

# IEC 61499 (I)

Miguel Ángel García Blanco

# IEC61499

- Introducción
- Bloques Funcionales
- Gestión de dispositivo
- Entornos de desarrollo
- Reconfiguración

# IEC 61499

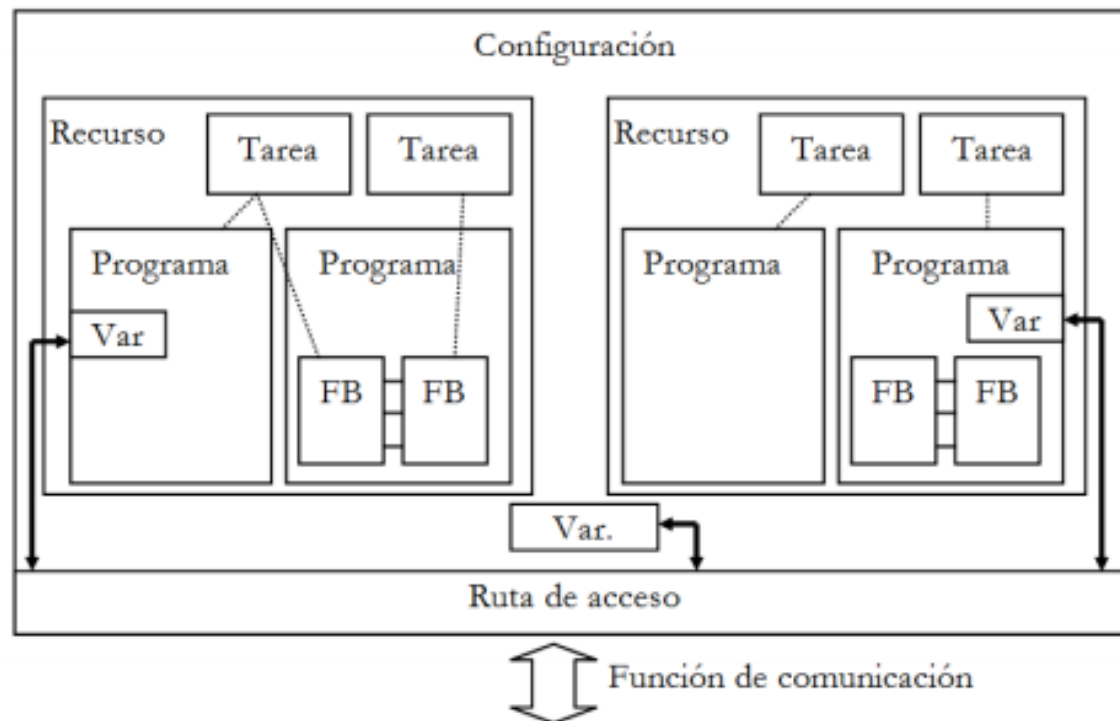
- Nueva familia de estándares de IEC en Sistemas de Control y Medida en Procesos Industriales (IPMCS)
- Su propósito principal es definir Bloques Funcionales (FB)
- Está basado en los lenguajes de programación y modelos definidos en IEC 61131-3
- Extiende los lenguajes y modelos a un entorno de automatización distribuida
- Extensible a RMS (Reconfigurable Manufacturing Systems)

# IEC 61131

- Lenguaje de los PLCs
- Contexto de los FMS (Flexible Manufacturing Systems)
- Cuatro lenguajes de programación:
  - IL, Instruction List: similar a ensamblador
  - ST, Structured Text: similar a Pascal
  - LD, Ladder Diagram: basado en lógica de relés
  - FBD, Function Block Diagram
  - Incluye secuencias SFC, Sequential Function Charts

# Modelos en IEC 61131

- Elementos de alto nivel: Configuraciones, recursos y tareas
- Unidades de organización:; Funciones, FB y programas
- Tipos de datos y variables

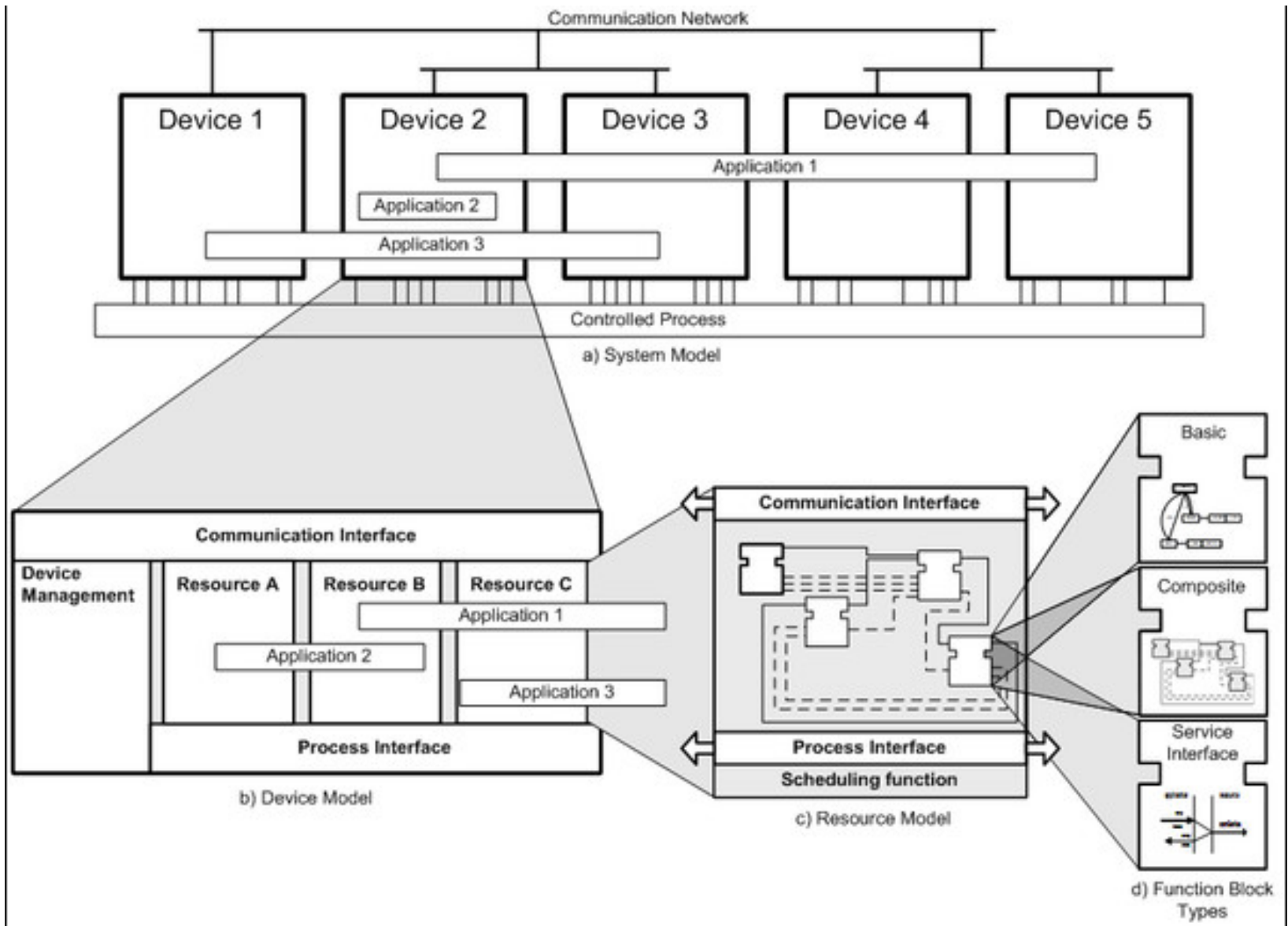


# Modelos en IEC 61499

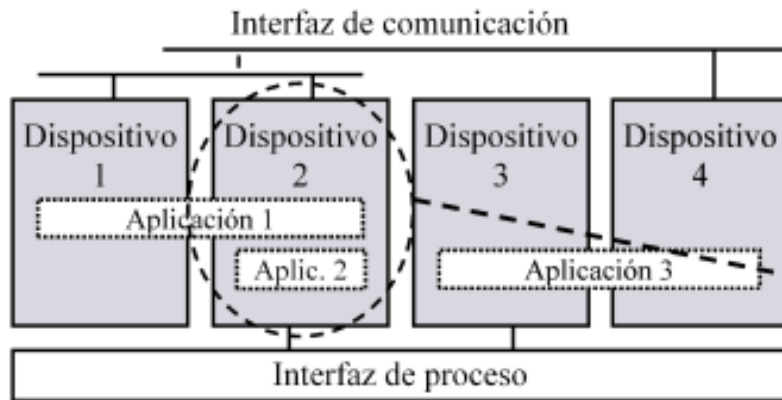
- Aplicación
- Sistema
- Dispositivo
- Recurso
- Bloque Funcional (FB)

