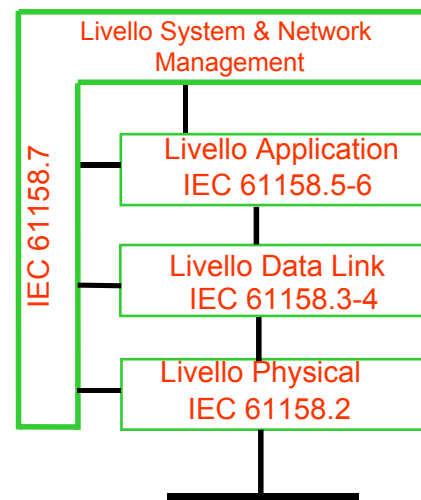


# Bus di Campo IEC 61158

- Il Fieldbus IEC 61158 , sviluppato a partire dal 1986, rappresenta il modello più completo di Fieldbus dal punto di vista funzionale.
- Anche se molti aspetti sono stati stravolti in fase di votazione ufficiale, costituisce tuttora un valido modello di riferimento.



Comprende tre livelli:

- Physical layer
- Data Link layer
- Application layer

Il communication stack è gestito da un **Network Management Agent**.

## Physical Layer

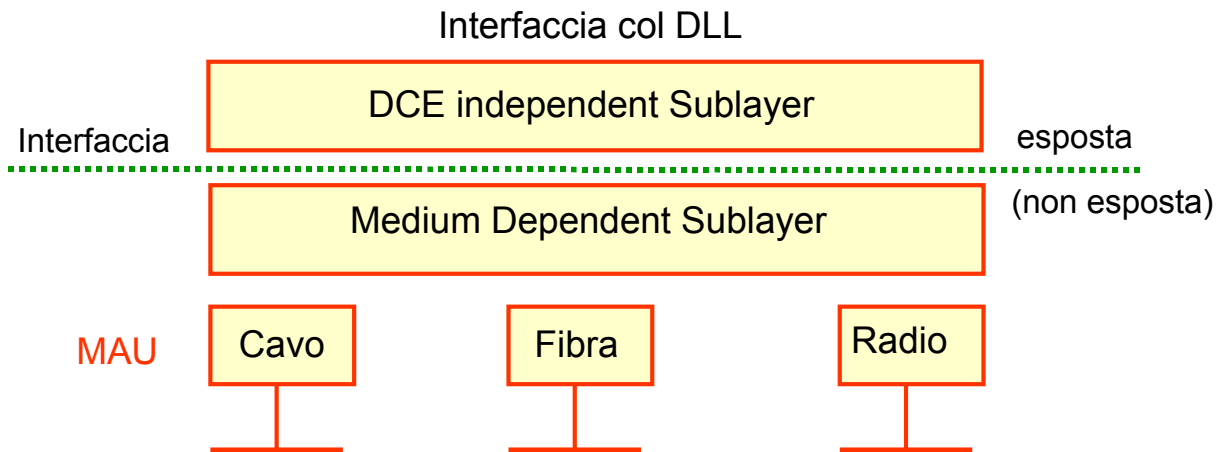
- Provvede all'adattamento dei dati, alle caratteristiche del mezzo fisico prescelto.
- Il Physical layer riceve le unità dati dal Data link Layer, **aggiunge il preambolo** ed i **delimitatori**, **effettua la codifica** e trasmette i segnali così ottenuti, sul mezzo fisico.
- Sicurezza Intrinseca.
- Distribuzione di alimentazione sul mezzo fisico.
- Topologia a stella, a bassa velocità per cablaggi pre-esistenti.
- Topologie multi-drop ad alta velocità per nuovi cablaggi.

Tre differenti velocità:

- 31,25 Kbps
- 1Mbps
- 2,5Mbps

**Mezzi utilizzati: Cavo, fibra ottica e radio.**

# Modello del Physical Layer



Il Physical layer è organizzato in tre sublayers:

- un **DCE independent sublayer** le cui caratteristiche non dipendono dal particolare mezzo fisico impiegato.
- Un **Medium Dependent Sublayer** che effettua gli adattamenti richiesti dallo specifico mezzo fisico adottato.
- Un **MAU** relativo allo specifico mezzo fisico adottato.

Differenti mezzi fisici possono essere mescolati per soddisfare le esigenze di particolari topologie.

## Servizi all'interfaccia PHL-DLL

La granularità dei Phs-User data, scambiati all'interfaccia è 1 ottetto.

### Servizio di notifica delle caratteristiche del PhL:

- Ph-characteristics indication (minimum data rate, framing overhead)

### Servizio di trasmissione e Ricezione:

- Ph-data request (class, data)
- Ph-data indication (class, data)
- Ph-data confirm (status)

dove class può essere:

- start\_of\_activity
- data
- end\_of\_data/end\_of\_activity

data è un ottetto

## Servizi all'interfaccia Station Management-PhL

- Questi servizi sono richiesti per l'inizializzazione e la selezione delle varie opzioni adottate nel PhL.
- **Ph-RESET** request  
Serve a resettare tutte le funzioni del Physical layer.
- **Ph-SETVALUE** request (parameter name, new value)
- **Ph-SETVALUE** confirm (status)  
Serve a modificare i parametri del PhL
- **Ph-GETVALUE** request (parameter name)
- **Ph-GETVALUE** confirm (current value)  
Consente allo Station Management di leggere i parametri del PhL
- **Ph-EVENT** indication (parameter name)  
notifica allo SM una modifica non richiesta di qualche parametro.

## Wire Medium

- **Digital data transmission**
- **Self Clocking**
- **Half Duplex** communication (bidirectional but in only one direction at a time)
- **Manchester coding**

Sono disponibili due modalità di accoppiamento (coupling) e tre velocità di trasmissione:

- **Voltage mode** (parallel coupling) 31.25 Kbps
- **Voltage mode** (parallel coupling) 1 Mbps
- **Current mode** (serial coupling) 1 Mbps
- **Voltage mode** (parallel coupling) 2.5 Mbps

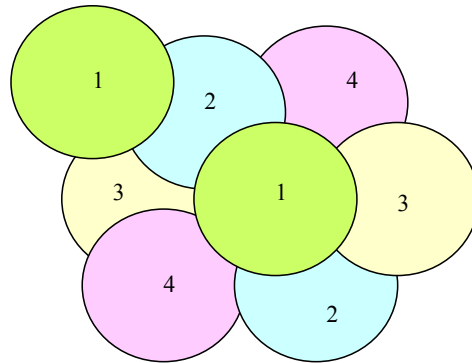
Sono inoltre possibili le seguenti opzioni:

- **Power / No Power on the bus**
- **Intrinsically safe / not intrinsically safe**

Un nuovo MAU a 1 Mbit/s, 1 Ampere è stato sviluppato per dispositivi che richiedono potenze elevate.

