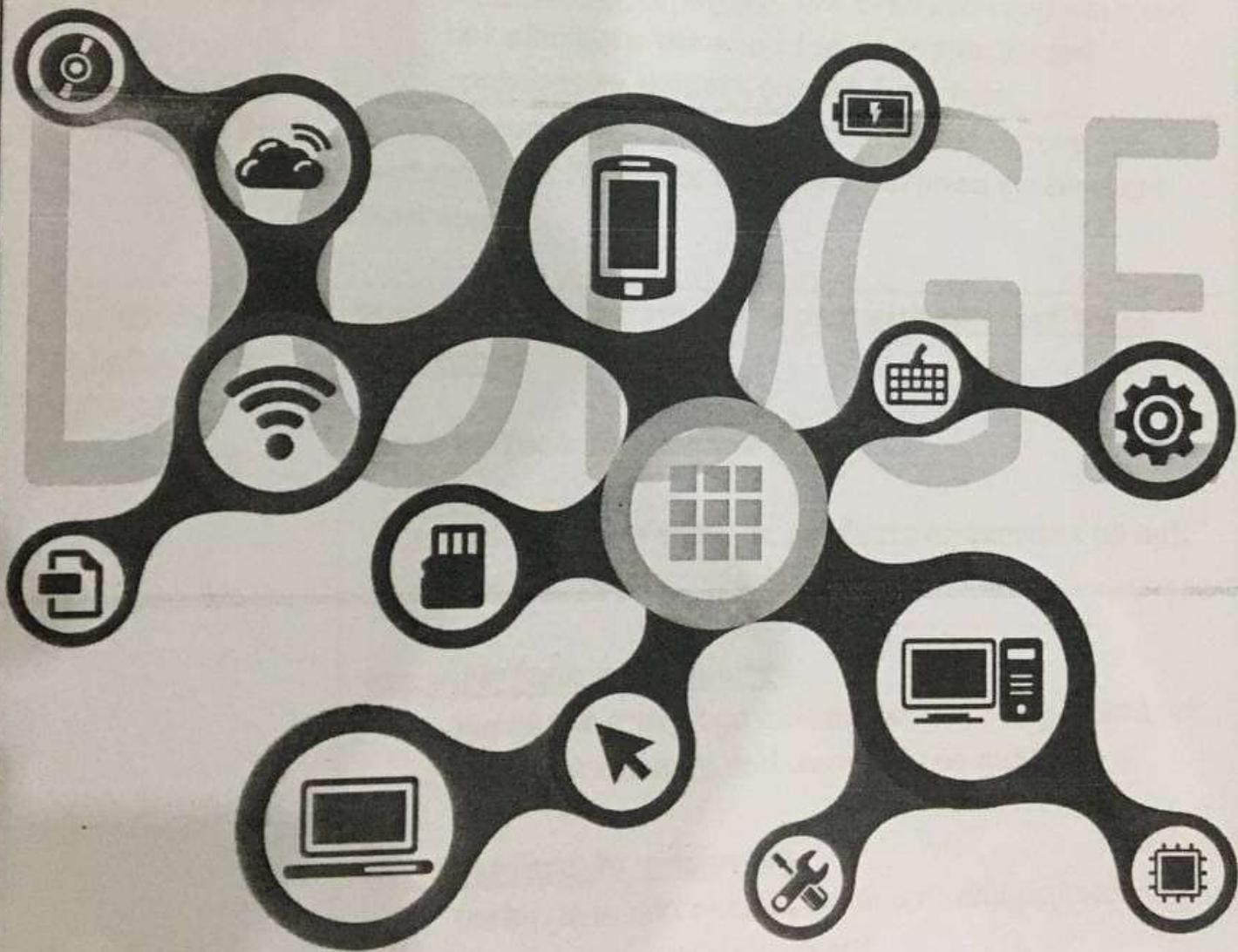




MED REV

2022

System Integration





SYSTEM INTEGRATION

System integration:

This area deals with the problem of

- connecting all systems
- providing asynchronous message exchange between them
- coordinating the execution of business processes on top of an infrastructure comprised of heterogeneous applications.

Enterprise Resource Planning (ERP)

- provide a single system to support a wide range of business functions, but even such approach did not eliminate the need for more specialized packages to support certain functions.

service-oriented

plays a prominent role in the landscape of enterprise systems, and it shapes the way enterprise systems are built today.

