ALE (Application Linking and Enabling)

Ale Technology is SAP's technology to support distributed yet integrated processes across several SAP systems.

**Distributed Process:**
A distributed process is one in which part of a business process is carried out on one system and part on another. The two systems would exchange data with each other at appropriate points to stay synchronized.

**Need for Distributed Process:**
- Business in Different Geographical Locations.
- Non availability of dedicated network.
- Cultural and language differences in Geographical locations.
- Running of Mission-critical Applications (Like Maintenance downtime etc.).
- Separate up gradation of Modules.

**SAP Distributed environment:**
ALE allows for efficient and reliable communication between distributed processes across physically separate SAP systems.

**Benefits of ALE:**
- Integration with non-SAP systems: ALE architecture allows third party applications to integrate with SAP system.
- Reliable Distribution: Once message type created and the receiver of the message is determined, ALE delivers the message to the recipient. If there is any network problem it will buffer the message and delivers the message once the network is restored. It also ensures that the message is not delivered twice.
- Release Upgrade: Any of the distributed system can be upgraded to the new release of SAP without affecting the functionality. The ALE layer ensures backward compatibility of messages exchanged between systems.

**ALE Architecture:**
It consists of an Outbound process, an Inbound process, and an Exception - Handling process.

**Outbound Process:**
ALE Outbound Process in SAP sends data to one or more SAP Systems. It involves four steps.

1. **Identify the need of IDoc:** This step starts upon creating an application document, can relate to a change to a master data object.
2. **Generate the Master IDoc:** The document or master data to be sent is read from the database and formatted into an IDoc format. This IDoc is called as a Master IDoc.
3. **Generate the Communication IDoc:** The ALE Service layer generates a separate IDoc from the Master IDoc for each recipient who is interested in the data. Separate IDocs are generated because each recipient might demand a different version or a subset of the Master IDoc. These recipient-specific IDocs are called Communication IDocs and are stored in the database.
4. **Deliver the Communication IDoc:** The IDoc is delivered to the recipients using an asynchronous communication method. This allows the sending system to continue its processing without having to wait for the destination system to receive or process the IDoc.

**Inbound Process:**
The inbound process receives an IDoc and creates a document in the system.

1. **Store the IDoc in the database:** The IDoc is received from the sending system and stored in the database. Then the IDoc goes through a basic integrity check and syntax check.
2. **Invoke the Posting Module:** The control information in the IDoc and configuration tables are read to determine the posting program. The IDoc is then transferred to its posting program.
3. **Create the Document:** The posting program reads the IDoc data and then creates a document in the system. The results are logged in the IDoc.

**IDoc Type:**
An IDoc is an instance of an IDoc Type and consists of three types of records.

**One Control record:** each IDoc has only one control record. The control record contains all the control information about an IDoc, including the IDoc number, the sender and recipient information, and information such as the message type it represents and IDoc type. The control record structure is same for all IDocs.

**One or Many Data records:** An IDoc can have multiple data records, as defined by the IDoc structure. Segments translate
One or Many Status records: An IDoc can have multiple status records. Status record helps to determine whether an IDoc has any error.

Message in IDoc Type:
A message represents a specific type of document transmitted between two partners.

Outbound Process in IDocs:
Outbound process uses the following components to generate an IDoc. A customer model, and IDoc structure, selection programs, filter objects, conversion rules, a port definition, an RFC destination, a partner profile, service programs, and configuration tables.

The Customer Model:
A customer model is used to model a distribution scenario. In a customer model, you identify the systems involved in a distribution scenario and the message exchanged between the systems.

Message control:
Message control is a cross-application technology used in pricing, account determination, material determination, and output determination. The output determination technique of Message control triggers the ALE for a business document. Message control separates the logic of generating IDocs from the application logic.

Change Points:
The change points technique is based on the change document technique, which tracks changes made to key documents in SAP, such as the material master, customer master and sales order. Changes made to a document are recorded in the change document header table CDHDR, and additional change pointers are written in the BDCP table for the changes relevant to ALE.

IDoc Structure:
A message is defined for data that is exchanged between two systems. The message type is based on one or more IDoc structures.

Selection Program:
Selection programs are designed to extract application data and create a master IDoc. A selection program exists for each message type. A selection program's design depends on the triggering mechanism used in the process.

Filter Objects:
Filter Objects remove unwanted data for each recipient of the data basing on the recipient's requirements.

Port Definition:
A port is used in an outbound process to define the medium in which documents are transferred to the destination system. ALE uses a Transactional RFC port, which transfers data in memory buffers.

RFC Destination:
The RFC destination is a logical name used to define the characteristics of a communication link to a remote system on which a function needs to be executed.

Partner Profile:
A partner profile specifies the components used in an outbound process (logical name of the remote SAP system, IDoc Type, message type, TRFC port), an IDoc's packet size, the mode in which the process sends an IDoc (batch versus immediate), and the person to be notified in case of error.

Service Programs and Configuration Tables:
The outbound process, being asynchronous, is essentially a sequence of several processes that work together. SAP provides service programs and configuration tables to link these programs and provide customizing options for an outbound process.

Process flow for Distributing Transactional Data:
Transactional data is distributed using two techniques: with Message control and without message control.

Process flow for Distributing Master Data:
Master data between SAP systems is distributed using two techniques: Stand-alone Programs and Change Pointers.

Triggering the Outbound Process via Stand-Alone Programs:
Stand-alone programs are started explicitly by a user to transmit data from one SAP system to another. Standard Programs for several master data objects exist in SAP. Ex. The material master data can be transferred using the RBDSEMAT program or transaction BD10. The stand-alone programs provide a selection screen to specify the objects to be transferred and the receiving system. After the stand-alone program is executed, it calls the IDoc selection program with the specified parameters.