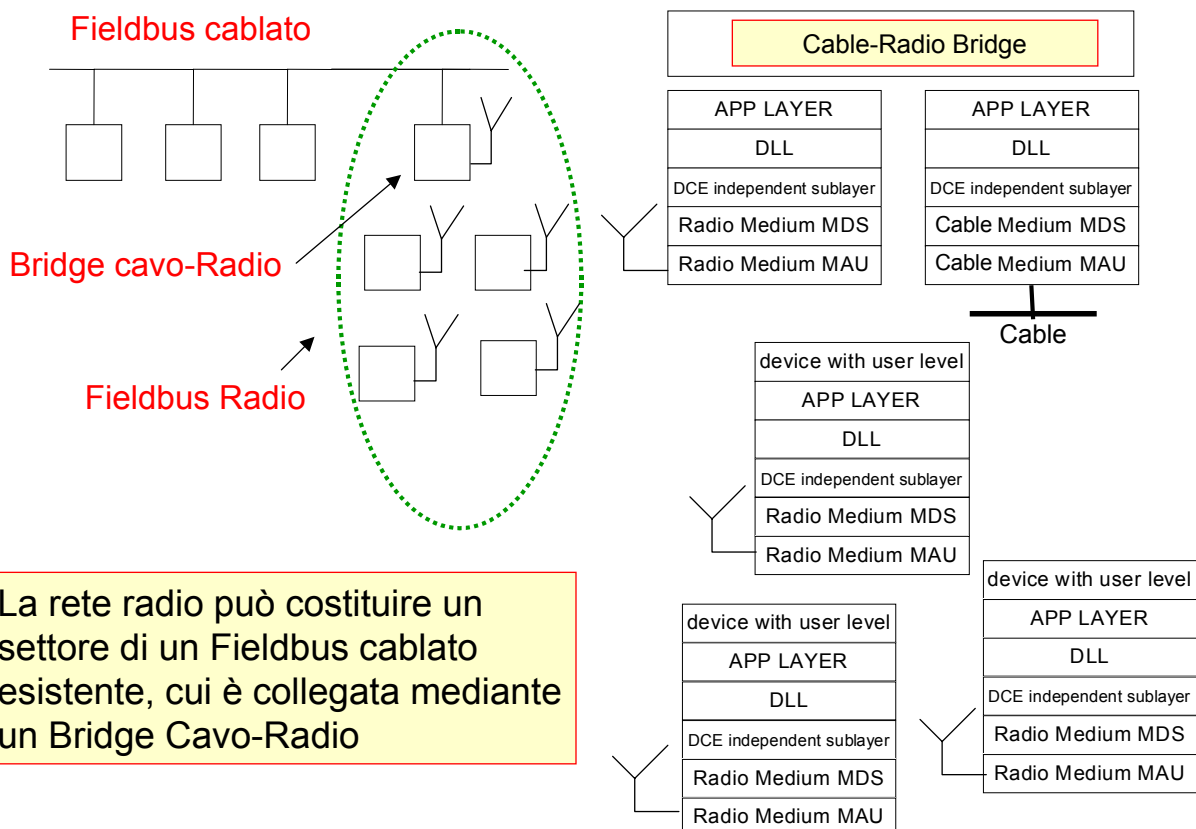
# Radio

- L'uso della radio richiede la soluzione di diversi problemi che sono assenti nei cavi:
- Shared Medium
- Potential interference, malicious and accidental
- Propagation, power budgets
- Security, privacy
- Technological advance

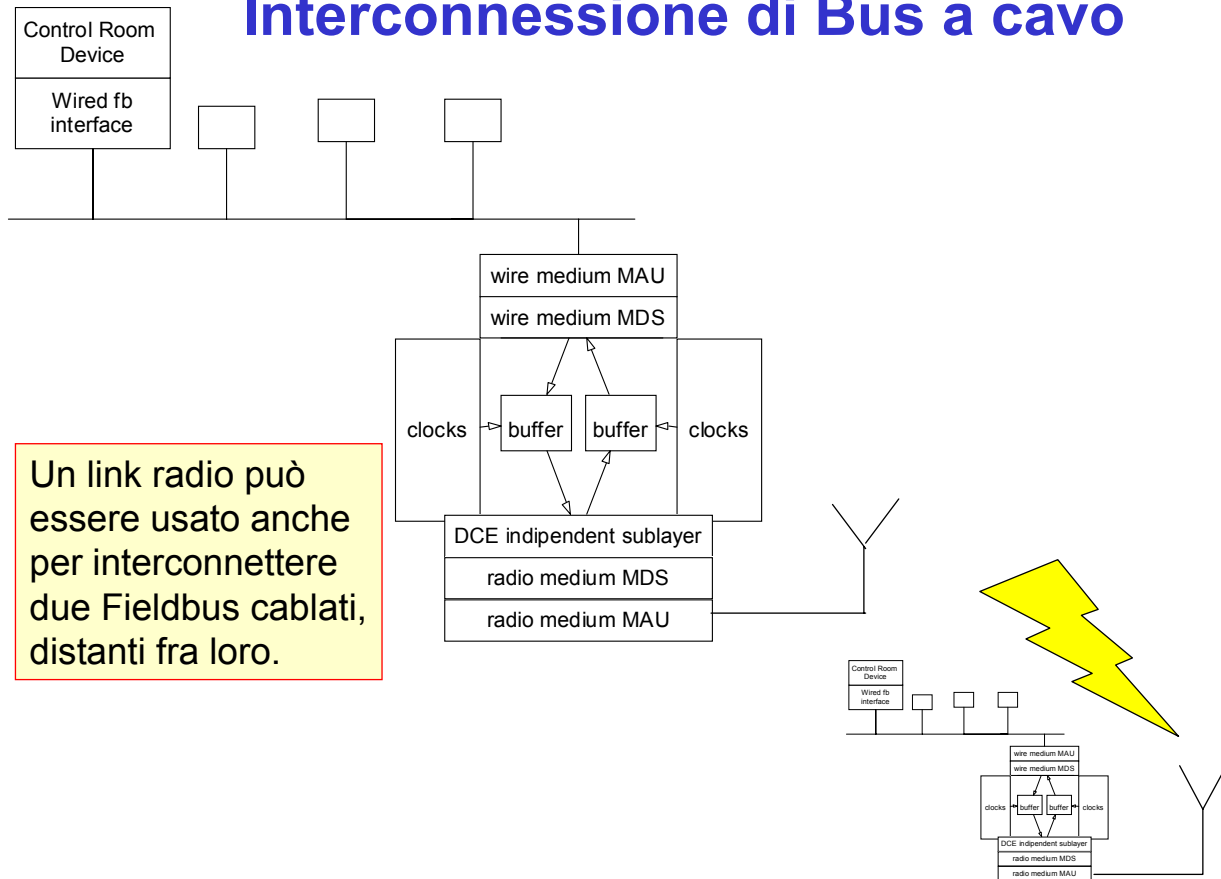


- Viene adottata la topologia cellulare ed il riuso delle frequenze.
- La trasmissione è in Narrow band a 12.5 KHz nelle bande VHF and UHF.
- Trasmissione è a 4800 bps nelle bande libere per telemetria (de-licensed telemetry bands)
- E' anche possibile disporre di 1 Mbit/s and 4 Mbit/s utilizzando lo standard IEEE 802.11