1-Essential Systems of a Business Organization

business organization has to deal with at least three interfaces:

the interface to customers :

- sales orders come in; products or services go out; and payments come in.

the interface to suppliers

- purchase orders go out; materials, components, or services come in; and payments go out.

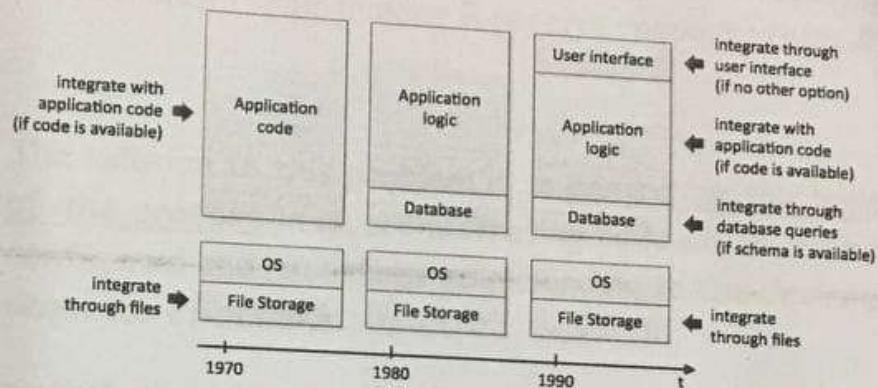
the interface to employees

- tasks go in and results come out; and payments, in the form of salaries, go out.

SYSTEM INTEGRATION

2 Evolution of System Architectures

- it was commonly accepted that every system architecture should comprise three layers:
 - the data layer
 - the application layer
 - the user interface layer



- development of applications where these layers were distributed across the network
- The need for distributed applications led, in turn, to the development and widespread use of distributed object technologies such as CORBA and Java RMI

Integration type :

- flat files**, i.e., text files with a format that is compatible with the target system
- database schema is available, then it may become easier to interact with the system database directly, and in this case to both read and write data to the application in the form of SQL queries



M.D



ST.DRONE

SYSTEM INTEGRATION

3 Integrating Several Applications Together

- A total of 10 links are shown. In 5 application
- $N^*(n-1) \ll$ ad hoc
- However, we should consider that integrating application A with application B is different from integrating B with A, since making A able to receive requests from B may require a different integration mechanism from making B receive requests from A

The solution to this problem is to centralize the logic of the process in an orchestrating node which can be configured and reconfigured according to the desired sequence of actions

- this approach also reduces the number of required connections between applications.

The orchestrator

- sends requests to each application queue in turn and waits for reply.
- The response comes back to the orchestrator in the same way as to any other application, through the use of its own message queue.
- 5 application >> 5 links

1.4 Services: The Ultimate Solution?

Several technologies have been developed for this purpose, such as :

- Remote Procedure Calls (RPC)
- the Common Object Request Broker Architecture (CORBA)
- and Java Remote Method Invocation (Java RMI).



SYSTEM INTEGRATION

service

is a self-contained block of functionality with a well-defined interface expressed in a standard format

Mcq

1 - in 1970 the system are composed of layer

- a) 2
- b) 3
- c) 4
- d) 5

2-in 1980 the system are composed of layer

- a) 2
- b) 3
- c) 4
- d) 5

3-in 1990 the system are composed of layer

- a) 2
- b) 3
- c) 4

4-after the rise of the internet a these layers were distributed across

- a) Network
- b) Technology
- c) Cobra
- d) Java rmi



5-if neither the data layer nor the application layer is accessible , it may be possible to integrate via

- a) User interface layer
- b) Flat files
- c) Db
- d) Non-supported

6-the use of services involves

- a) Service provider which offers a service implementation
- b) Service requester which invoke and users the services
- c) Both a, b
- d) Non of the above

7- implement business process on top of heterogenous infrastructure

- a) An orchestrating
- b) Message plateform
- c) Accounting system
- d) Factory



SYSTEM INTEGRATION

9-integration six application in ad hoc way ,...links will be needed

- a) 15
- b) 8
- c) 21
- d) 10

10-integration 7 application in ad hoc way ,...links will be needed

- a) 15
- b) 21
- c) 6
- d) 10

11-integration 5 application in ad hoc way ,...links will be needed

- a) 15
- b) 21
- c) 6
- d) 10

12-in the case of centralizing the logic of the process in an orchestrating node links will be needed if we have six application

- a) 15
- b) 21
- c) 6
- d) 10

13- is a self-contained block of functionality with a well-defined interface expressed in a standard format

- a) Network
- b) service
- c) Cobra
- d) Erp

14- provide a single system to support a wide range of business functions, but even such approach did not eliminate the need for more specialized packages to support certain functions.