Permiten desarrollar aplicaciones de control distribuido de forma gráfica

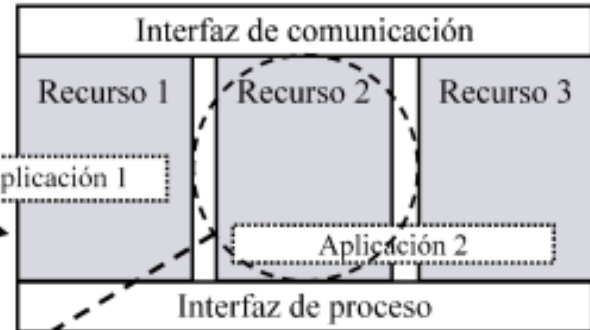
# Modelos en IEC 61499



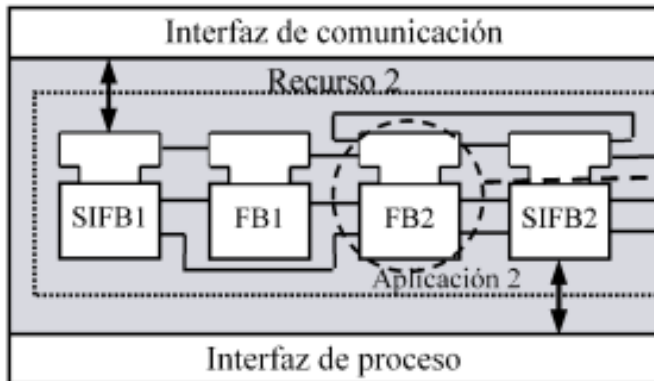
# Modelos en IEC 61499



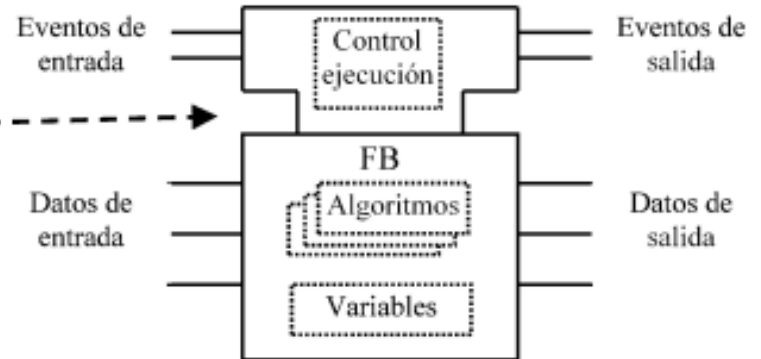
a) Sistema



b) Dispositivo



c) Recurso

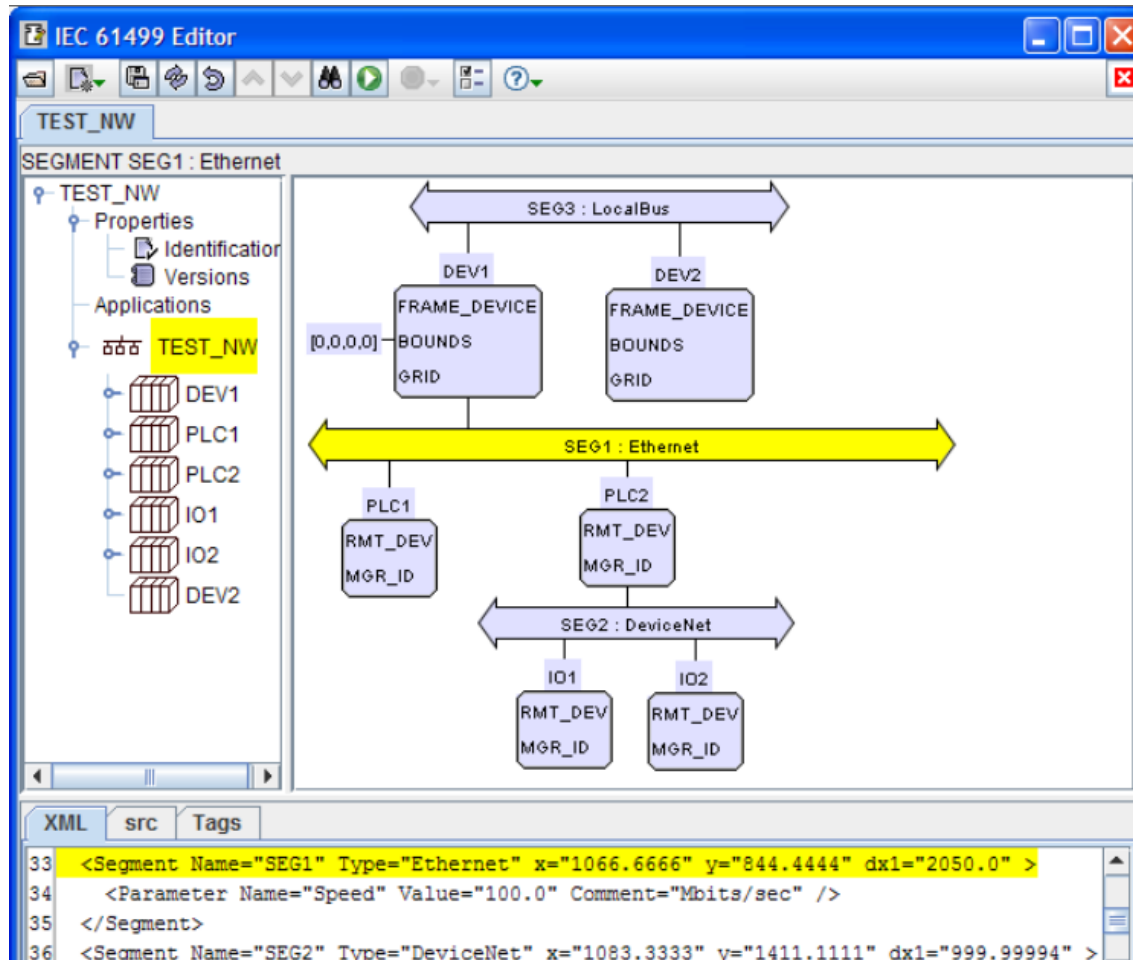


d) Bloque de función



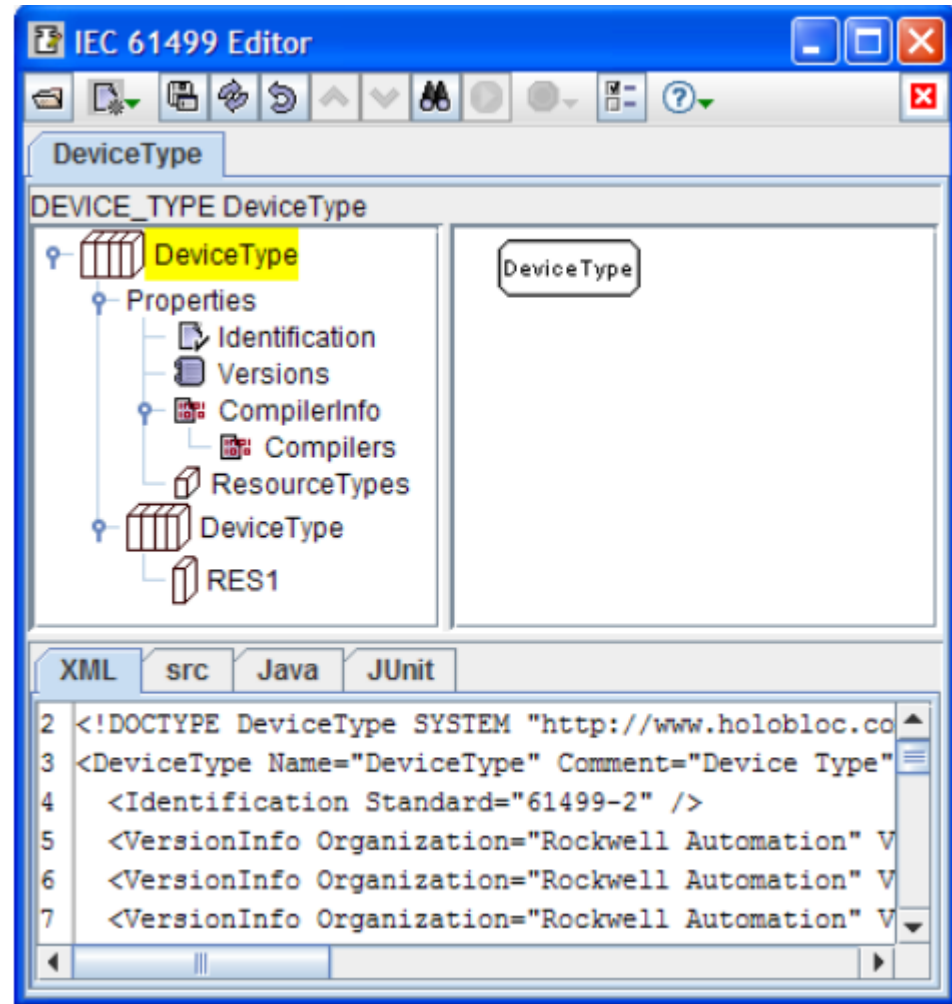
# System

- Formado por dispositivos, que pueden formar redes mediante segmentos



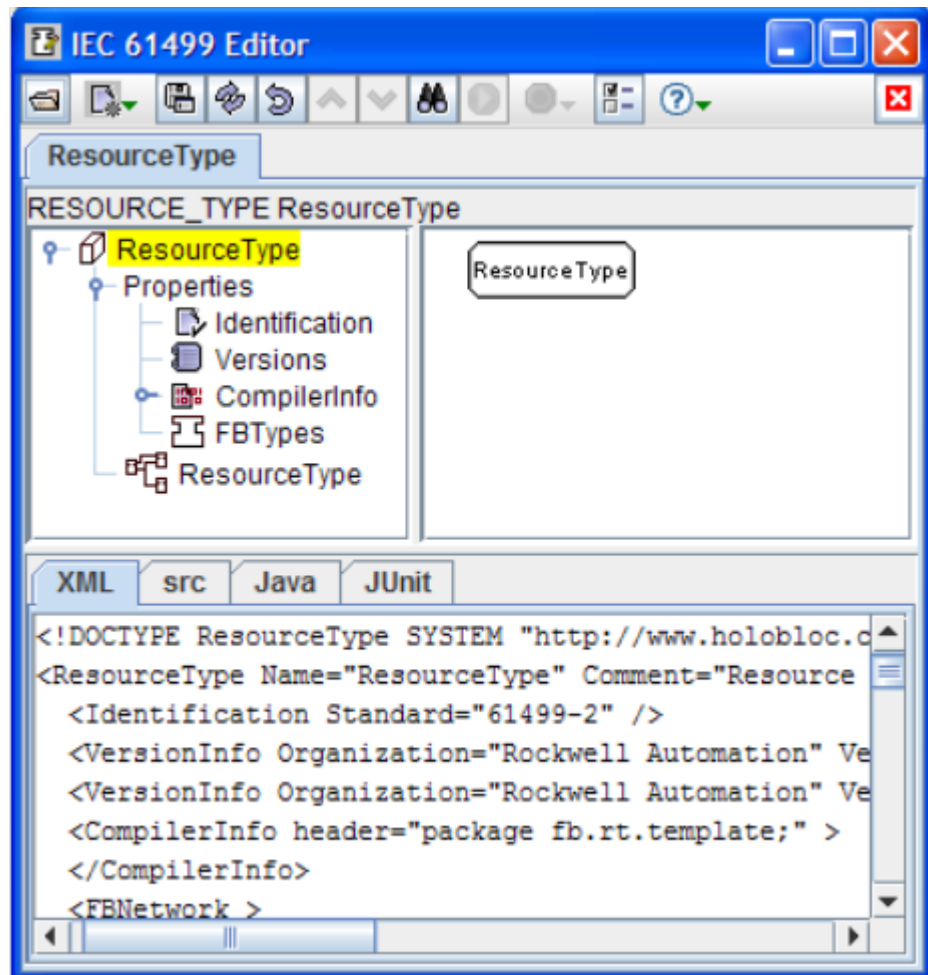
# Device

- Modelo de dispositivo



# Resource

- Recursos modulares dentro de cada dispositivo
- Son el contenedor de bloques funcionales



# Bloque Funcional (FB)

- Modelo básico en IEC 61499
- Componente software autocontenido que provee su función a través de una interfaz definida.
- Adoptado de IEC 61131-3 y extendido con una interfaz de eventos adicional
- Disparo de uno de los eventos de entrada inicia la ejecución del FB.
- Durante la ejecución del FB se procesan datos de entrada, se generan datos de salida, y/o se disparan eventos de salida

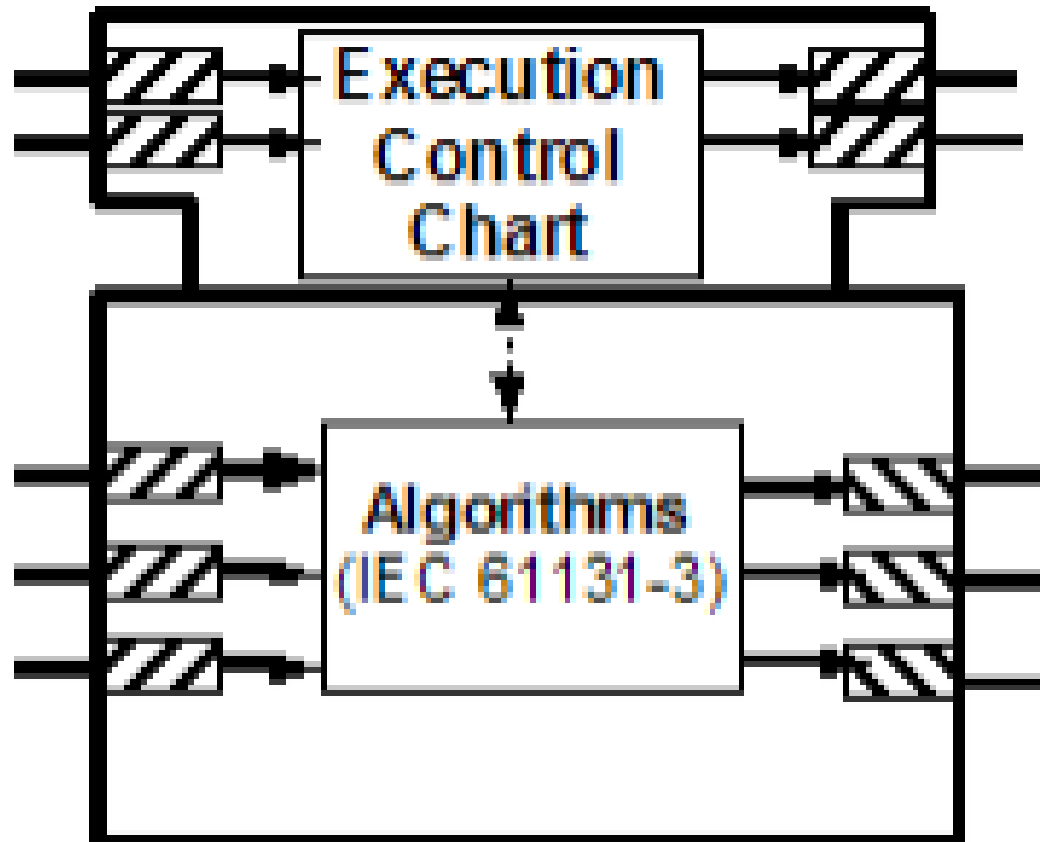
# Tipos de FB

- Básicos (BFBs)
- Composite (CFBs)
- Service Interface FBs (SIFBs)