## Topologie Radio



## Interconnessione di Bus a cavo



## La PDU del Radio Physical Protocol

La struttura della PDU è più complessa rispetto a quella prevista per altri mezzi fisici più affidabili

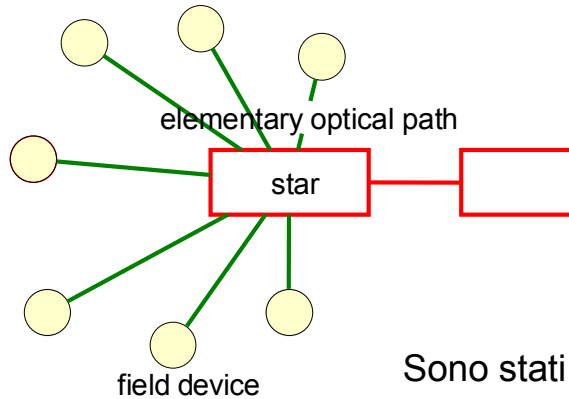
<b>PREAMBLE</b> 40 symbol times plus 100 us	<b>SFD</b> 1 octect	<b>Length</b> 2 octects	<b>Length</b> 2 octects	<b>PhSDU Sequence</b> up to 300 octects	<b>CRC</b> 2 or 4 octets
---	------------------------	----------------------------	----------------------------	--	-----------------------------

- Il preambolo è usato per sincronizzare il ricevitore col codice in arrivo. E' costituito da una sequenza di 1010....10
- Il Frame delimiter contiene un pattern 00110011 che lo differenzia dal preambolo
- i campi "Length" contengono la lunghezza del Ph-PSU e sono ripetuti due volte
- il CRC copre i campi Length e PhSDU

# Fiber Optics Medium

La fibra ottica può essere utilizzata in reti con topologia a stella. I dispositivi sono connessi alla “optical star” tramite percorsi ottici elementari.

Anche le optical active star sono connesse tramite percorsi ottici elementari.



Sono previsti quattro differenti MAU:

31.25 Kbit/s, single fiber mode MAU

31.25 Kbit/s double fiber mode

MAU

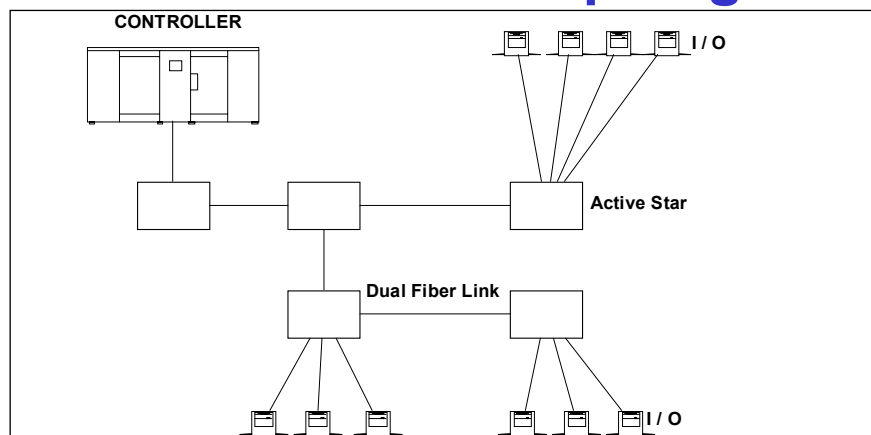
1 Mbit/sec MAU

2.5 Mbit/s MAU

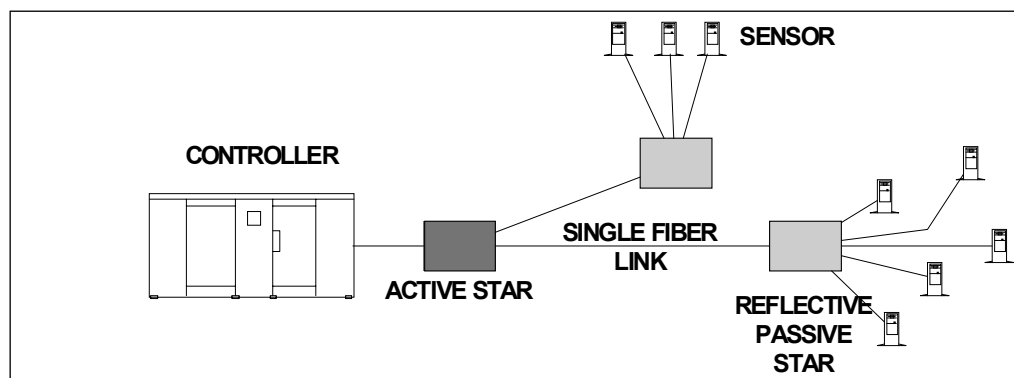
Sono stati considerati tre diversi tipi di link :

- Link fra optical active stars.
- Link fra optical active stars e field device
- link tra optical active stars e optical passive stars

## Possibili Topologie in fibra



Esempio di topologia a stella a 1,0 Mbit/sec



Esempio di topologia a stella a 31,25 Kbit/sec

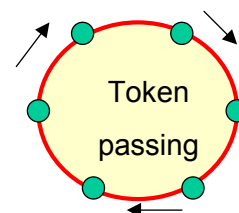
# Generalità sul Data Link Layer

- Rappresenta il livello più critico per quanto riguarda la definizione dei servizi e delle procedure. Suo compito è la gestione real-time delle esigenze trasmissive dei processi afferenti al Bus di Campo.
- I processi sono spesso caratterizzati da stringenti, vincoli temporali, tra loro differenti. In generale è possibile distinguere due tipologie di processi tempo-critici: processi periodici ed asincroni.
- **Processi periodici:**
  - caratterizzati dalla frequenza fissa di produzione delle informazioni.
  - I parametri tipici sono sempre noti a priori.
  - L'esigenza di comunicazione tipica è la garanzia che ciascuna informazione prodotta venga trasmessa entro una "deadline".
- **Processi asincroni:**
  - Sono caratterizzati dall'aleatorietà con cui le informazioni sono prodotte.
  - I parametri tipici di generazione risultano difficilmente prevedibili.
  - L'esigenza di comunicazione tipica è che l'informazione prodotta venga consegnata in tempi quanto più bassi possibile.

## Strategie di gestione del Bus

- **Problema:** due tipologie di processi richiedono differenti strategie di comunicazione per soddisfare le esigenze di comunicazione del traffico generato.
- **Soluzione:** gestione centralizzata dell'accesso, basata su pre-schedulazione e gestione distribuita, tramite uso di token.
- **La gestione centralizzata** è in grado di soddisfare tutte le esigenze trasmissive dei processi periodici. Tale gestione si basa sull'uso di una tabella di schedulazione.