- a) Network
- b) service
- c) Cobra
- d) Erp



M.D

schemas

- the first step is to define the data formats
- define what kind of messages will be sent and received
- If the data format used by the source application is different from the one used by the target application, then there are two different schemas

orchestration

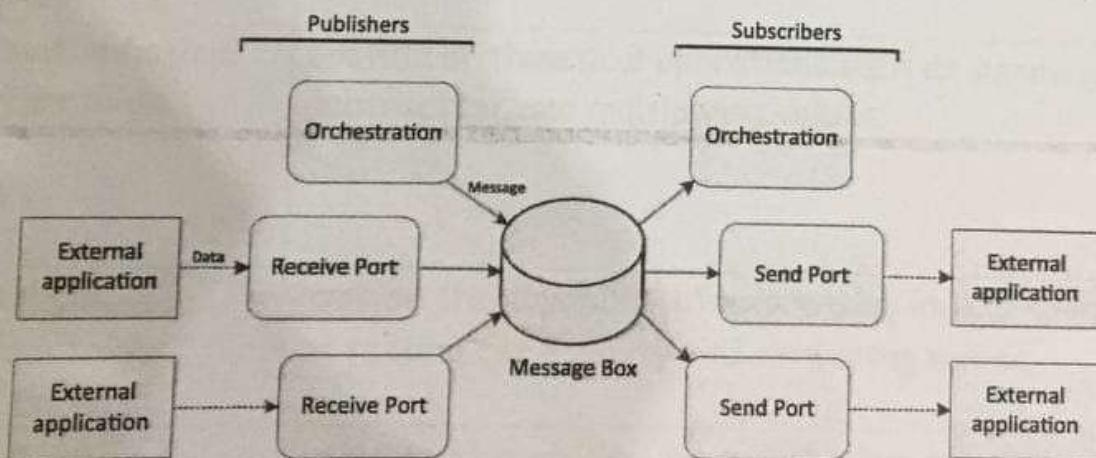
- data flow between all systems that have to be integrated to support a given business process. This data flow and transformation between systems can be specified as an orchestration

port

- The way in which an integration platform connects to an existing system in order to fetch or deliver data

The Message Box

- is a publish-subscribe system that receives messages from publishers and dispatches them to subscribers.





SYSTEM INTEGRATION

<u>a publisher</u>	is usually defined as a <u>receive port</u> through which message box receives messages. It can also be an orchestration intending to dispatch messages to external systems
<u>A subscriber</u>	is usually a <u>send port</u> through which message box dispatches messages to external applications.
<u>Business Activity Monitoring (BAM)</u>	Is tracking the load and rate of data exchanges between system or content of the messages
<u>Transformation map</u>	<ul style="list-style-type: none"><input type="checkbox"/> define how messages from one schema can be transformed into another schema<input type="checkbox"/> All messages that enter or exit the message box needed to have a known format be based on <u>xml</u>
<u>functoids</u>	applying mathematical, string, and other kinds of operators
<u>string functoids :</u>	to concatenate strings, to extract part of a string, etc
<u>mathematical functoids</u>	provide arithmetical operations such as summing, subtracting, and multiplying values;
<u>logical functoids:</u>	provide the capability of expressing logical conditions such as testing for equality and comparing values;
<u>advanced functoids</u>	provide the ability to count elements, loop through elements, and so on; and many other types of functoids are available as well



SYSTEM INTEGRATION

Pipelines

- define how messages can be encoded or decoded to fit the requirements of external applications

can be used for several purposes, namely:

- to validate messages;
- to encode and to decode messages;
- to divide a message into several parts;
- to convert between XML and plain text;
- to encrypt or to decrypt messages;
- to digitally sign or to verify the signature in a message.

