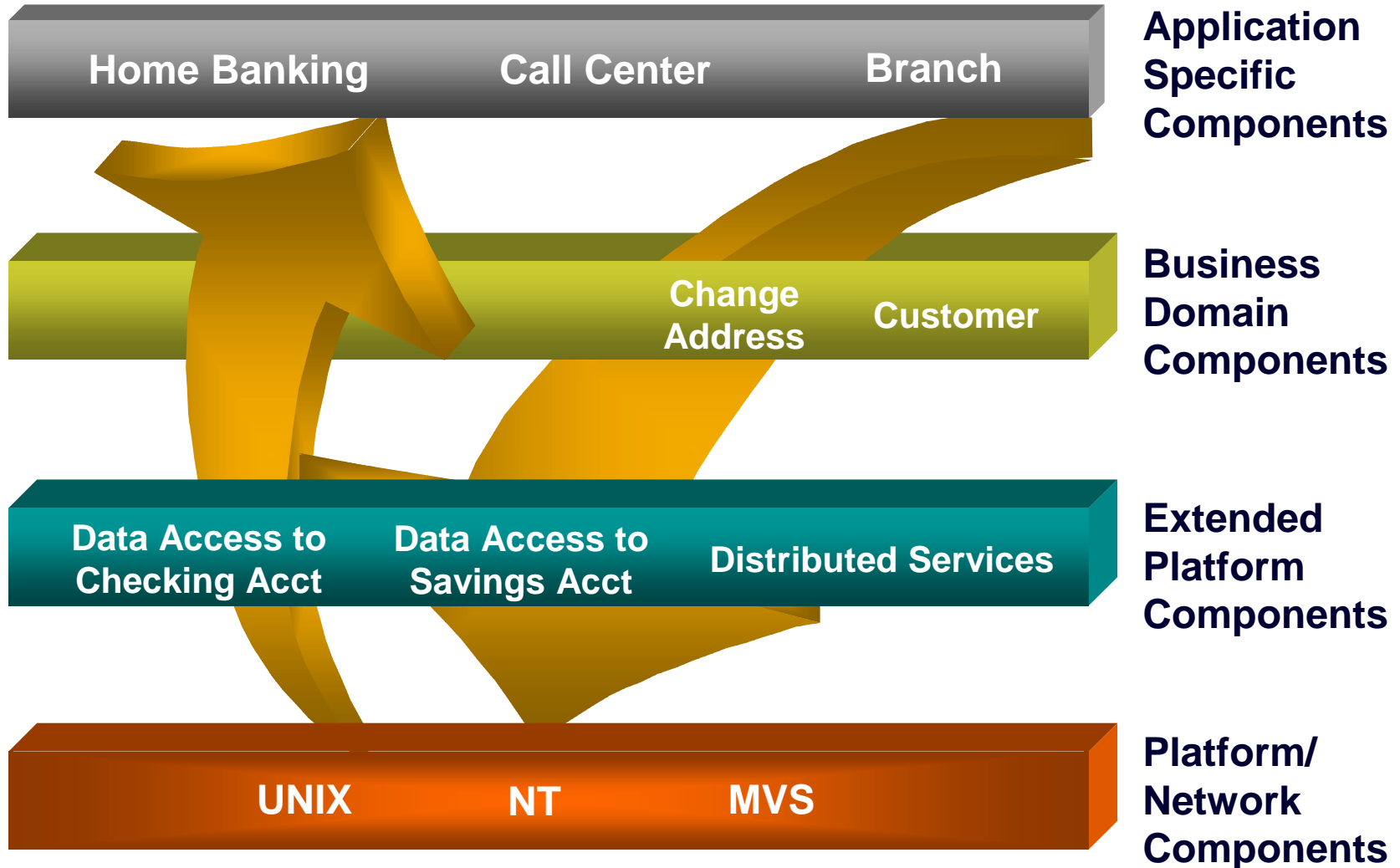
UNIX

NT

MVS

Platform/ Network Components

Go to a Branch Office to Change Address



Technology Considerations

Messaging Oriented
Middleware

Distributed Object
Technology

Object Transaction
Monitor

- ▶ Common user interface
- ▶ Common programming model
- ▶ Common security model
- ▶ Synchronous integration
 - DOT - CORBA, DCOM, Java, etc.
- ▶ Asynchronous integration
 - Message Queuing
 - Publish & Subscribe
- ▶ Transactional integration
 - CICS → TM & OTM
- ▶ Work flow/flow control
- ▶ Web application services
- ▶ Data transformation/translation
- ▶ Component model (containers or composite objects)
- ▶ Connectors (adapters)
- ▶ Integration with systems management frameworks