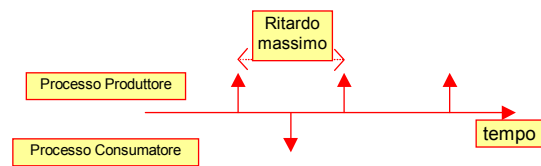

Tabella di schedulazione



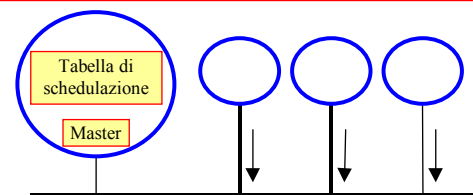
- **La gestione** dell'accesso al mezzo di comunicazione **di tipo distribuito** tramite l'uso del token, permette di soddisfare le esigenze trasmissive dei processi asincroni.

# Gestione centralizzata

- Basata sull'uso di una tabella di schedulazione preparata sulla base delle esigenze temporali dei processi.
- Nella tabella vengono determinati a priori tutti gli istanti di trasmissione di ciascun processo periodico.
- La tabella è relativa ad un intervallo temporale limitato detto macrociclo, allo scadere del quale tutte le informazioni in essa contenute vengono ripetute nel tempo.
- La tabella di schedulazione (che è solitamente statica) può essere aggiornata quando vengono aggiunti nuovi processi.

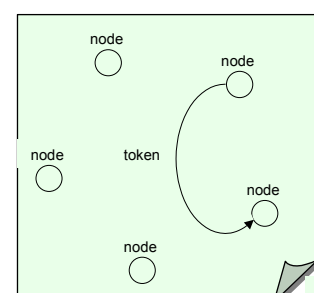
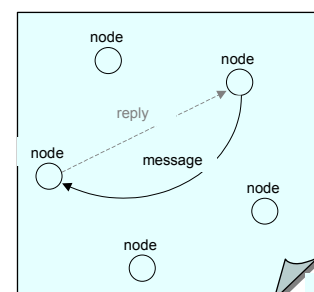
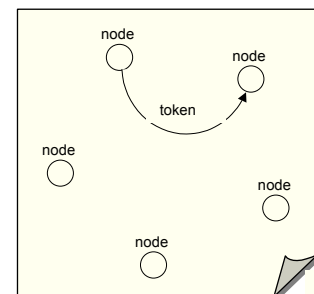


La gestione centralizzata è basata sull'uso di un Link Active Scheduler (LAS) che spedisce le varie autorizzazioni utilizzando la tabella di schedulazione.



# Gestione distribuita

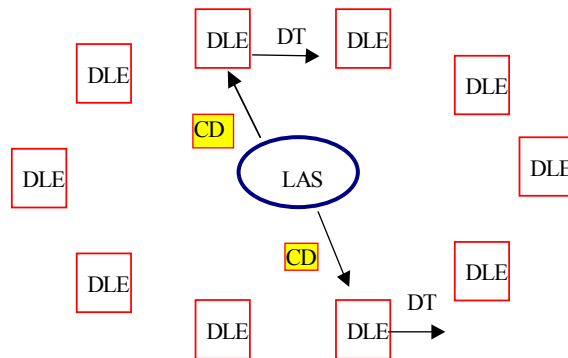
- Basata sull'uso di **token**.
- Permette di soddisfare efficientemente le **esigenze trasmissive dei processi asincroni**.
- Per tali processi l'uso della **pre-schedulazione** comporterebbe uno **spreco di larghezza di banda**.
- L'uso del meccanismo del token, consente la trasmissione solo quando essa è effettivamente richiesta
- Non è soddisfacente per i processi periodici con stringenti vincoli temporali.





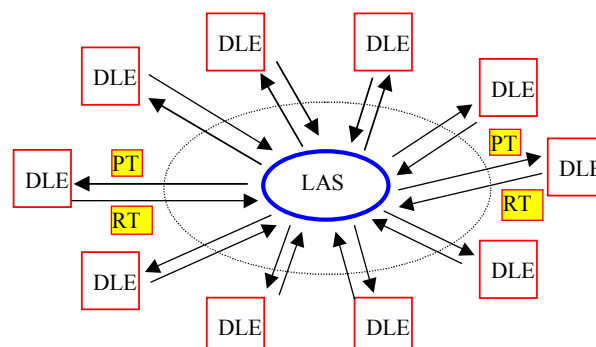
# Politiche del LAS 1

- Il LAS utilizza meccanismi per la gestione del traffico ciclico e di quello asincrono, che conciliano gli approcci centralizzato e distribuito del FIP e ProfiBus.
- Può autorizzare un DLE a trasmettere un singolo dato, tramite l'invio di una frame chiamata **Compel Data** (CD).
- Può autorizzare un DLE ad una sequenza di trasmissioni (tutte ovviamente pre-schedulate) tramite l'invio, al DLE stesso, di una frame chiamata **Execute Sequence** (ES), nella quale viene specificata la sequenza trasmissiva pre-schedulata che può essere eseguita e il tempodisponibile.



# Politiche del LAS 2

- Gli intervalli temporali non occupati da trasmissioni pre-schedulate vengono utilizzati dal LAS per realizzare il meccanismo di distribuzione del token.
- Ciò viene effettuato tramite l'uso di una particolare frame detta Pass Token (PT), inviata a ciascun DLE incluso in una lista di circolazione del token.
- In ciascun PT viene incluso una durata temporale, e la priorità.
- Il DLE che riceve il PT usa la larghezza di banda per trasmettere informazioni caratterizzate da priorità non più bassa di quella specificata nel PT, fino allo scadere del tempo associato ad esso.



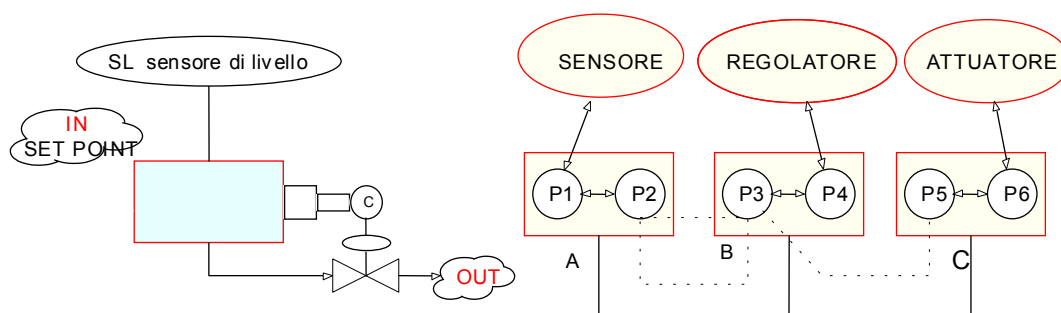
## Politiche del LAS 3

- Il token viene restituito al LAS tramite una frame **Return Token** (RT).
- Quando il LAS riceverà tale frame, potrà cedere il token al successivo DLE presente nella lista di circolazione del token, se non vi è alcuna trasmissione pre-schedulata.
- Il passaggio del token è **fisicamente effettuato dal LAS**, ma **virtualmente è realizzato dal DLE**.
- Il messaggio RT fisicamente è realizzato settando un bit nell'ultima frame trasmessa dal DLE (per il LAS ciò corrisponde alla restituzione del token).
- L'uso del token è inframmezzato alla trasmissione di traffico ciclico, poiché utilizza gli spazi liberi nella tabella di schedulazione.