2.3 Ports, Pipelines, and Adapters

21

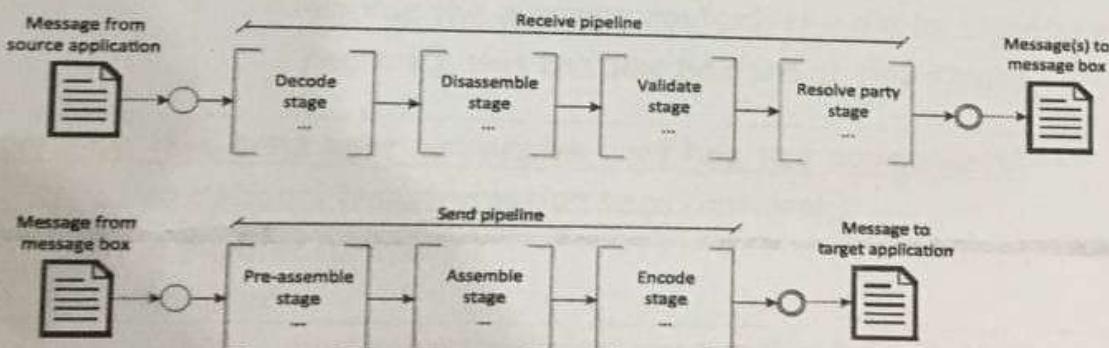


Fig. 2.3 Anatomy of receive and send pipelines

A receive pipeline

- **decode the message** (e.g., if the message arrives by an e-mail protocol such as POP3, it may be necessary to use a MIME or S/MIME decoder to decode the message parts),



SYSTEM INTEGRATION

- disassemble the message (i.e., to split an incoming message into several parts)
- validate the message (by comparing and ensuring that its structure is consistent with that of an existing, previously defined schema),
- resolve the external party (through the use of digital certificates, if the message has been digitally signed this component can verify that signature).

a send pipeline

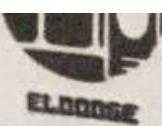
- assemble the message (here, assembling means just converting the message to another format, typically converting an XML message to a plain text format)
- encode the message (if, for example, the message is to be sent by an e-mail protocol such as SMTP, it may be necessary to use a MIME or S/MIME encoder to encode the message content; if a digital signature is required, this can also be done at this stage).

each port be it a send port or receive port has two components:

- an optional transformation maps (optional)
- a pipeline.(primary)
- Adapter

orchestrations

- implement the desired message flow between those applications.
- Created through shapes
- activating receive that triggers the whole orchestration (i.e., it creates a new instance of that orchestration)
- activating receive is the shape that the orchestration begin with



SYSTEM INTEGRATION

1- decode the message

- a) Receive pipeline
- b) Send pipeline
- c) Orchestration
- d) Artifacts

2..... Disassemble the message

- a) Receive pipeline
- b) Send pipeline
- c) Orchestration
- d) Artifacts

3.... validate the message (by comparing and ensuring that its structure is consistent with that of an existing, previously defined schema)

- a) Receive pipeline
- b) Send pipeline
- c) Orchestration
- d) Artifacts

4..... encode the message

- a) Receive pipeline
- b) Send pipeline
- c) Orchestration
- d) Artifacts

5..... resolve the external party

- a) Receive pipeline
- b) Send pipeline
- c) Orchestration
- d) Artifacts

6-..... assemble the message

- a) Receive pipeline
- b) Send pipeline
- c) Orchestration
- d) Artifacts

7- triggers the whole orchestration (i.e., it creates a new instance of that orchestration)

- a) Send port
- b) Receive port
- c) Activating receiving
- d) Receive shape

8- activating receive is the shape that the orchestration with

- a) Begin
- b) End
- c) Does not concern
- d) Not include



M.D



SYSTEM INTEGRATION

9-receive ports may contain several

- a) Receive locations
- b) Receive port
- c) Activating receive
- d) Receive shape

10- orchestrations created from

- a) shapes
- b) Receive port
- c) Activating receive
- d) Receive shape

The fundamental concepts:

- Channel
- Message
- Pipeline
- Routers
- Translator
- Endpoints

Channels:

there may be different channels for different

- channels correspond to the message queues available or created in the messaging system.
- channel for dropping off invalid messages
- [dead-letter channel]:**
 - o drop off messages that could not be delivered
- [the dead-letter channel]:**
 - o It is the job of an administrator to keep monitoring to check if there were messages that could not be delivered.
- a channel adapter may be required. The purpose of this adapter is to read data from the application and publish messages in a channel (when sending), and to receive messages from the channel and write data to the application (when receiving).

bridge channels

- that both the sending and the receiving application appear to be using a common channel when in fact it is a set of channels bridged, or connected, together.