Integration Tools

Application Servers

Target delivery of legacy information to the web, with the addition of new logic

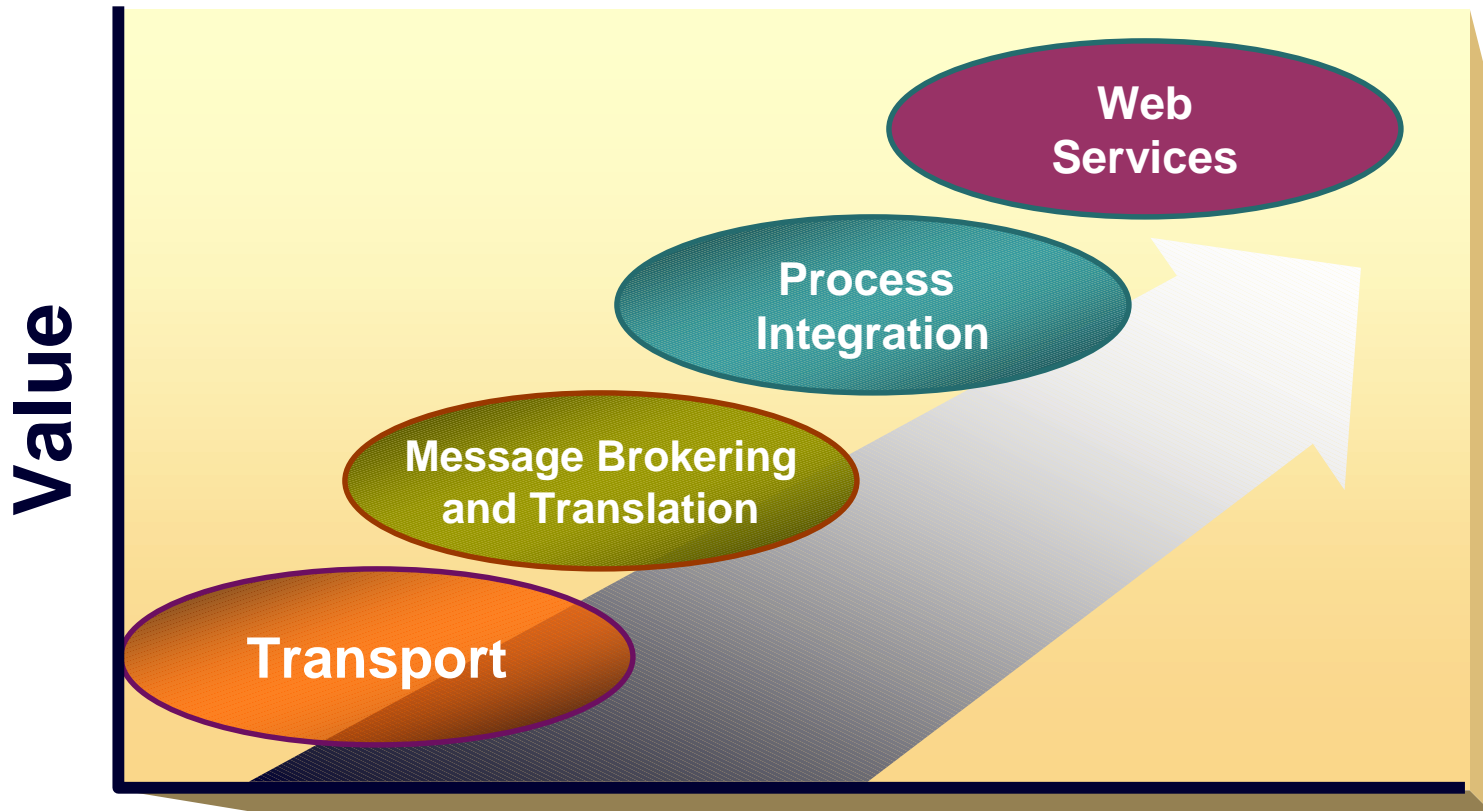
Message Brokers / MOM

Target legacy to legacy integration and transform information between applications