## APPLICATION LAYER

Consente alle Applicazioni di accedere ai servizi della rete.

Una Applicazione è costituita da più processi applicativi cooperanti. La cooperazione locale è supportata dal Sistema Operativo, quella fra processi remoti dal sistema di comunicazione.



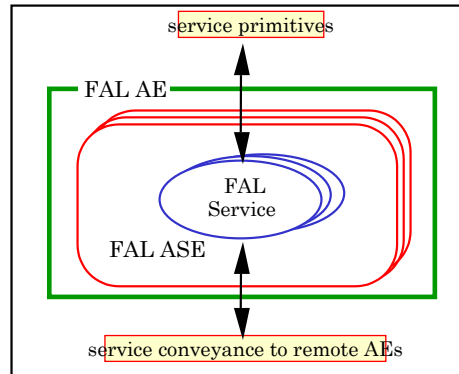
Un Application Process AP è la rappresentazione astratta di quegli elementi del sistema reale che, attraverso l'elaborazione di informazione, realizzano una applicazione.

Le AP possono interagire con modalità **Client/Server** e modalità **Publisher/Subscriber**.

Una AP è rappresentata in termini di :

- **Oggetti**: elementi astratti che rappresentano risorse fisiche, eventi, ecc.
- **Application entities**: forniscono capacità di comunicazione.

Una **Application Entity** fornisce servizi che sono raggruppati in **Application Service Elements (ASE)** correlati logicamente in modo da costituire moduli funzionali.



Una ASE rappresenta un gruppo di funzionalità di una AE, per uno scopo specifico.

- APs possono comunicare fra di loro solo utilizzando Application Entities dello stesso tipo.
- Le AEs non possono esistere indipendentemente dagli APs che esse supportano.
- AEs di tipo differente possono supportare uno stesso AP.
- AEs dello stesso tipo possono supportare differenti APs.

Gli APs cooperano attraverso link logici chiamati **Application Relationship**

- Le ASEs forniscono due tipi di servizi:
  - **Object Management** services
  - **Operational** services
- Le **Object Management ASEs** includono servizi che permettono di accedere e modificare l'ambiente operativo. In particolare creare, cancellare, identificare attributi di istanze di oggetti.
- Le **Operational ASEs** includono servizi per le interazioni fra i vari APOs durante il funzionamento del sistema. Possono essere classificate in:
  - **AP-ASEs** che forniscono i servizi per identificare e controllare le APs.
  - **AR-ASEs** che forniscono i servizi per stabilire e mantenere le Application Relationship necessarie per convogliare le FAL APDUs fra diverse APs.
  - **APO\_specific ASEs**, i cui servizi possono essere raggruppati nei seguenti tipi:
    - **Data Type ASEs**
    - **Variable ASEs**
    - **event ASEs**
    - **Load Region ASEs**
    - **Function Invocation ASEs**

# Management ASE

Supporta i servizi di management di base per la creazione e cancellazione della visione attraverso la rete di oggetti e l'accesso ai loro attributi.

**Create**: rende visibile un oggetto che non è attualmente accessibile attraverso la rete

**Delete** rende invisibile (quindi inaccessibile) un oggetto che è attualmente accessibile attraverso la rete.

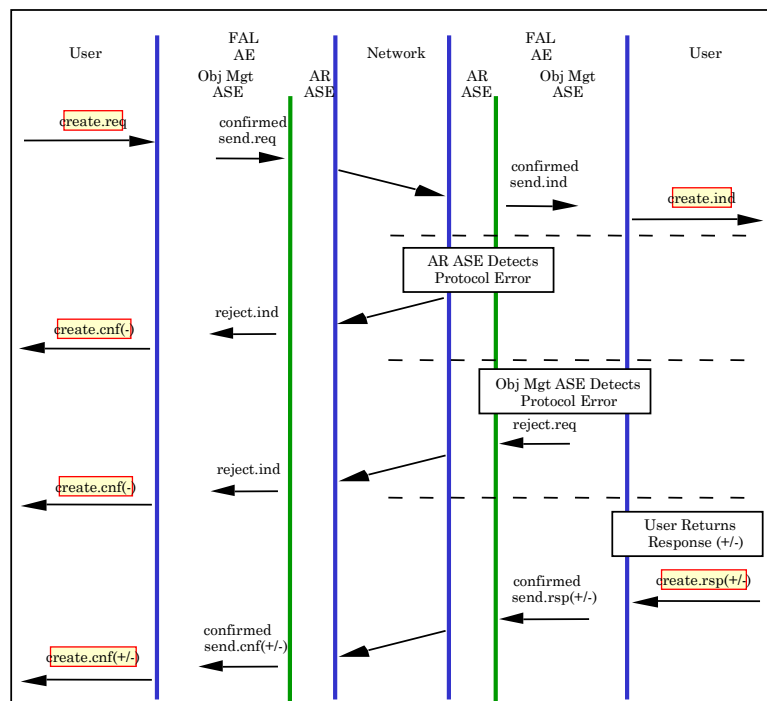
**GET Attributes** consente di leggere il valore di una lista di una lista di attributi di uno specifico oggetto

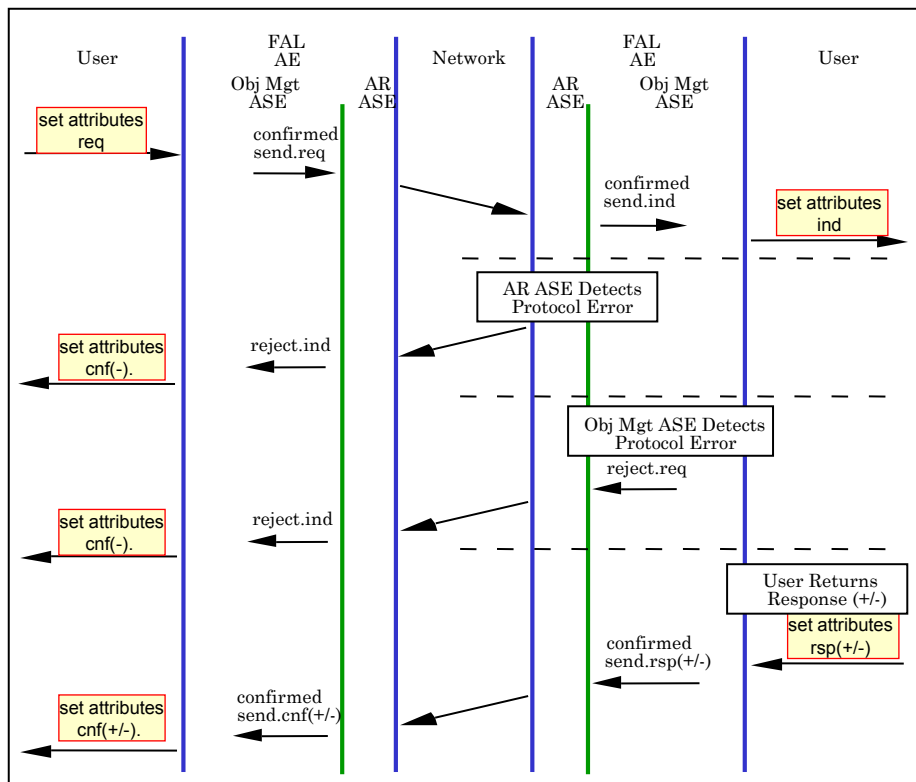
**GET Attributes list** consente di leggere il valore di una lista di attributi di una lista di oggetti