SYSTEM INTEGRATION

2-Messages

- A message can be seen as a package of data that travels through a channel.
- the message may carry a document, a command, or an event.
- A message typically has :**
 - **header** (in a mail system, this can be seen as the **envelope**)
 - **body** (its actual content).
- The message header** contains metadata such as information about the sender, the destination, and the message priority

content-based routing

- determine **the routing of messages based on the actual content of the message body**

Pipelines

- splitting** and **aggregating messages**. These concepts are equivalent to disassembling and assembling messages

Splitting

- means creating several messages out of a single message, by dividing the original message in parts.

Aggregation

- wraps up those parts as a single output message

There are therefore three options for routing messages between applications.

- **the sending application** specifies the recipient, in which case the routing logic is embedded in the **sender**;
- **decides the recipient**, in which case the routing logic is embedded in the system.
- **have no routing logic at all**, and the messaging system just broadcasts messages to every possible recipient.

Mcq

1- A channel for dropping off invalid messages

- a) dead-letter channel
- b) aChannel adapter
- c) bridge channels
- d) channel

2- It is the job of an administrator to keep monitoring of

- a) dead-letter channel
- b) aChannel adapter
- c) bridge channels
- d) channel

3- both the sending and the receiving application appear to be using a common channel when in fact it is a set of channels bridged, or connected, together.

- a) dead-letter channel
- b) aChannel adapter
- c) bridge channels
- d) channel

4- contains metadata such as information about the sender

- a) Header
- b) Body



M.D

- c) Channel
- d) Component

5-if the message is not delivered within the specified time frame , it ends up being

- a) Expiry date
- b) Body
- c) Channel
- d) Component

6- system determine the routing of messages based on the actual content of the message body

- a) Content - based routing
- b) Header routing
- c) Bridge routing
- d) Channel routing

7-information about the sender , destination , and the message priority

- a) Header
- b) Body
- c) Channel
- d) Component

8-removing duplicate message :

- a) De duping
- b) A channel adapter
- c) Bridge channel
- d) Channel



SYSTEM INTEGRATION

9-are comprised of a linear sequence of stage

- a) De duping
- b) A channel adapter
- c) Bridge channel
- d) Pipeline

10 - splitting and aggregating messages are equivalent to Messages ,respectively

- a) Disassembling and assembling
- b) Assembling and disassembling
- c) Non of the above

Routers

- routers are the components of a messaging system that decide the ultimate destination of messages.
- it becomes useful to have a special component called a message filter.

translators

provide the ability to convert message content from one structure into another

content filter

only the minimal required data are transferred to the target message

content enricher

Augmenting a target message with additional data that is not available in the source message

polling, the application

invokes a synchronous method on the messaging system, and this method blocks until a new message arrives in the channel.

callback

it is the messaging system which invokes a synchronous method on the application to deliver a newly arrived message

Message Acknowledgments

is confirmation that a message was successfully received at its destination.

Message Correlation:

there needs to be a mechanism to allow applications to correlate messages, such that they will know that a message that has just arrived is related to another message that was previously sent or received.

Message-broker

control how applications will interact with each other. Rather than letting each application decide the destination for a message



SYSTEM INTEGRATION

is a composite system that comprises a messaging platform and an orchestrator. Using a message broker, it is possible to integrate applications at two different levels:

- at the level of the messaging platform,
- at the level of the orchestrator

the level of the messaging platform,

One way is to use the messaging platform alone and manually configure a set of publish-subscribe rules in the messaging platform. These rules are *static* and, if needed, they must be changed by a system administrator.

an orchestration

the publish-subscribe rules will be configured automatically in the messaging platform when the orchestration is *deployed*.

Deployment

means installing and configuring the orchestration in the message broker, both in the orchestrator and in the messaging platform, so that everything is prepared for the orchestration to run

a message filter

is a component that can be associated with a receiving application; the application will receive a message only if the message complies with the condition specified in the filter

positive result

will let the message through to the corresponding receiving application

Properties

Promoted Properties are used at the message level to implement message routing based on filters associated with send ports.

Distinguished properties are used at the orchestration level to specify how the orchestration will behave at run-time according to the content of messages.

- orchestrator becomes a subscriber for all messages that the deployed orchestrations are waiting to receive.
- On the other hand, the orchestrator also becomes the producer of messages that are to be delivered to target applications.