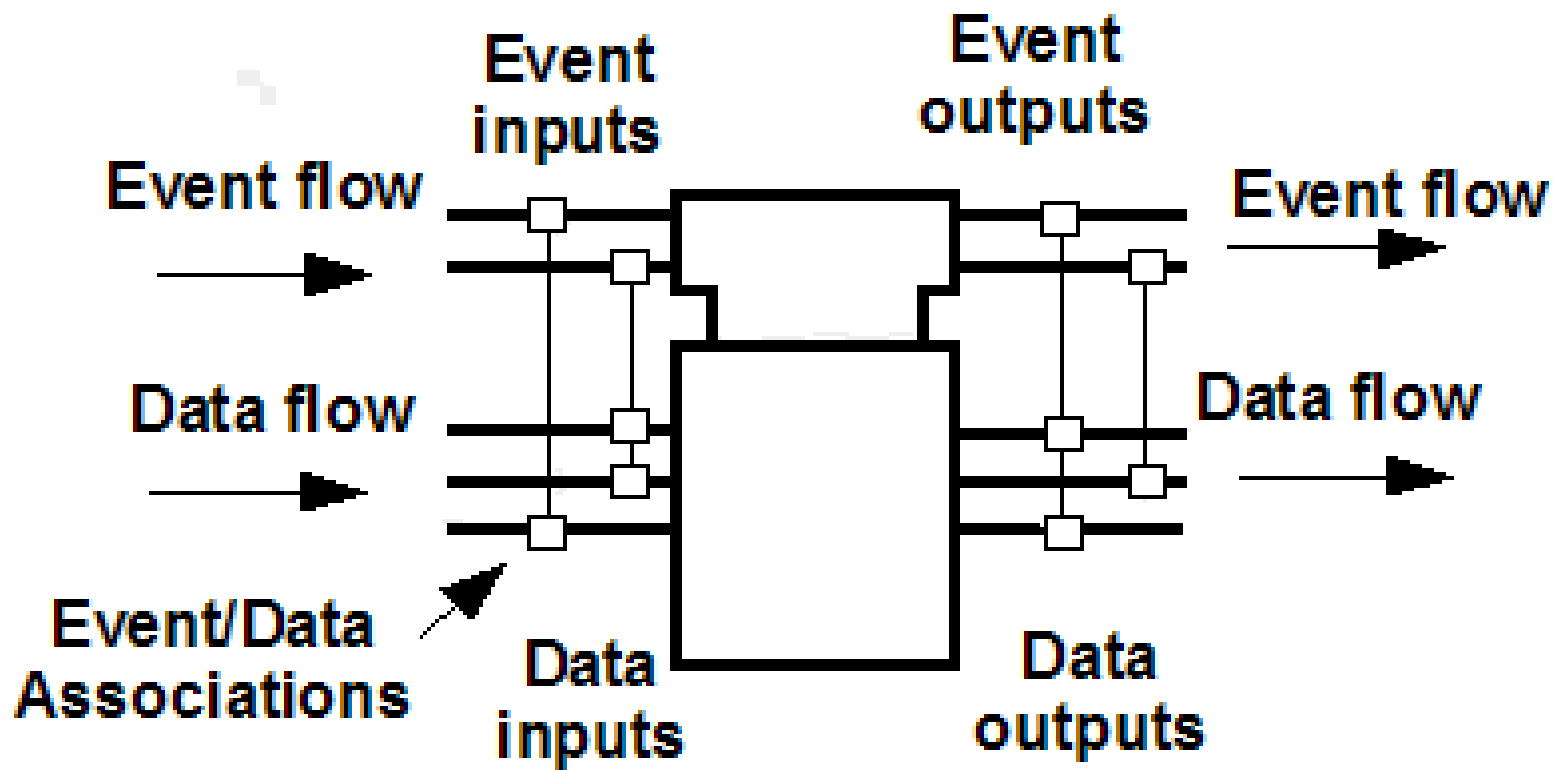
# BFBs

- Contienen una máquina de estados llamada Execution Control Chart (ECC) que controla la ejecución interna basada en la llegada de eventos de entrada
- ECC tiene tres partes:
  - Estados ECC
  - Acciones ECC asociadas a los estados
  - Transiciones ECC que conectan los estados
- Las acciones consisten en la ejecución de algoritmos y/o el disparo de eventos de salida