**SET Attributes** permette di aggiornare il valore di una lista di attributi di un oggetto

# Management ASE

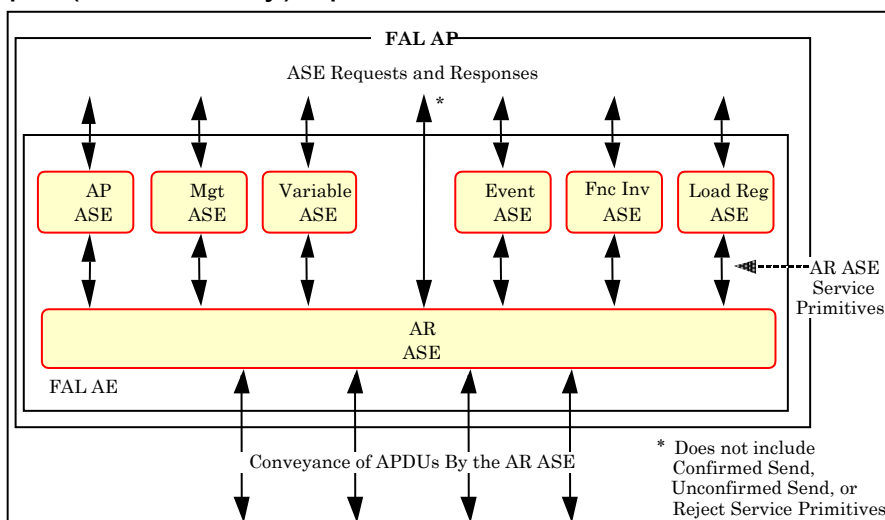
- I servizi forniti dal management ASE sono supportati da quelli, di comunicazione, forniti dalla Application Relationship ASE.





## APPLICATION RELATIONSHIP ASEs

L'Application Relationship (AR) è il canale di comunicazione fra due (one to one) o più (one to many) Aps.



**Application Relationship EndPoint (AREP)** rappresenta l'estremità di una relazione in ognuna delle APs coinvolte.

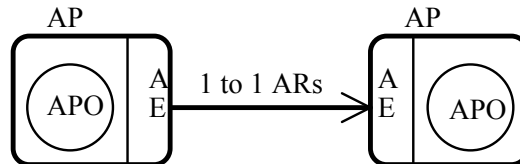
Un AREP è definito dal suo contesto che specifica il set di attributi che lo caratterizzano:

- compatibilità con altre AREP
- le operazioni che può eseguire

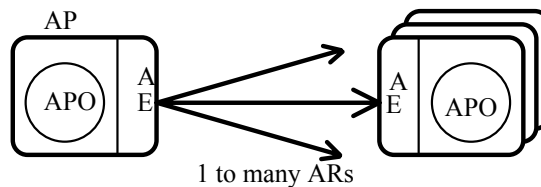
## Cardinalità di una AR

Rappresenta il numero di AREP remoti che possono ricevere l'APDU spedita dall'AREP.

- La Cardinalità è 1 in una AR di tipo Client/Server



- La Cardinalità è più di uno in una AR Publisher/Subscriber



## AR di tipo Client / Server

I possibili ruoli dell'AREP sono:

- **Client** - spedisce una richiesta di servizio confermato all'AREP Server.
- **Server** - restituisce una "response" al Client.
- **Peer** - opera come Client o Server (per quelle applicazioni in cui i ruoli sono interscambiabili).

## AR di tipo Publisher/Subscriber

Un Publisher spedisce dati per uno o più Subscriber.

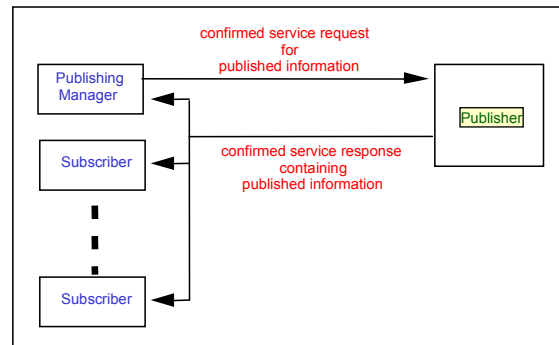
I possibili ruoli dell'AREP sono:

- **Publishing Manager** (1 per AR, gestisce la produzione dell'informazione, può coincidere col publisher)
- **Publisher** (1 per AR, produce le informazioni)
- **Subscriber** (1 o più per AR, riceve le informazioni)

## Modelli di AR di tipo Publisher/Subscriber

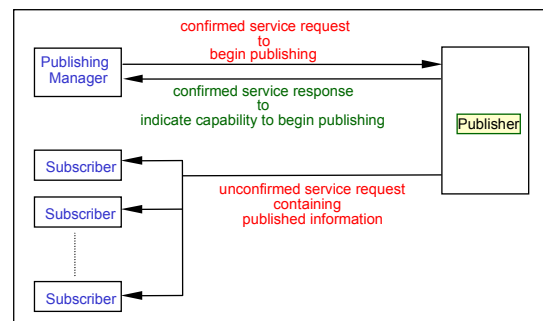
### Pull Model:

- il Publishing Manager sollecita il Publisher a spedire i dati.
- il Publisher risponde pubblicando i dati immediatamente(in multicast).
- il Publishing Manager ha il controllo dell'istante di inizio trasmissione.



### Push Model:

- La richiesta di pubblicazione e la trasmissione sono effettuate attraverso due servizi indipendenti
- Il Publishing manager chiede al Publisher di pubblicare i dati.
- Il Publisher conferma.
- Al tempo opportuno, il Publisher inizia la trasmissione dei dati (in multicast).
- Il publisher decide l'istante di inizio trasmissione.



## Politica di Trigger

Considerando le diverse esigenze dei processi applicativi, sono utili due diverse politiche per il trigger della trasmissione.

**User triggered:** (Asincrona) è l'utente che forza la trasmissione, spedendo un **Compel Request** a un DLL (richiesta esplicita).

**Network Scheduled:** è la rete che decide l'istante di trasmissione. In tal caso sono possibili due diverse soluzioni:

- **Schedulazione sincrona della rete:** si esegue l'attività schedulata dal DLL
- **Schedulazione asincrona della rete:** viene eseguita quando il DLL ha banda disponibile. In questo caso l'informazione è accodata dall'utente locale. La trasmissione è attivata da:
  - la rete, in modo locale, quando è possibile.
  - la rete, come conseguenza di un compel request spedito da un utente remoto.