**Object Request Brokers/
Object Transaction Monitors**

Mechanisms that find the object and prepare the implementation for the request, then communicate the data

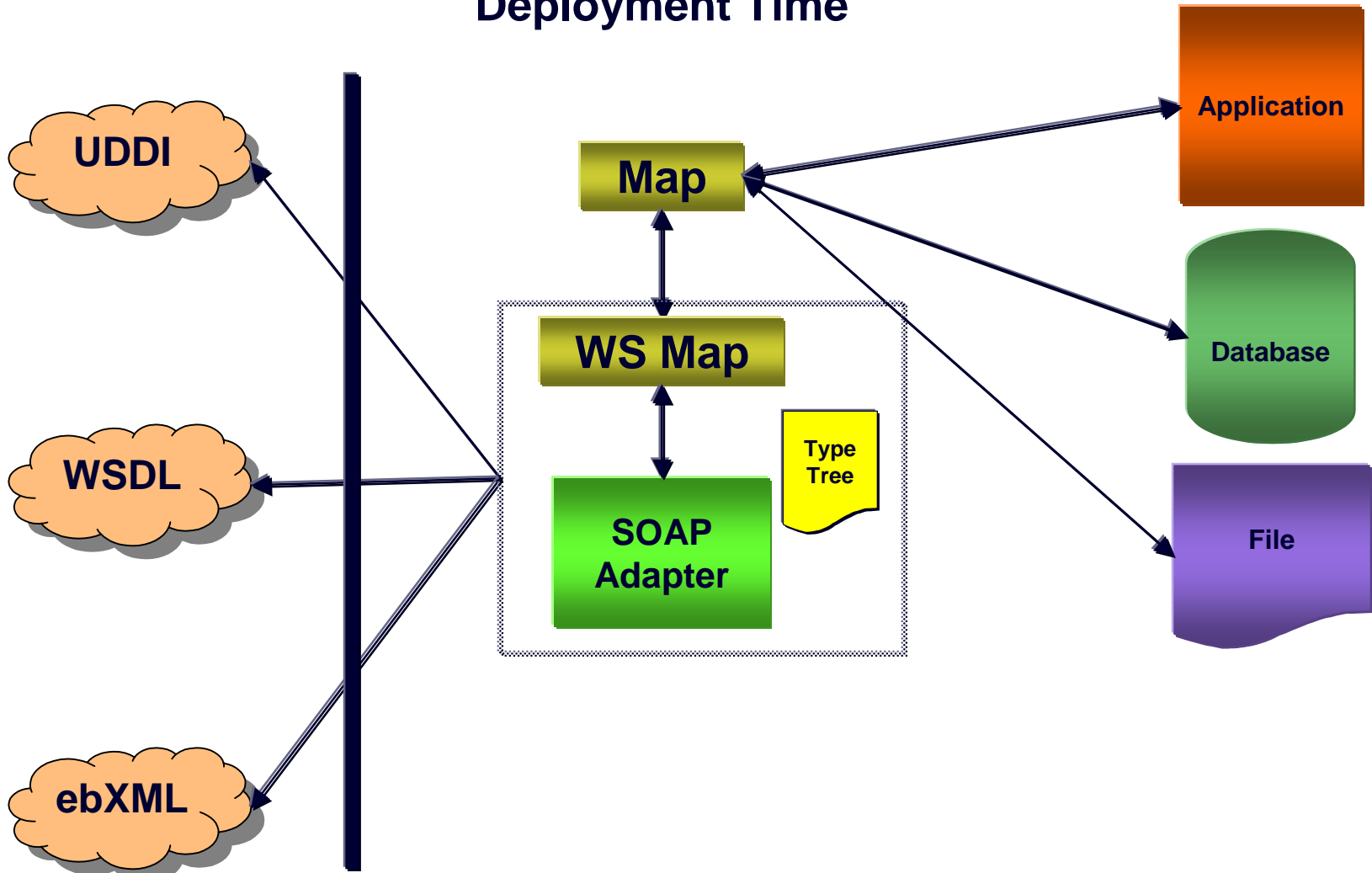
Progression Of Application Integration Solutions (EAI and B2B)