# BFBs

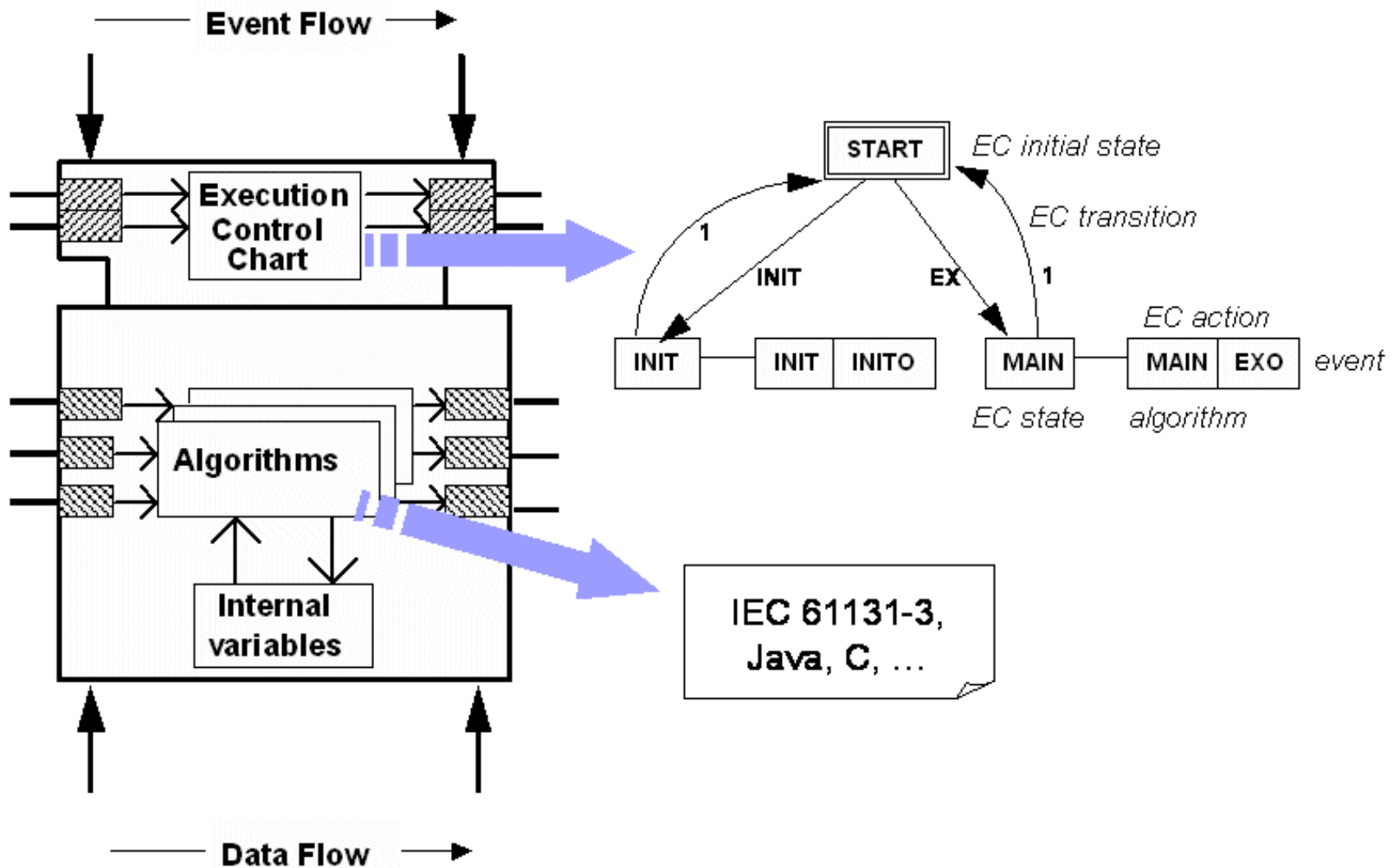


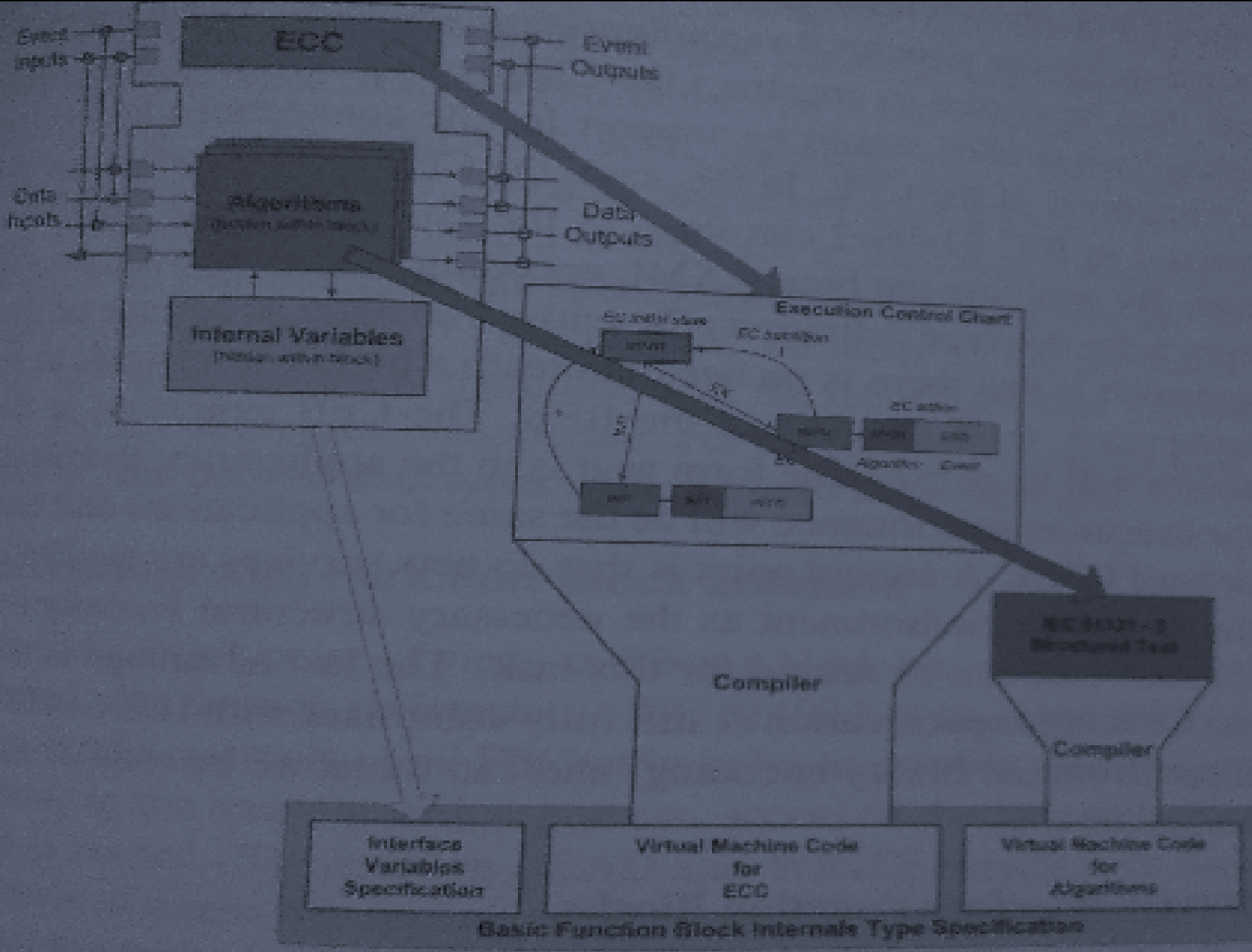
# BFBs





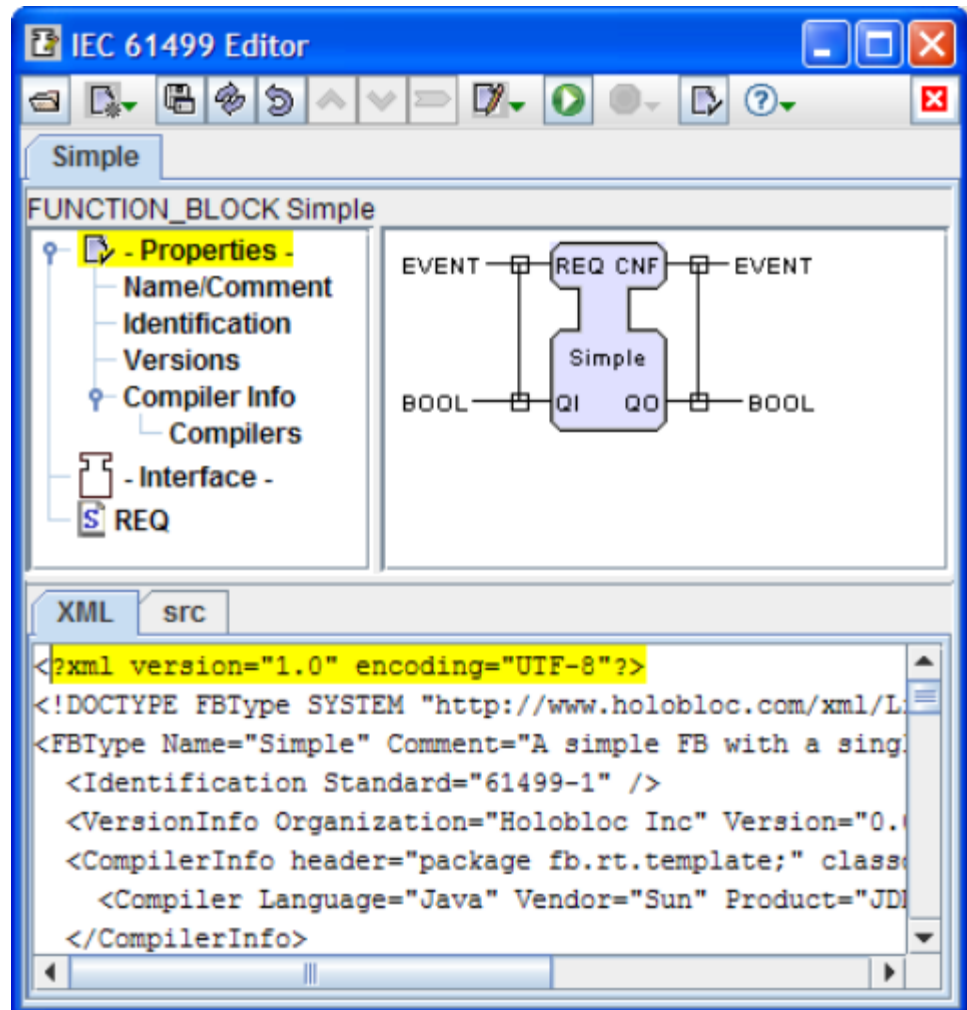
# Execution Control Chart (ECC)





# Simple FB

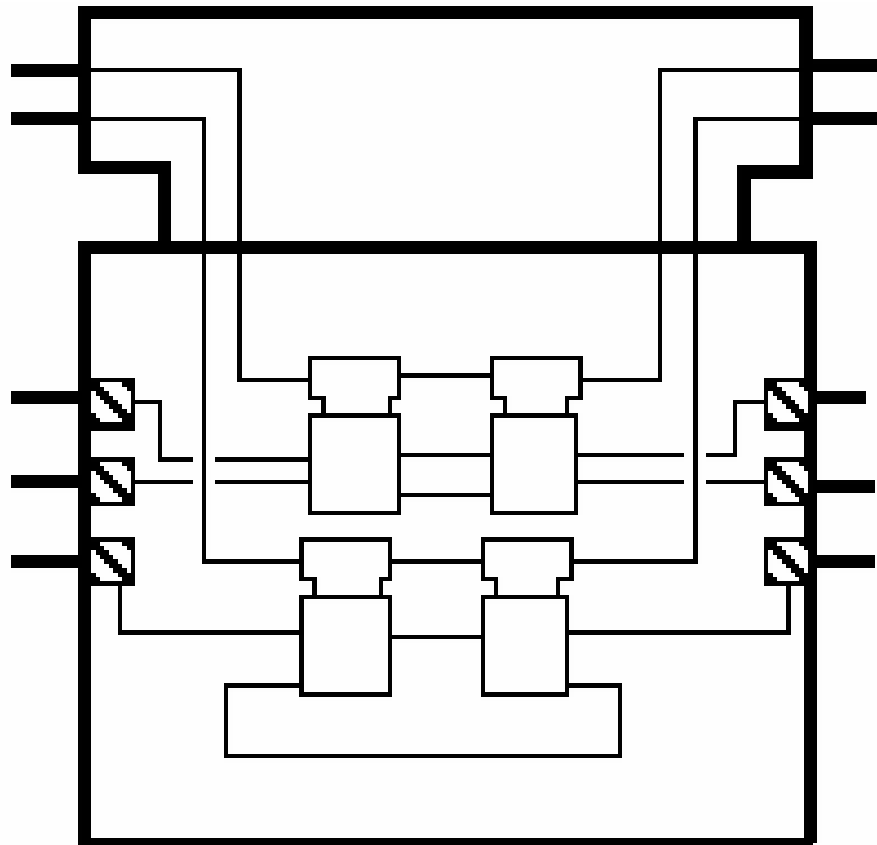
- Caso particular de BFB, sin algoritmos y con un solo evento y dato booleano de entrada y otros de salida



# Composite FBs (CFBs)

- Contenedor de FBs
- Puede contener un grupo de FBs y sus conexiones de eventos y datos
- Los eventos y datos entrantes son pasados a los FBs internos, y viceversa para las conexiones salientes

# CFBs



# CFBs en FBDK

The screenshot displays the IEC 61499 Editor interface. The main window shows a 'FUNCTION\_BLOCK Composite' diagram. The diagram features a central 'Composite' block with two internal blocks: 'INIT INITO' and 'REQ CNF'. The 'INIT INITO' block has two 'EVENT' inputs and two 'EVENT' outputs. The 'REQ CNF' block has two 'EVENT' inputs and two 'EVENT' outputs. The 'Composite' block has two 'BOOL' inputs labeled 'QI' and 'QO' and two 'BOOL' outputs labeled 'QI' and 'QO'. The interface includes a toolbar with various icons, a 'Composite' tab, and a 'FUNCTION\_BLOCK Composite' section. Below the diagram, there are tabs for 'XML', 'src', 'Java', and 'JUnit'. The 'XML' tab is active, showing the following XML code:

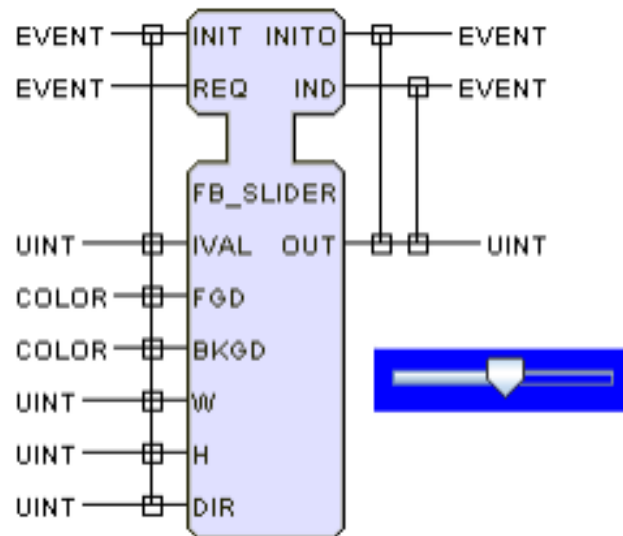
```
2 <!DOCTYPE FBType SYSTEM "http://www.holobloc.com/xml/Libra
3 <FBType Name="Composite" Comment="Composite Function Block
4   <Identification Standard="61499-2" />
5   <VersionInfo Organization="Rockwell Automation" Version=
6   <VersionInfo Organization="Rockwell Automation" Version=
7   <VersionInfo Organization="Rockwell Automation" Version=
8   <CompilerInfo header="package fb.rt.template;" >
9   </CompilerInfo>
```

# Service Interface FBs (SIFBs)

- Interfaz de los FBs a funcionalidades y a los servicios de bajo nivel provistos por el sistema operativo o por el hardware de los dispositivos
- Encapsulan la funcionalidad
- Ejemplos: Acceso al hardware, como la interface I/O o la interfaz de comunicación
- Librerías que proveen funciones necesarias para el sistema de control pueden accederse también a través de SIFBs
- Modelo de eventos y datos permite secuenciar las primitivas de servicio

# SIFBs, clasificación según funcionalidad

1. Graphical User Interface (GUI): IEC 61499 provee un SCADA básico a partir de elementos gráficos simples.
  - Ej.: sliders o barras de desplazamiento, botones, pilotos luminosos

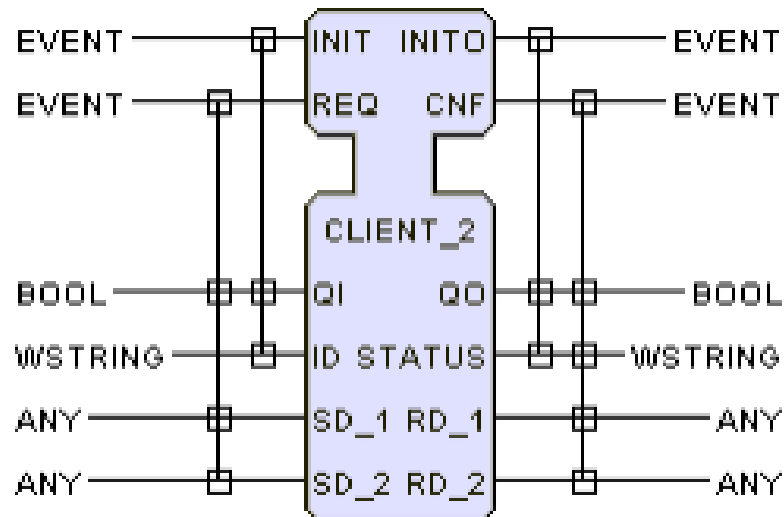




# SIFBs, clasificación según funcionalidad

2. Servicios de comunicación: comunicaciones cliente/servidor y editor/subscriptor soportadas por el sistema operativo, locales y remotas