# Servizi operazionali dell'AR

Supportati da ogni classe di AR:

- Unconfirmed Send
- AR Abort
- AR Compel
- AR Status

Servizi aggiuntivi:

- Confirmed Send
- Establish
- De-Establish
- Reject
- Get Buffered Message
- Schedule communication
- Cancel Schedule Communication

## Application Process ASE

Permette la gestione di un AP, che può essere specificato in termini di attributi e servizi.

**Attributi** - i più interessanti sono:

- lista degli indirizzi dei DLSAP disponibili, che possono essere usati con l'AP.
- Lista degli AREP che prendono parte ad una relazione già stabilita.
- Lista di AREP preconfigurati, disponibili per stabilire una relazione.

**Servizi:** esiste un servizio operativo (**Publish**) e diversi servizi di management già visti nel Management ASE (**Get attributes**, **Get attributes list**, **Set attributes**).

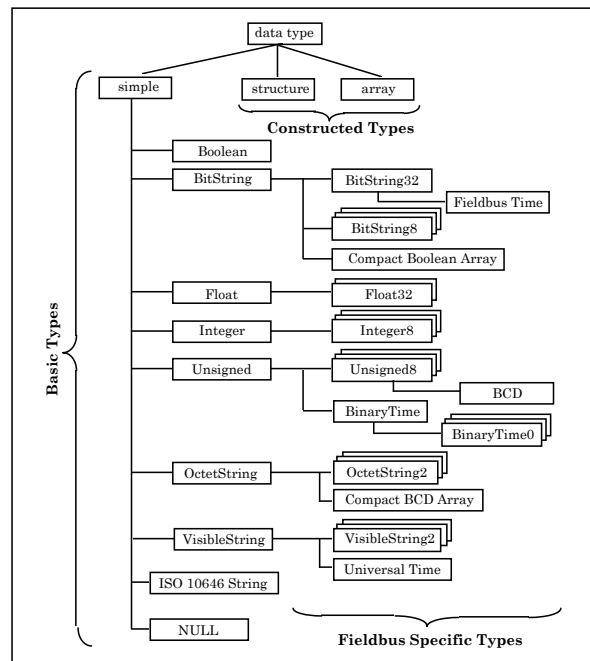
**PUBLISH:** è usato da un subscriber per chiedere ad un publisher AP di iniziare a pubblicare dei dati preconfigurati o di modificare i vincoli associati ad un oggetto pubblicato. Consente di entrare a far parte di una Relationship.



# Data Type model

Specifica la sintesi dei due applicativi scambiati attraverso il Fieldbus.

E' supportata la definizione ed il trasferimento di tipi base e composti.



## Variabile ASE

- Definisce gli attributi visibili attraverso la rete di una “variabile APO”.
- Fornisce un set di servizi per accedere alle variabili.

Una variabile:

- è univocamente identificabile
  - da un identificatore numerico (16bit), oppure
  - da un nome (una stringa) oppure
  - da un indirizzo (numerico o simbolico)
- ha associato un tipo di dato
  - definito dallo standard o dall'utente
  - totalmente accessibile o parzialmente (es un elemento di un array)
- ha un valore.

# Servizi Operazionali della Variable ASE

**Read / Write:** per una singola variabile o una lista **Predefinita** di variabili.

- Servizio usabile per code e buffers.
- Identifica una specifica variabile.
- Read restituisce un valore; per Write l'utente specifica un valore.
- Quando viene specificata una lista, l'accesso è "tutto o niente".
- A seconda del tipo di AR, nella "Read response" vengono fornite informazioni sulla "timeliness".

## Exchange:

- Servizio comune per code e Buffers.
- Combina una "Read" ed una "Write" in un solo servizio.
- Non è specificato l'ordine delle operazioni.
- Una delle operazioni può avere successo, l'altra fallire.

## Information Report:

- Servizio comune per code e buffers non dedicate.
- Fornisce il valore corrente di una variabile o di una lista predefinita.

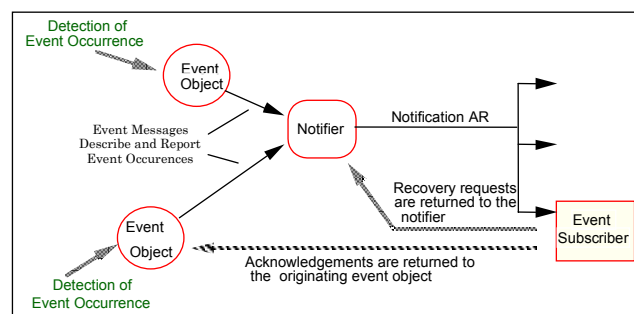
## Event ASE

Permette di notificare a processi remoti la rivelazione di un evento che prevede il monitoraggio di particolari variabili.

In tal modo è possibile rivelare situazioni critiche che possono avvenire nel sistema

**Event Object:** definisce il tipo di messaggi generati all'occorrenza di un evento:

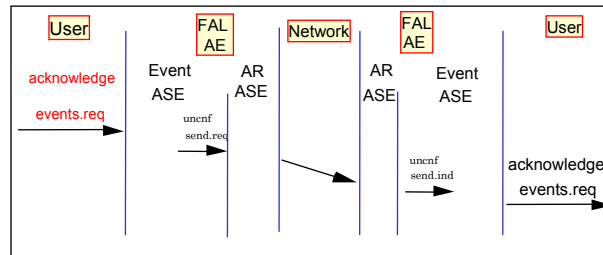
- Semplice messaggio
- Conteggio degli eventi
- Individuazione del tempo di occorrenza degli eventi
- Composito



**Notifier:** è responsabile di collezionare *event messages* dai vari *event objects* e distribuirli in una singola invocazione del servizio *Event notification*.

## Servizi Operazionali

**Acknowledge Event Messages:** è usato dal ricevitore di un event message per confermare la ricezione di un event message.



**Event Notification:** è usato dal “notifier” di un AP per notificare ad altri AP che si è verificato un evento.

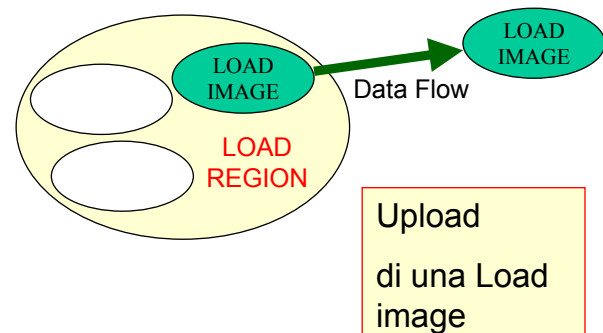
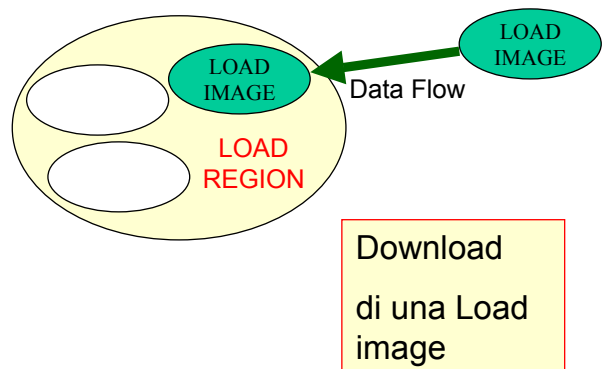
**Notification Recovery:** è usato per richiedere che un numero specificato di *event notifications* memorizzato in una AP sia trasmesso al richiedente.