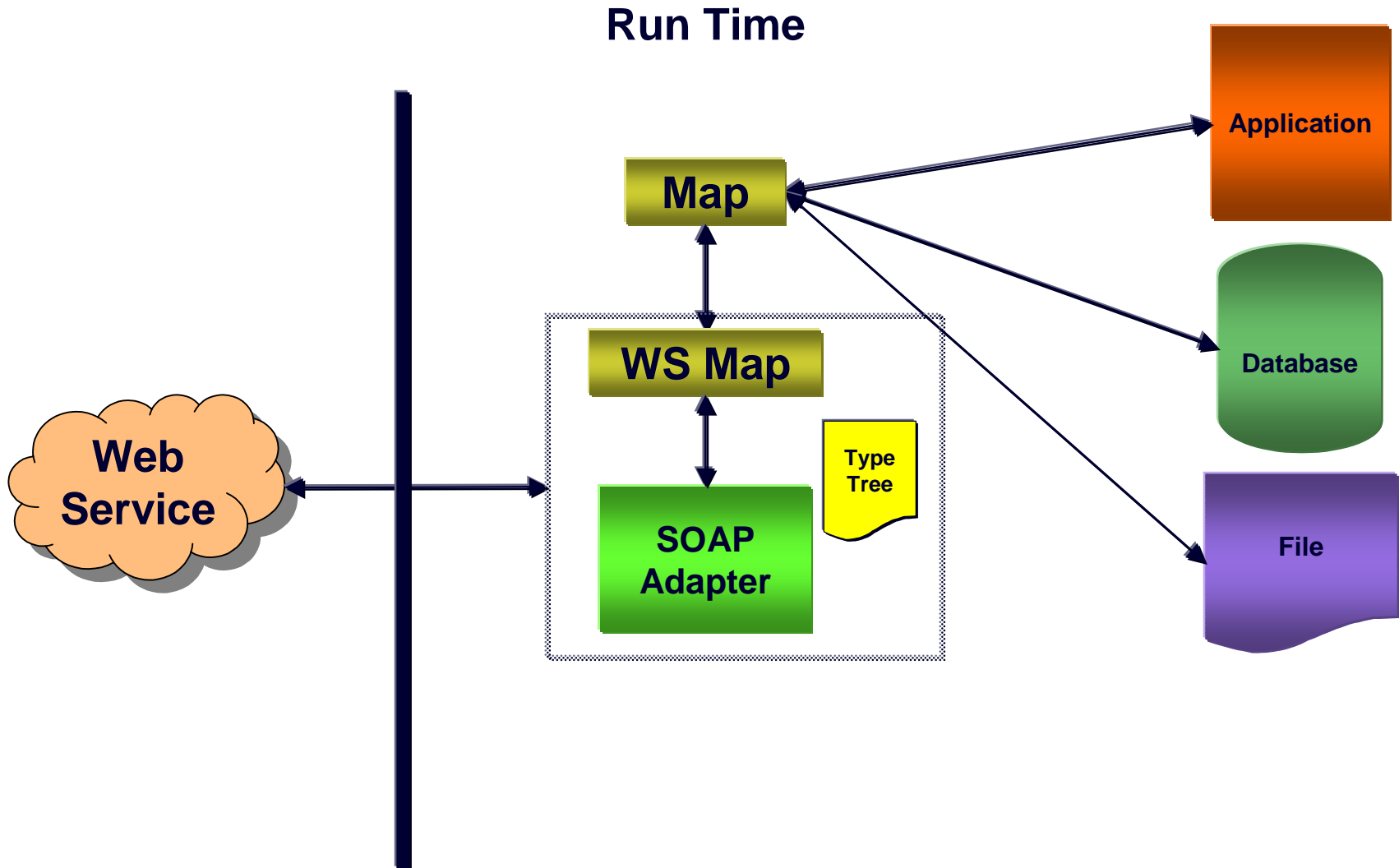
**Application Integration
Implementation Order**