Triggering the Outbound Process via Change Pointers:
The change pointer technique is used to initiate the outbound process automatically when master data is created or changed. A standard program, RBDMIDOC, is scheduled to run on a periodic basis to evaluate the change pointers for a message type and start the ALE process for distributing the master data to the appropriate destination. The RBDMIDOC program reads the table TBDME to determine the IDoc selection program for a message type.

Processing in the Application Layer:
The customer distribution model is consulted to make sure that a receiver has been defined for the message to be transmitted. If not, processing ends. If at least one receiver exists, the IDoc selection program reads the master data object from the database and creates a master IDoc from it. The master IDoc is stored in memory. The program then calls the ALE service layer by using the function module MASTER_IDOC_DISTRIBUTE, passing the master IDoc and the receiver information.

Processing in the ALE Interface Layer:
Processing in the ALE Layer consists of the following steps:

- Receiver Determination: The determination of the receiver is done through Customer Distribution Model.
- IDoc Filtering: If an IDoc filter is specified in the distribution model for a receiver, values in the filter are compared against the values in the IDoc data records. If a data record does not meet the filter criteria, it is dropped.
- Segment Filtering: For each sender and receiver combination, a set of segments that are not required can be filtered out.
- Field conversion: Field values in data records are converted by using the conversion rules specified for the segment.
- Version change for segments: Segments are version-controlled. A new version of a segment always contains fields from the preceding version and fields added for the new version. Release IDoc type field of the partner profile to determine the version of the segment to be generated.
- Version change for IDocs: IDocs are also version controlled. The version is determined from the Basic Type field of the partner profile.
- Communication IDocs generated: The final IDoc generated for a receiver after all the conversions and filtering operations is the communication IDoc. One master IDoc can have multiple communication IDocs depending on the number of receivers identified and the filter operations performed. IDoc gets the status record with a status code of 01 (IDoc Created).
• **Syntax check performed:** IDoc goes through a syntax check and data integrity validation. If errors found the IDoc get the status of 26
(error during syntax check of IDoc - Outbound). If no errors found the IDoc gets the status 30 (IDoc ready for dispatch - ALE Service).
• **IDoc dispatched to the communication Layer:** In the ALE process, IDocs are dispatched using the asynchronous RFC method, which
means that the sending system does not wait for data to be received or processed on the destination system. After IDocs have been
transferred to the communication layer, they get a status code 01 (Data Passed to Port OK).

**Processing in the Communication Layer:**
To dispatch an IDoc to a destination system, the system reads the port definition specified in the partner profile to determine the destination
system, which is then used to read the RFC destination. The RFC destination contains communication settings to log o to the remote SAP
system. The sending system calls the INBOUND_IDOC_PROCESS function module asynchronously on the destination system and passes the
IDoc data via the memory buffers.

**Inbound process in IDocs:**
An inbound process used IDoc structure, posting programs, filter objects, conversion rules, a partner profile, service programs, and configuration
tables to post an application document from an IDoc.

**Posting Program:**
Posting programs, which are implemented as function modules, read data from an IDoc and create an application document from it. A posting
program exists for each message. Each posting program is assigned a process code. A process code can point to a function module or a work
flow. In the standard program process codes always point to a function module.
Ex. The posting program for message type MATMAS is IDOC_INPUT_MATMAS which has a process code MATM.

**Workflow:**
A workflow represents a sequence of customized steps to be carried out for a process. The workflow management system is used to model the
sequence, identify information required to carry out the steps and identify the person responsible for the dialog steps.

**Partner Profile:**
A partner profile specifies the components used in an inbound process (partner number, message type, and process code), the mode in which
IDocs are processed (batch versus immediate), and the person to be notified in case of errors.

**Process flow for the Inbound process via a Function Module:**
In this process, IDocs are received from another system and passed to the posting function module directly.

1. **Processing in the communication Layer:**
The IDOC_INBOUND_ASYCHRONOUS program, triggered as a result of an RFC from the sending system, acts as the entry point for all inbound
ALE processes. The IDoc to be processed is passed as an input parameter. Control is transferred to the ALE/EDI layer.

1. **Processing in the ALE/EDI Interface Layer:**
   • **Basic integrity check:** A basic integrity check is performed on the control record.
   • **Segment Filtering and conversion:** Filtering out unwanted segments and carry out any required conversion of field values.
   • **Creation of Application IDoc:** The application IDoc is created and stored in the database and a syntax check is performed. If
     there are errors it gets status code of 60 (Error during Syntax check of IDoc - Inbound). At this point a tangible IDoc, which can
     be monitored via one of the monitoring transactions, is created and the IDoc gets status code 50 (IDoc Added).
   • **IDoc Marked ready for Dispatch:** IDoc gets the status code 64 (IDoc ready to be passed to application).
   • **IDoc is passed to the posting program:** The partner profile table is read. If the value of the Processing field is set to Process
     Immediately, the IDoc is passed to the posting program immediately using the program RBDAPP01.

1. **Processing in the Posting Module:**
The process code in the partner profile points to a posting module for the specific message in the IDoc. The posting program implemented as a
function module either calls a standard SAP transaction by using the Call Transaction command for posting the document or invokes a direct input
function module.
The results of execution are passed back via the function module's output parameters. If the posting is successful IDoc gets the status code 53
(Application Document Posted) or it gets status code 51 (Error: Application Document Not Posted).