Ej.: FB tipo cliente para dos variables que se comunica con servidor remoto



# Tipos de SIFBs

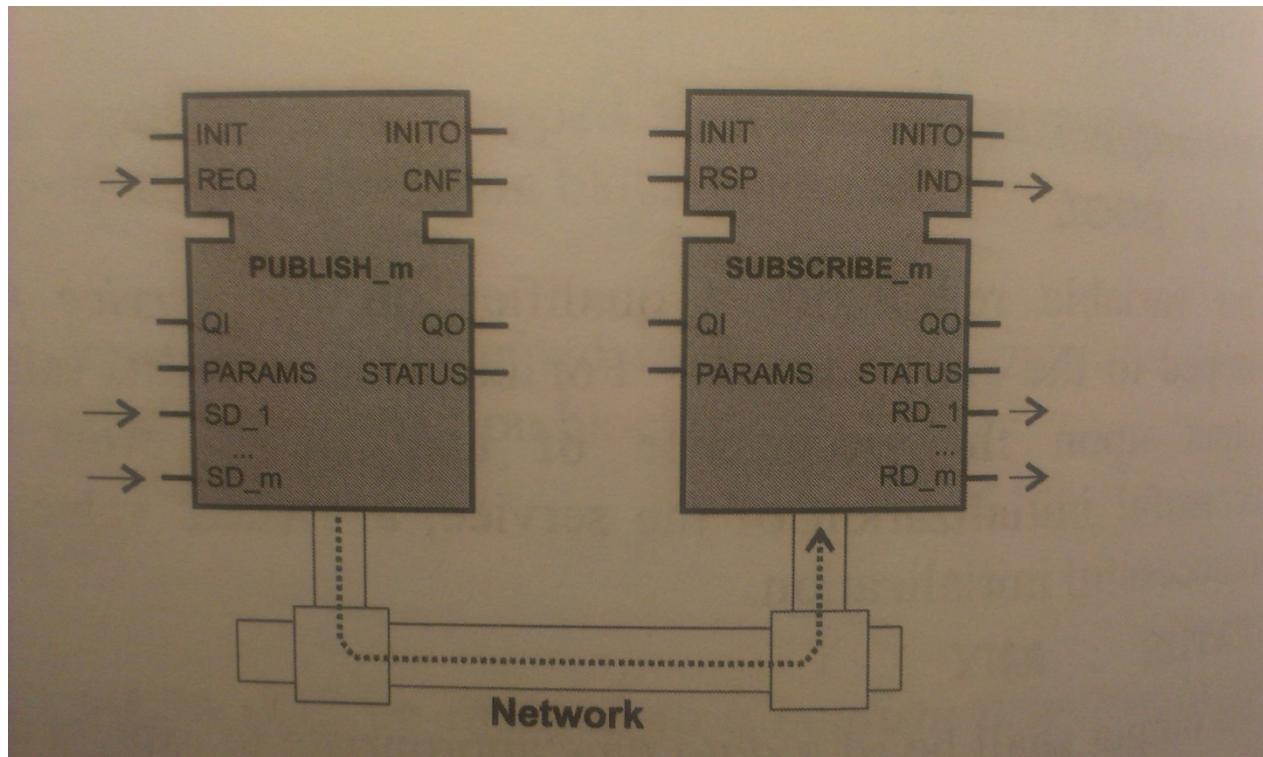
- Requester SIFBs: FB disparado por aplicación que permanece pasivo hasta que llega un evento a una de sus entradas de evento
  - Cuando llega el evento dispara el servicio que encapsula
- Responder SIFBs: FB disparado por hardware
  - Puede enviar eventos de salida causados por acciones en el recurso o el hardware (P.E. interrupciones) sin un disparo de evento de entrada previo

# Communication function blocks

- Son la base de los sistemas de control distribuido
- El estándar establece dos patrones genéricos de comunicación:
  - Publish/Subscribe para transacciones unidireccionales
  - Client/Server para comunicaciones bidireccionales
  - \* Definidos en el estándar, no responden a definiciones genéricas de dichos patrones

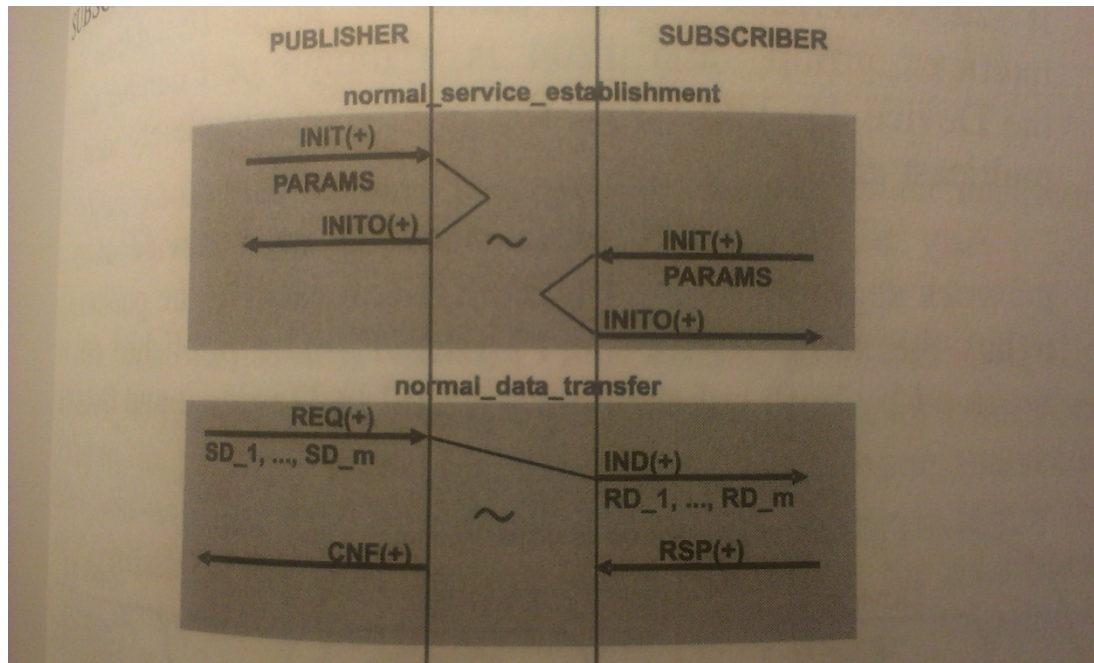
# Publish/Subscribe

- El Publisher publica los datos SD\_1 a SD\_m que provienen de uno o más bloques de la aplicación



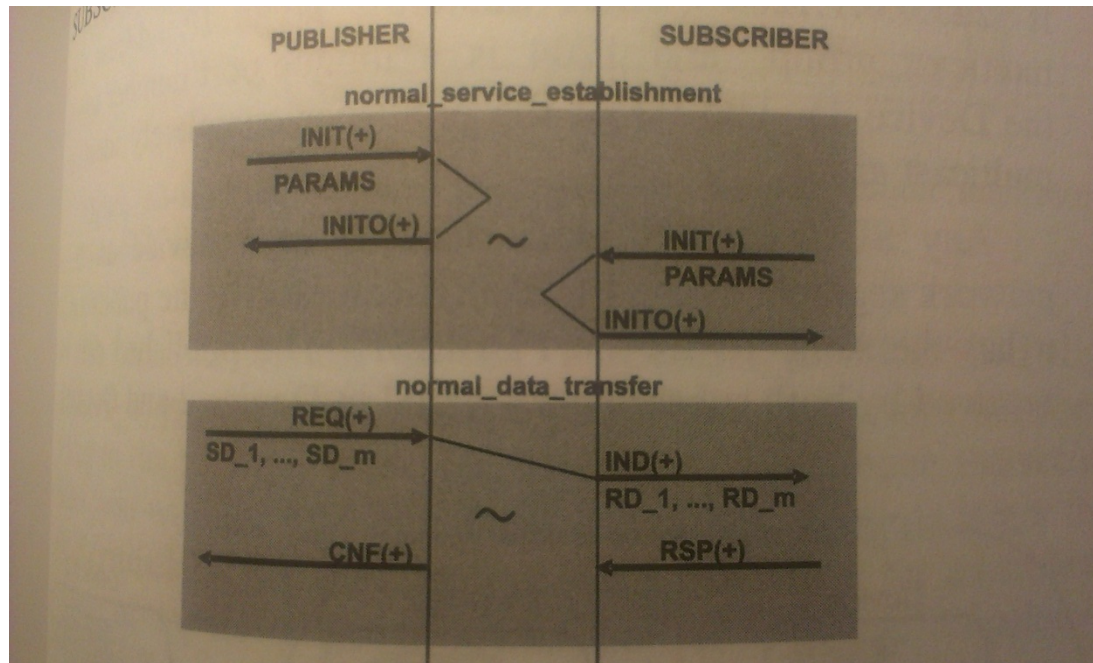
# Eventos de la comunicación con Publisher/Subscriber

- Bajo un evento request de la aplicación en la entrada de eventos REQ los datos son enviados por el Publisher a través de una red dependiente de implementación
- Una vez enviados, el Publisher informa a su aplicación con el evento de salida CNF
- El bloque funcional Subscriber es inicializado por su aplicación
- Diagrama de tiempo-secuencia:



# Eventos de la comunicación con Publisher/Subscriber

- La aplicación que envía inicia la transferencia normal de datos con un evento de entrada REQ al Publisher
- El Publisher envía los datos y dispara el evento IND del Subscriber para notificar a la aplicación del Subscriber que están disponibles nuevos valores de los datos en las salidas RD\_1 a RD\_m del Subscriber
- La aplicación del Subscriber notifica al Subscriber que los datos se han recibido por el evento RSP



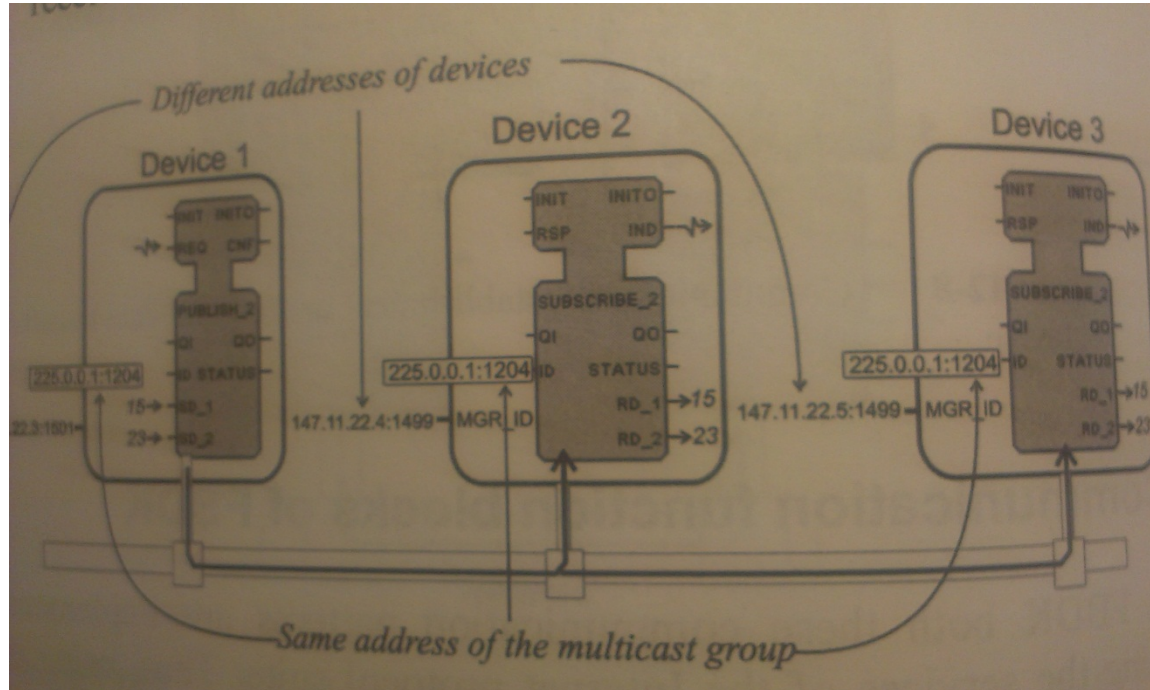
# Bloques funcionales de comunicación en FBDK