Integration Brokers Invoked As Web Services

Deployment Time

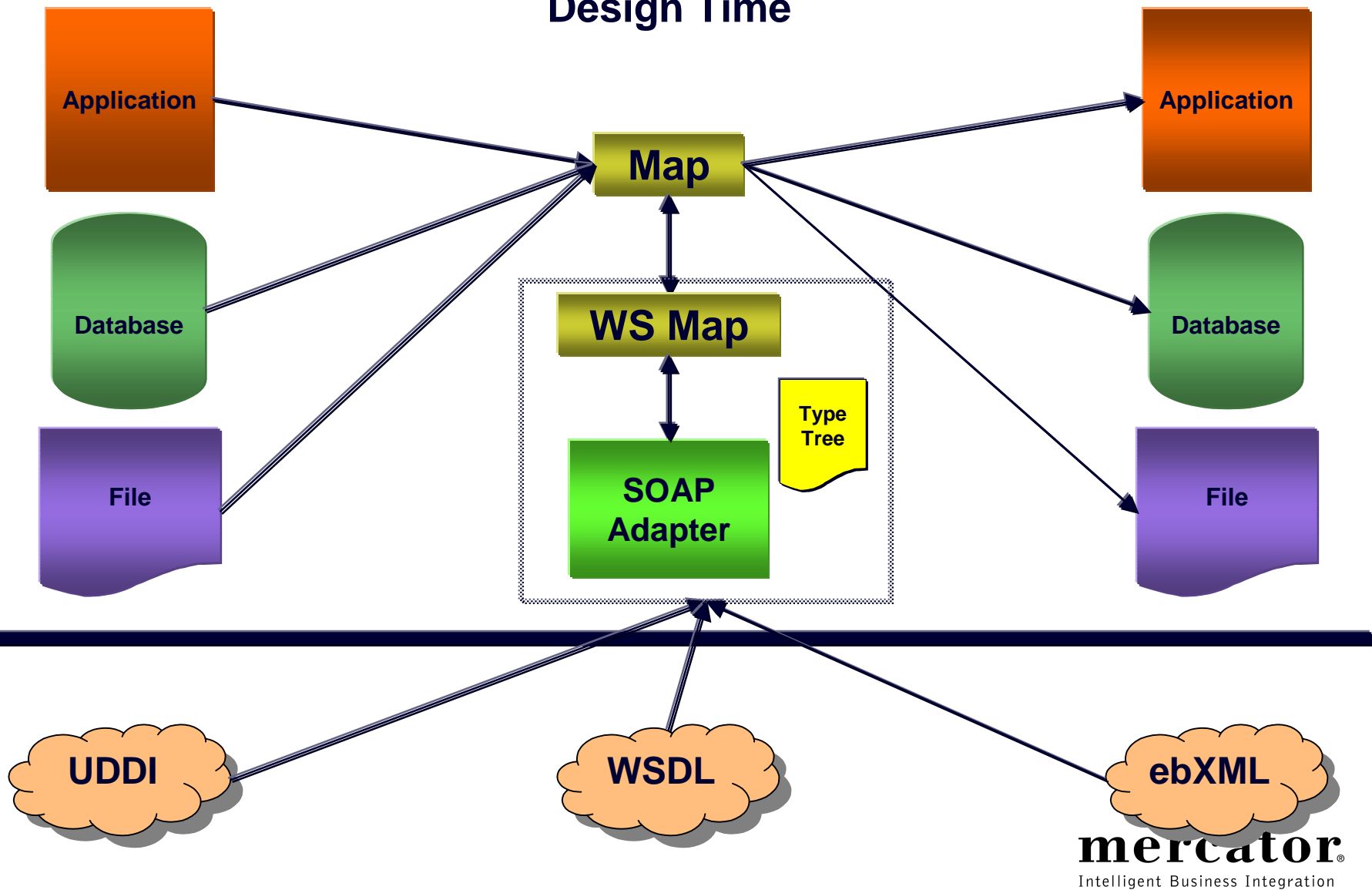