**Get Event Summary:** è usato per richiedere un insieme di informazioni relative ad uno specifico evento. E' un servizio “tutto o niente” che opera attraverso code o buffers.

**Get Event Summary List:** è usato per richiedere informazioni relative ad uno specifico evento, indicate in una lista.

## Load Region ASE

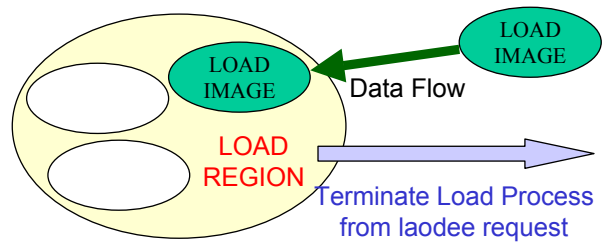
- La “**Load Region**” è un’area di memoria che può essere caricata o scaricata.
- il contenuto di una Load Region è definito “Load Image” e può contenere programmi o dati.
- il trasferimento di una Load Image nella Load Region è chiamato **Download**
- il trasferimento di una Load Image dalla Load Region è chiamato **Upload**
- I Servizi presenti permettono di iniziare il Dowload e l’Upload. Questo può essere attivato anche da una applicazione remota chiamata **Load Manager**.



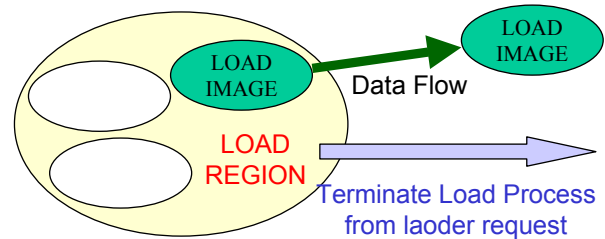
## Servizi della Load Region ASE

- **Initiate Download** consente di iniziare il processo di *Downloading* di una *Load Region*. Può essere attivato da un'applicazione remota (Load Manager) o dalla stessa *Load Region*.
- **Initiate Upload** consente di iniziare il processo di *Uploading* di una *Load Region*. Può essere attivato da un'applicazione remota (Load Manager) o dalla stessa *Load Region*.
- **Terminate Load Process** è usato dal ricevitore o dal mittente di una Load Image per chiedere di terminare il processo di Load.

La richiesta può essere inviata sia dal Load Manager che dalla Load Region.



La richiesta può essere inviata dal ricevente della Load Image: *Terminate Load Process from Loadee*



La richiesta può essere inviata dal mittente della Load Image: *Terminate Load Process from Loader*

## Servizi della Load region ASE

Altri servizi sono:

- **Push Segment** è usato per trasferire un singolo segmento di una Load Image attraverso una request e la relativa *indication*. Le primitive di *response* e *confirmation* sono usate per comunicare il successo o fallimento dell'operazione.

Il servizio può essere richiesto sia dalla Load Region che dal Load Manager.

- **Pull Segment** è usato per richiedere che uno specifico segmento di una Load Image venga consegnato nelle primitive di response e confirmation.

Il servizio può essere richiesto sia dalla *Load Region* che dal *Load Manager*.

- **Discard** è usato per chiedere che il contenuto di una *Load Region* venga scaricato. L'utente sottomette una *Discard request* alla sua Load Region ASE per chiedere di cancellare il contenuto della Load Region.

## Function Invocation ASE

E' usata per modellare il comportamento di Processi Software e di Task Operazionali (es. loop di regolazione).

Sono supportate due classi di oggetti:

- **State-oriented Function Invocation**
- **Stateless Action Object**
- **State-oriented Function Invocation:** è usata per modellare una Function Invocation di tipo "state-oriented" o altre funzioni le cui operazioni possono essere controllate durante la loro evoluzione.
- Non restituisce un valore come risultato.

### Servizi:

- Start
- Stop
- Resume
- Reset
- Kill
- Create
- Delete
- Get attributes
- Get attributes list
- Set attributes

## Function Invocation ASE

### Stateless Action Object:

- è usata per modellare una Function invocation che viene eseguita fino a completamento e non può essere interrotta.
- Può essere considerata un'azione atomica.
- Restituisce un valore come risultato quando modella una funzione software.
- Non restituisce alcun valore se usata per modellare procedure software.

### Servizi:

- Action invoke
- Action return
- Create
- Delete
- Get attributes
- Get attributes list
- Set attributes

# System Management

L'obiettivo del System Management è supportare l'integrazione dei singoli dispositivi in una applicazione di controllo, gestendo tutti i problemi che le attività possono comportare.

Operazioni fondamentali del System Management sono:

- Stabilire e ristabilire le comunicazioni
- Identificare le relazioni tra le diverse entità funzionali
- Allocare le risorse dei dispositivi
- Coordinare le varie funzioni del sistema.

Queste funzioni normalmente sono utilizzate quando:

- Una nuova applicazione è configurata per la prima volta o una nuova entità è aggiunta ad una applicazione esistente.
- Un applicazione sta per essere sostituita o modificata
- E' stato individuato un problema e bisogna prendere una decisione.

## Requisiti della gestione dei FieldBus

Le funzionalità del System Management si concretizzano nella realizzazione di attività differenti nelle diverse fasi operative.

### • Gestione della configurazione

- Inizializzazione della rete
- Rivelazione dell'aggiunta o eliminazione di un dispositivo nella rete
- identificazione di tale dispositivo
- Downloading / Uploading della configurazione di comunicazioni
- Downloading / Uploading dei processi applicativi

### • Configurazione del sistema

- Controllo e monitoraggio del dispositivo da una postazione remota del sistema, per leggere e modificarne lo stato.

### Gestione della sicurezza

- Capacità di rimanere in stato di sicurezza quando sono presenti più Gestori del sistema
- Capacità di prevenire l'invio di informazioni, da entità non autorizzate, ad agenti.

# Requisiti della Gestione dei FieldBus

- **Gestione della Dependability**
  - Rivelazione dei problemi
  - Notifica dei problemi
  - Modifica dei parametri della comunicazione in conseguenza di un Fault.
  - Lettura, modifica, e sincronizzazione del clock di sistema e del tempo delle applicazioni
- **Gestione delle performance**
  - Caratteristiche dei dati
  - Routing dei dati
  - Tipi di trasferimento dei dati
  - Consistenze temporali e spaziali
  - Schedulazione della trasmissione delle PDU
  - Distribuzione e sincronizzazione del tempo sulla rete
  - Applicazione della schedulazione