- Los dos patrones de comunicación están implementados usando los servicios del protocolo Internet
- Se utiliza el direccionamiento mediante IDs de parámetros de entrada



# Direccionamiento en Publish/Subscribe

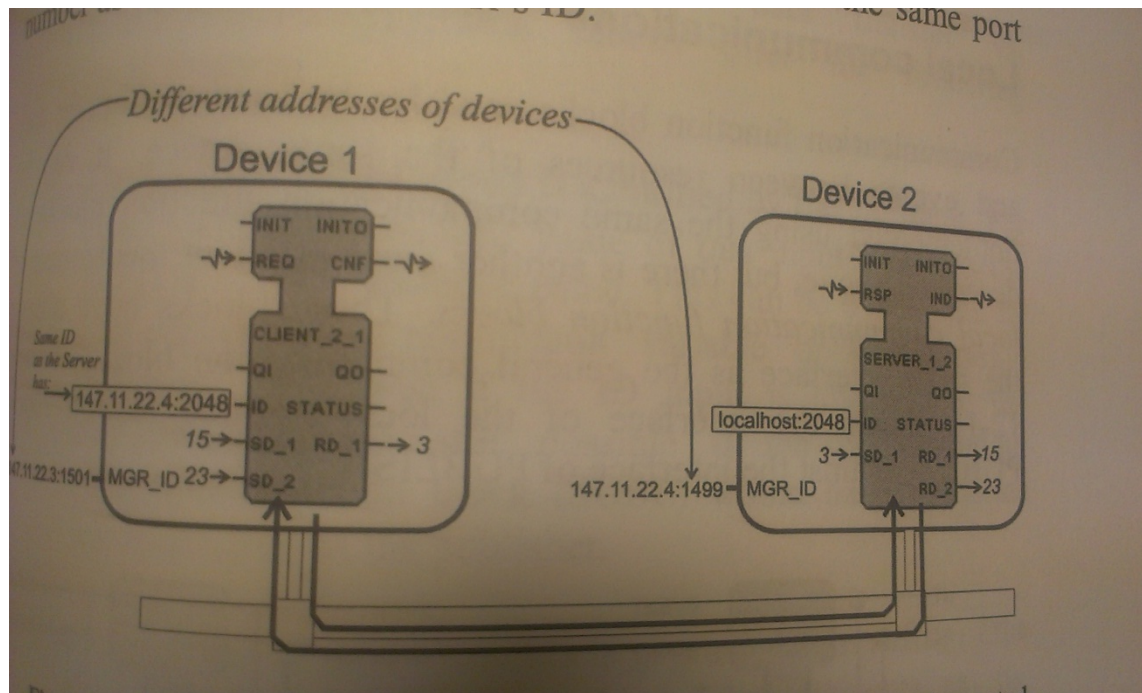
- Puede haber varios subscribers por cada Publisher
- Utiliza servicios Multicast UDP
  - Direcciones 224.0.0.255 a 239.255.255.255
- En el ejemplo, 225.0.0.1 es la dirección IP del grupo multicast, y 1204 es un número de puerto único seleccionado en el Device 1
- El ID del Publisher identifica al grupo multicast
- Cualquier bloque Subscribe en un dispositivo sobre el mismo segmento de red que el Publish puede recibir los datos si tiene el mismo ID que el Publish.





# Direccionamiento en Client/Server

- Comunicación punto a punto: Solo un cliente puede enviar y recibir datos de/a un servidor
- Direccionamiento del servidor referido al dispositivo que lo alberga (localhost), y reserva un puerto
- ID del cliente se forma con IP del dispositivo servidor y el puerto que aparece en la ID del servidor

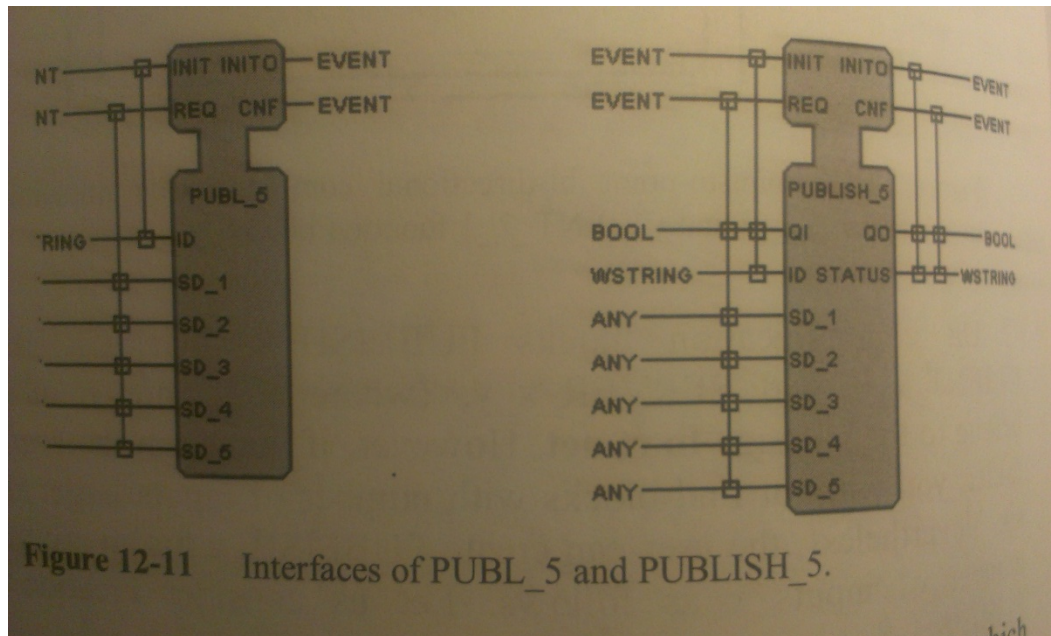


# Cantidad de datos

- PUBLISH\_x, SUBSCRIBE\_x, CLIENT\_x\_y y SERVER\_y\_x tienen cantidades x e y de datos
- Se encuentran en fb.rt.net hasta cantidades de seis
- Se pueden crear bloques funcionales con cantidades de datos mayores, por ejemplo para un PUBLISH\_8:
  - Open PUBLISH\_6
  - Add data inputs SD\_7, SD\_8
  - Rename a PUBLISH\_8
  - Save as XML

# Comunicaciones locales

- Los bloques funcionales de comunicaciones se pueden usar para transferir datos y eventos entre recursos del mismo dispositivo
- Se pueden usar los bloques funcionales de comunicaciones genéricos o un mecanismo más eficiente que son los local communication function blocks, con una interfaz similar
- Se usan para comunicaciones dentro del mismo proceso en ejecución



# Comunicaciones locales

- Velocidad es mayor
- El grupo multicast se identifica por el ID de entrada, al que se asigna el nombre del bloque Publ
  - Mismo nombre del bloque Publ y Subl, sin necesidad de configurar IDs
- Puede no darse la correspondencia entre el número de datos del Publ\_x y los Subl\_y
- Puede no darse la correspondencia de tipo de datos: no habrá copia de valores pero no se producirán errores

# SIFBs, clasificación según funcionalidad

3. Interfaces al hardware: Sensores de temperatura, controladores de velocidad de motores control de válvulas, controlador de intensidad de luz...

# FB Networks (FBN)

- Las conexiones de eventos y de datos conectan los FBs en Redes de FBs (FBN) que permiten el modelado de la funcionalidad de control en el modelo de aplicación
- Las Application se modelan sin una infraestructura de dispositivos o control previa en mente
- Es el modelo de System el que especifica el equipo de control junto con la red de comunicaciones usada para el intercambio de datos entre los controladores distribuidos

# Mapeado (mapping) en el modelo de sistema

- El mapeado establece qué parte de la aplicación se localiza en cada dispositivo de control
- Equivale al concepto de asignación en DCSs
- La aplicación 1 se mapea en los dispositivos 3, 4 y 5, mientras que la aplicación 2 se mapea solo en el dispositivo 2.

# Dispositivos (Devices)

- IEC 61499 modela el equipamiento de control capaz de ejecutar aplicaciones IEC 61499 como dispositivos
- Un dispositivo consta de:
  - Una interfaz de comunicaciones: Provee servicios de comunicación para el dispositivo y para las partes de la aplicación que residen en el dispositivo
  - Una interfaz de proceso: Provee los servicios de acceso a los sensores y actuadores necesarios para controlar el proceso (p.e, leer la posición actual de un motor)
  - Un componente de gestión de dispositivo
  - Puede contener o no recursos



# Recursos (Resources)

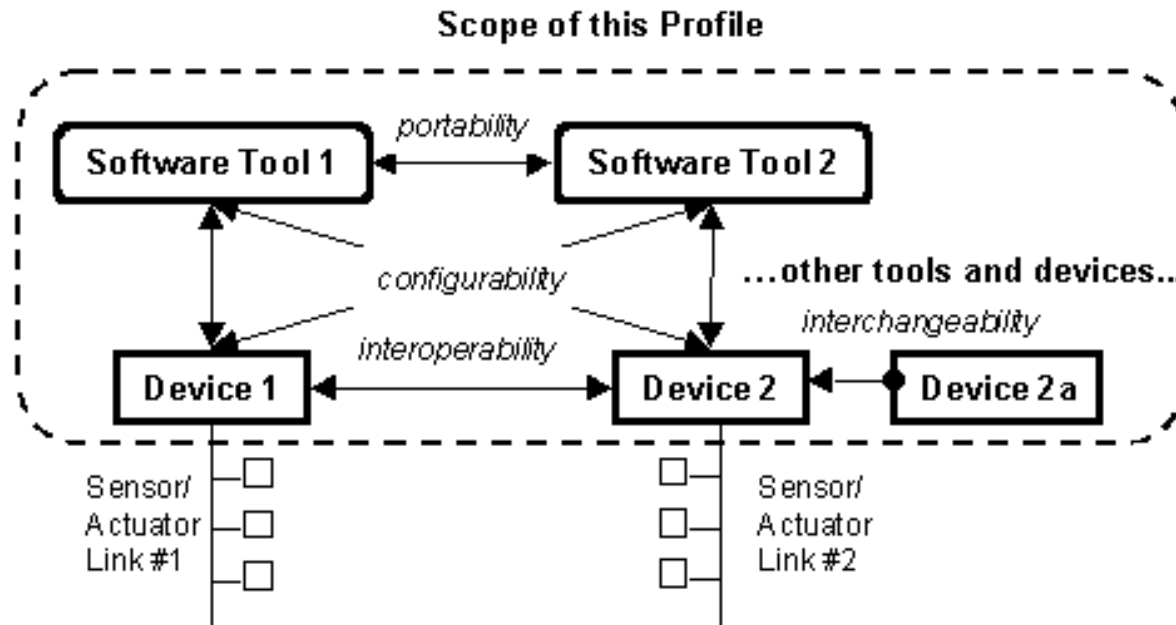
- Un recurso es una unidad funcional que contiene aplicaciones o partes de aplicaciones que residen en un dispositivo específico y que dispone de control independiente de su operación
- Dentro de un dispositivo, los recursos puede ser creados, borrados, configurados, etc, sin interferir con otros recursos y las aplicaciones de las que forman parte.