Integration Brokers Invoked As Web Services



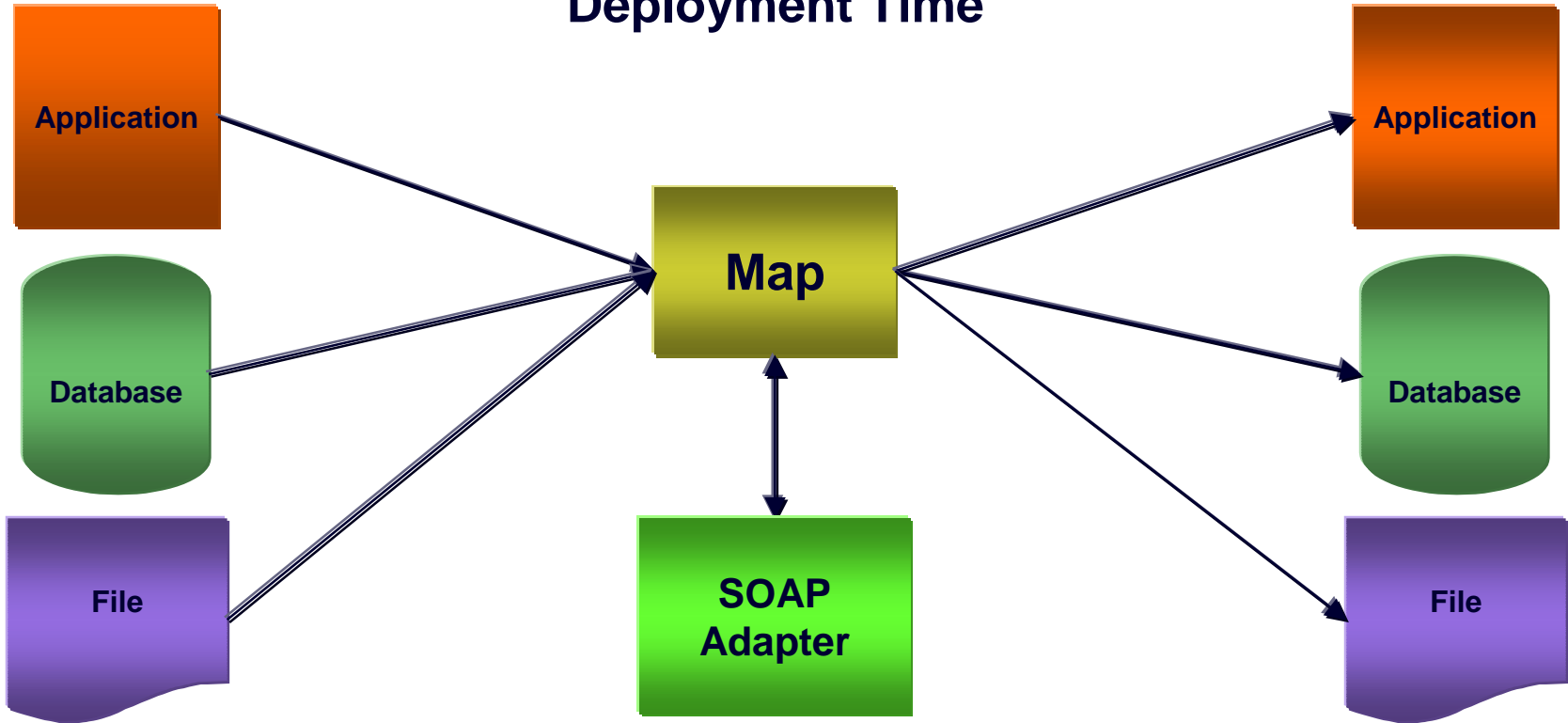
Web Services Invoked by Integration Brokers