- ~~are manually configured~~ a set of
publish - subscribe static rules
- a) Message platform
 - b) Orchestrator
 - c) Both
 - d) None of the above

- 2) message filter is a special kind of
- a) orchestrator.
 - b) Router
 - c) Translator
 - d) Message

To integrate through a one assumes that data base schema is known and that the database schema is known

- a) File
- b) Db
- c) Application layer
- d) The three tier model

The filters which yield ,will let the message go through to the corresponding receiving application.

- a) positive result
- b) negative result
- c) negative/positive result
- d) positive /negative result

The filters which yield ,will not let the message go through to the corresponding receiving application.

- a) positive result
- b) negative result
- c) negative/positive result
- d) positive /negative result

Making properties (that are inside the message body) accessible from the outside is referred to

- a) Property promotion.
- b) a negative acknowledgment
- c) implicit promotion
- d) enricher promotion

are used at the orchestration level

- a) Property promotion.
- b) a negative acknowledgment
- c) implicit promotion
- d) Distinguished properties



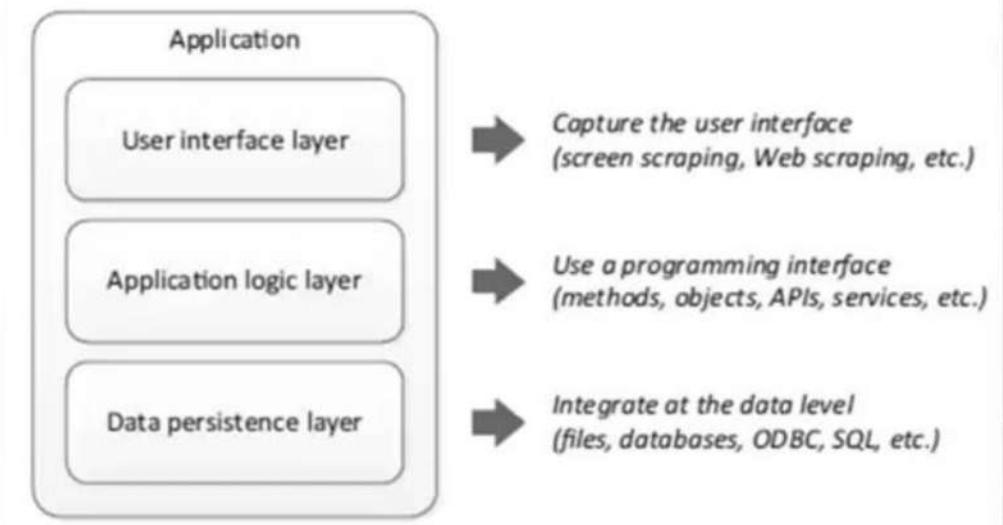
M.D



SYSTEM INTEGRATION

Correlation id	the set of promoted properties that are used to correlate messages is referred to as the
----------------	--

Three -tier modelling



	of messages.				
44.	the set of promoted properties that are used to correlate messages is referred to as the	correlation id _____	a negative acknowledgment	property schema	promoted properties
45.	applications are developed according to a common architecture which is ...	The user interface layer	the data persistence layer	The application logic layer	The Three -Tier Model _____
46.	at the it is a set of data that the user can visualize on the screen and possibly change by means of certain commands	The user interface layer _____	the data persistence layer	The application logic layer	The Three -Tier Model
47.	at the A class instance) with certain data attributes	The user interface layer	the data persistence layer _____	The application logic layer	The Three -Tier Model
48.	it is a row in a database table that stores all purchase orders	The user interface layer	the data persistence layer _____	The application logic layer	The Three -Tier Model
49.	Through ... one assumes that the file format used	files _____	database	The application logic layer	The Three -Tier



SYSTEM INTEGRATION

	data	The user interface layer	the data persistence layer	The application logic layer	The Three -Tier Model
51.	The ideal option is to integrate at the ...	The user interface layer	the data persistence layer	The application logic layer	The Three -Tier Model
52.	integration at the must be based on solid knowledge about how the application processes the data.	The user interface layer	the data persistence layer	The application logic layer	The Three -Tier Model
53.	one must have the original source code for the application to integrate through ...	The user interface layer	the data persistence layer	The application logic layer	The Three -Tier Model
54.	one must have a programming interface to invoke its functionality to integrate through ...	The user interface layer	the data persistence layer	The application logic layer	The Three -Tier Model
55.	where executes program logic	The user interface layer	the data persistence layer	The application logic layer	The Three -Tier Model

DODGE