# Ejecución de los recursos

- El recurso tiene que proveer un entorno de ejecución para la aplicación
  - El recurso debe enviar notificaciones de eventos a los FBs y debe permitir también a los FBs procesar los eventos de entrada correspondientes a su estructura interna
- El recurso accede a la interfaz de comunicaciones y a la interfaz de proceso a través del dispositivo.
- Los SIFBs son el medio para proveer a la aplicación con el acceso a las interfaces.

# IEC 61499 Compliance Profile for Feasibility Demonstrations

- Documento anexo al estándar
- Incorpora las características a implementar para demostrar
  - **interoperability** of *devices* from multiple suppliers;
  - **portability** of software between software tools of multiple suppliers;
  - **configurability** of devices from multiple vendors by software tools of multiple suppliers; and
  - **interchangeability** of devices and resources from multiple vendors.



# Contexto

- IEC [61131-3](#) - Programmable controller systems, Part 3 - Programming Languages
- IEC [61499-1](#) - Function blocks, Part 1 - Architecture
- IEC [61499-2](#) - Function Blocks, Part 2 - Software tool requirements
- IEC [61499-4](#) - Draft - Function Blocks, Part 4 - Rules for compliance profiles
- ISO/IEC [8802-3](#), Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

# Clases Java y CSIFB

- Correspondencia entre los SIFB del interfaz de comunicación y las clases Java que implementan su funcionalidad

<u>Generic CSIFB type</u>	<u>Encapsulated class</u>
PUBLISH	<a href="#"><u>java.net.DatagramSocket</u></a>
SUBSCRIBE	<a href="#"><u>java.net.MulticastSocket</u></a>
CLIENT	<a href="#"><u>java.net.Socket</u></a>
SERVER	<a href="#"><u>java.net.Socket</u></a>

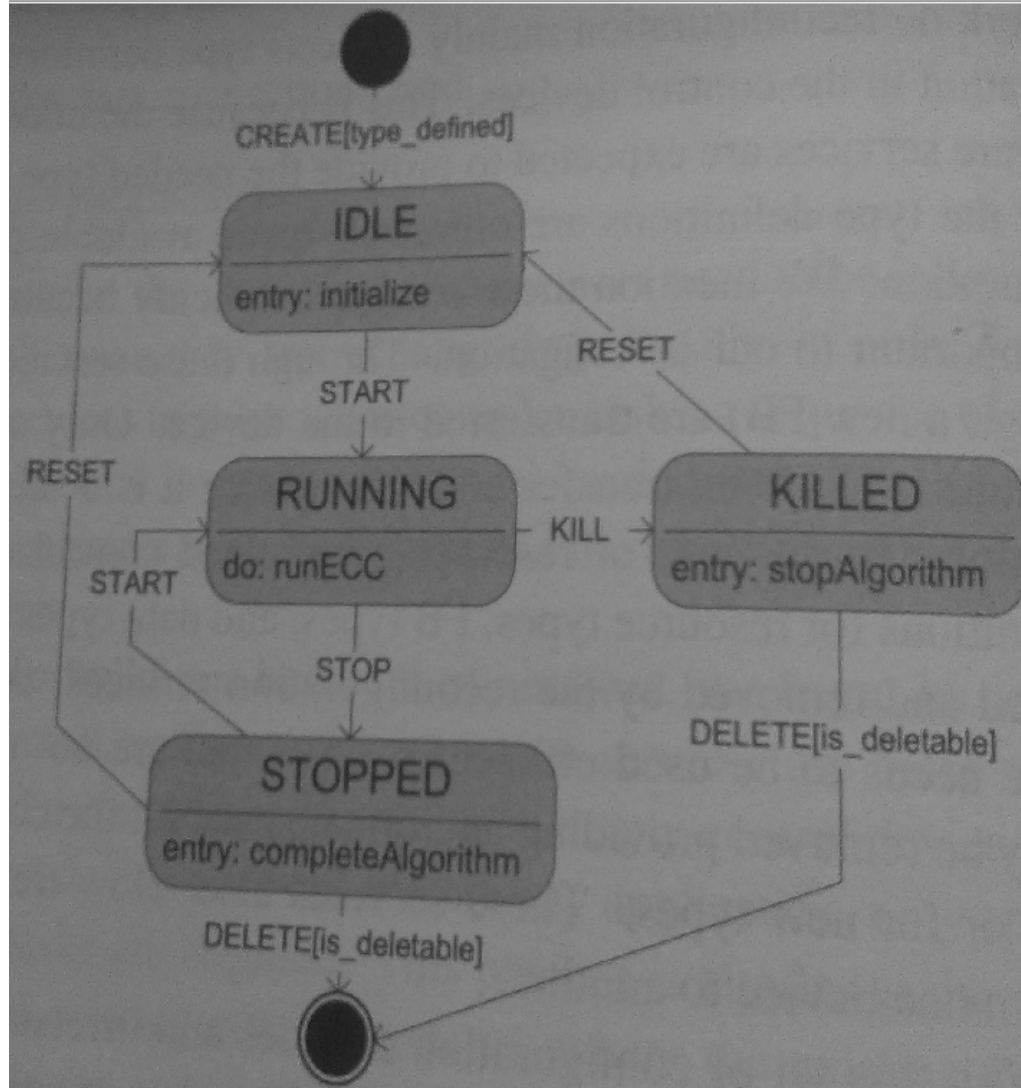
# Gestión de Dispositivo (Device Management)

- La funcionalidad principal de la Gestión de Dispositivo es gestionar las aplicaciones y recursos dentro del dispositivo.
- La funcionalidad de Gestión del Dispositivo incluye también proveer una interfaz externa para las herramientas de ingeniería que permite el download y upload de las aplicaciones hacia (o desde) el dispositivo

# Servicios de la Gestión de Dispositivo

- Query, Create, Initialize, Start, Stop, Kill y Delete
- Los provee la Gestión de Dispositivo
  - Desde el nivel de Dispositivo para los Recursos
  - Desde el nivel de Recursos para los FBs y sus interconexiones

# Máquina de estados operacional de un FB



- Comandos de transición de estado del FB



# Servicios de Gestión / Reconfiguración

- Servicios estructurales
- Servicios de librería
- Servicios de control de ejecución
- Servicios de interacción de estado
- Servicios de consulta

# Servicios estructurales

- Modifican la estructura y el comportamiento de la aplicación de control
- Los básicos son:
  - CREATE: Recursos dentro de un dispositivo, FBs dentro de un recurso, conexiones (eventos y datos) dentro de un FB
  - DELETE: Recursos, FBs y conexiones
  - WRITE: Valores de parámetros a entradas de datos de dispositivos, de recursos y de FBs

# Servicios de librería

- Modifica la librería de definiciones de tipos disponibles en el dispositivo
- Las definiciones de tipo modificables son tipos recurso, tipos FB y tipos de dato derivado IEC 61131-3
- Los básicos son:
  - CREATE Tipo en la librería de tipos del dispositivo
  - DELETE Tipo en la librería de tipos del dispositivo

# Servicios de Control de Ejecución

- Modifican el estado de ejecución de los FBs
- Especifican cuándo puede el FB procesar eventos de entrada y cuándo no puede
- Los básicos son:
  - START: Pone al FB en el estado de ejecución. Los eventos de entrada se procesan
  - STOP: Detiene el procesamiento de eventos de entrada
  - KILL: Aborta el procesamiento de eventos de entrada desde ese instante en adelante
  - RESET: Pone al FB en el estado inicial

# Servicios de Interacción de Estado

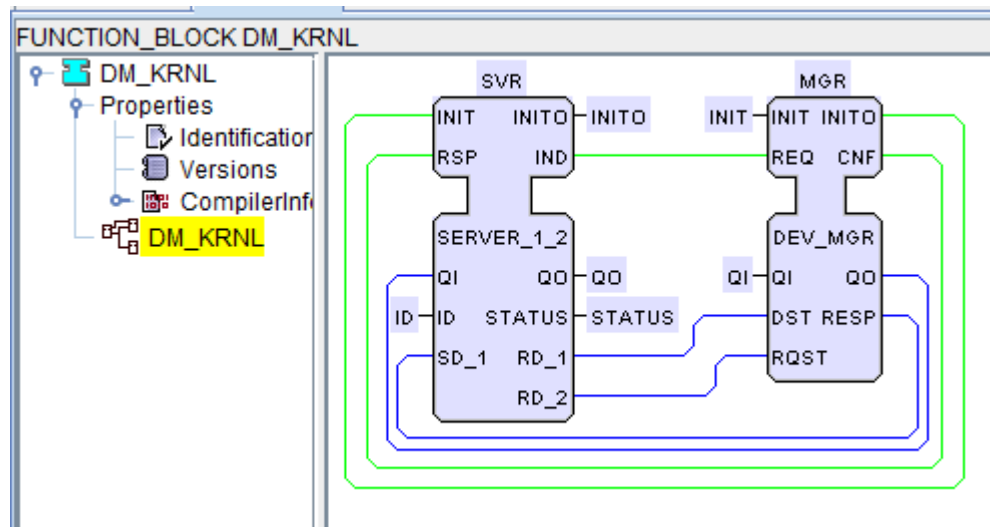
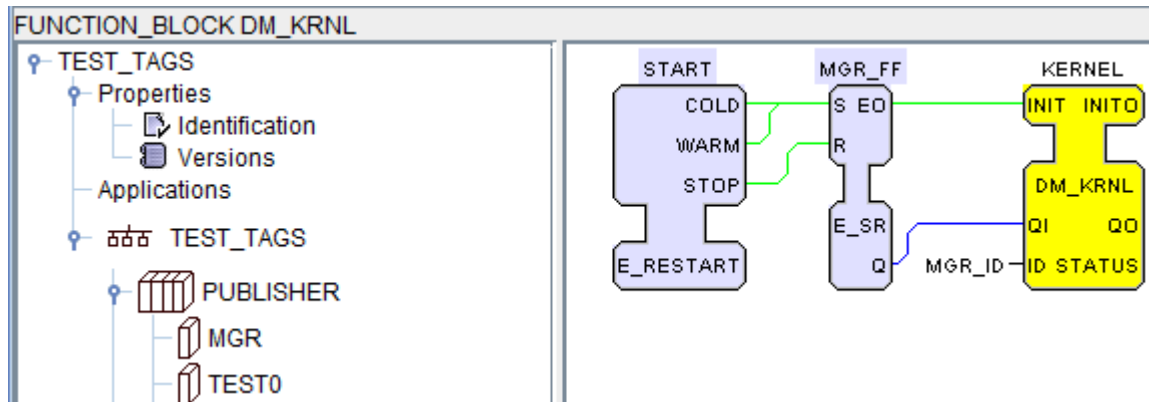
- Proveen acceso al estado específico de aplicación del FB
- Este estado se almacena en variables de entrada, salida e internas
- Los básicos son:
  - WRITE: Escribe en datos de entrada, salida y variables internas del FB
  - READ: Lee en datos de entrada, salida y variables internas del FB

# Servicios de consulta (Query)

- Recogen el estado actual del sistema de control
- Comprobación de consistencia equivalente a un upload: Instantánea del sistema
- Los básicos son:
  - QUERY Resources: Devuelve información de los recursos instanciados en un dispositivo de control
  - QUERY FBs
  - QUERY FB State
  - QUERY Connections
  - QUERY Type, Genérico