Design Time

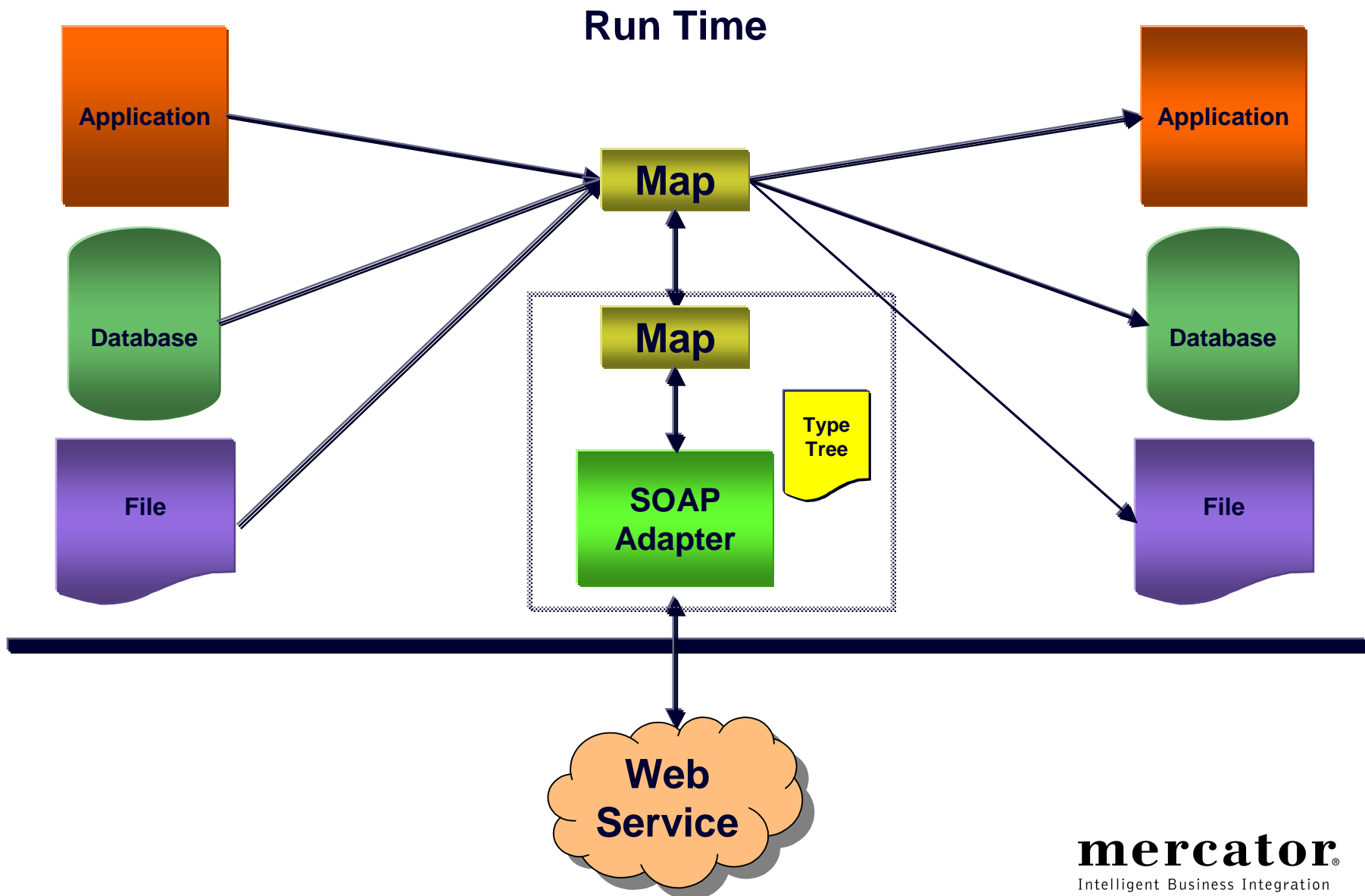


Web Services Invoked by Integration Brokers

Deployment Time



Web Services Invoked by Integration Brokers



Leveraging Web Services for Application Integration

Thank You!