# Gestor de Dispositivo

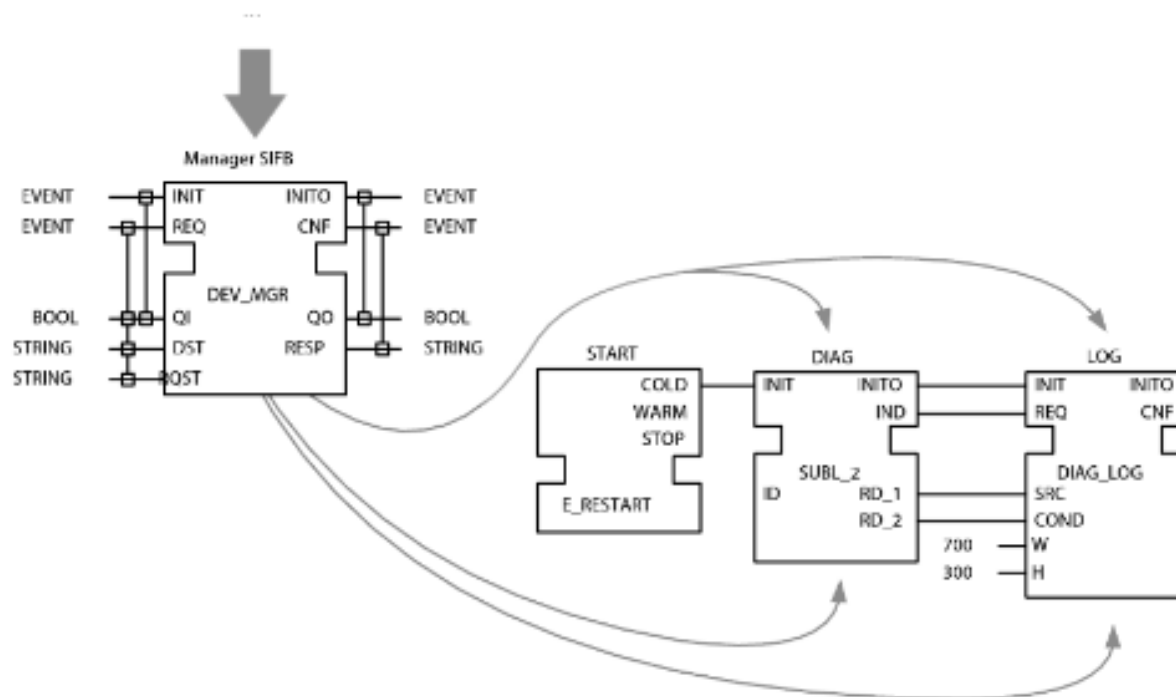
- Contexto del DEV\_MGR: FB Composite DM\_KRNL



# Gestor de dispositivo

- Basado en el FB DEV\_MGR
- Protocolo XML para el despliegue del contenido del recurso (FBs y conexiones)
  - Secuencia XML que puede llegar desde otro dispositivo

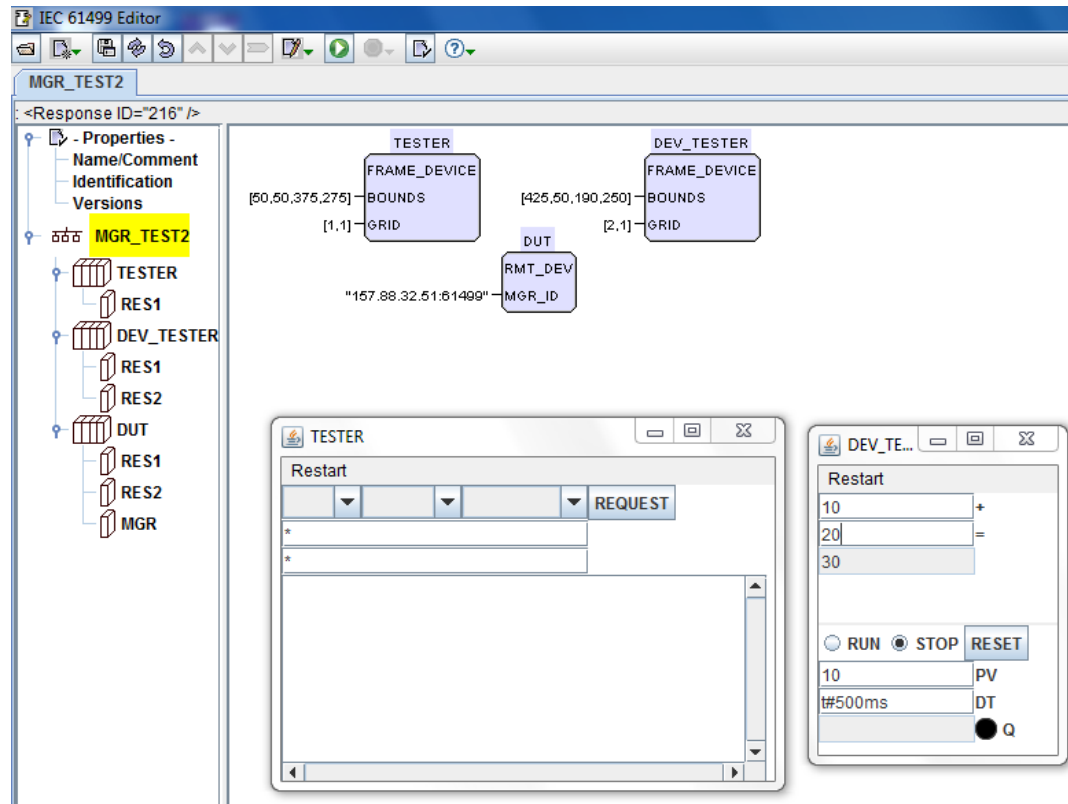
```
<Request ID="3" Action="CREATE" >  
<FB Name="DIAG" Type="SUBL_2" />  
</Request>  
  
<Request ID="4" Action="CREATE" >  
<FB Name="LOG" Type="DIAG_LOG" />  
</Request>  
  
<Request ID="7" Action="CREATE" >  
<Connection Source="DIAG.IND" Destination="LOG.REQ" />  
</Request>
```





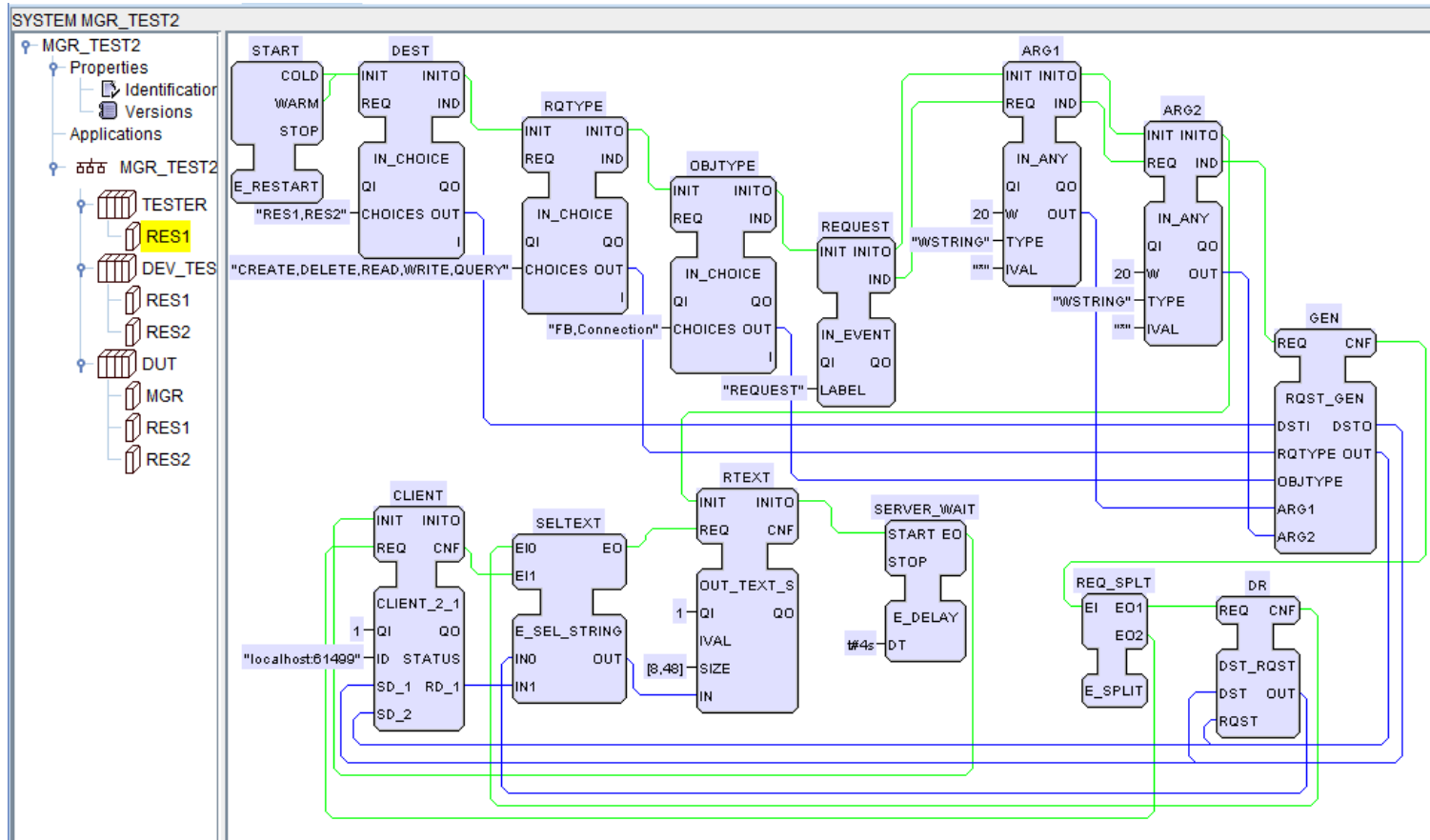
# Gestor de Dispositivo

- Aplicaciones pueden hacer uso de los servicios en línea de forma explícita



# Gestor de Dispositivo

- Cliente 2\_1 explícito:



- Cliente 2\_1 implícito en el entorno de desarrollo interviene en el Deployment inicial fuera de línea

# Interfaz de reconfiguración externa

- Acceso a los servicios de reconfiguración
- Utilizan FBNs, locales o distribuidas
- Especificación en IEC 61499 Compliance Profile for Feasibility Demonstration
- Para distinguir dispositivos que cumplen el perfil de conformidad se define un tipo de dispositivo llamado RMT\_DEV

# RMT\_DEV

- Contiene un tipo especial de recurso llamado RMT\_RES
- FBN definida para este recurso gestiona la interfaz de reconfiguración externa, encapsulada en un CFB llamado DM\_KRNL
- Dentro del DM\_KRNL, un SFIB de tipo Server de comunicación se encarga de comunicarse con los elementos externos
- El SFIB está conectado a un MFB del tipo DEV\_MGR y comunica su respuesta

# DEV\_MGR

- Es un MFB
- Se encarga de interpretar (parsing) las cadenas XML

# Peticiones al MFB

- Encapsuladas en lenguaje XML
- Ejemplo: creación de un FB llamado TEST de tipo TEST\_TYPE. Código en la petición:

```
<Request ID="3" Action="CREATE">  
  <FB Name="Test" Type="TEST_TYPE" />  
</Request>
```
- La única extensión necesaria es un mapeado de los servicios de reconfiguración a la estructura XML de los comandos de gestión

# Interfaz de reconfiguración interna

- DEV\_MGR también sirve como interfaz de reconfiguración interna, con la diferencia de que los datos no vengan de la red sino de otros FB internos de la RCA.
- DEV\_MGR interpreta las cadenas XML provistas en su interfaz y pasa los datos resultantes al entorno de ejecución en un formato que el entorno pueda manejar

# Interfaz de reconfiguración interna

