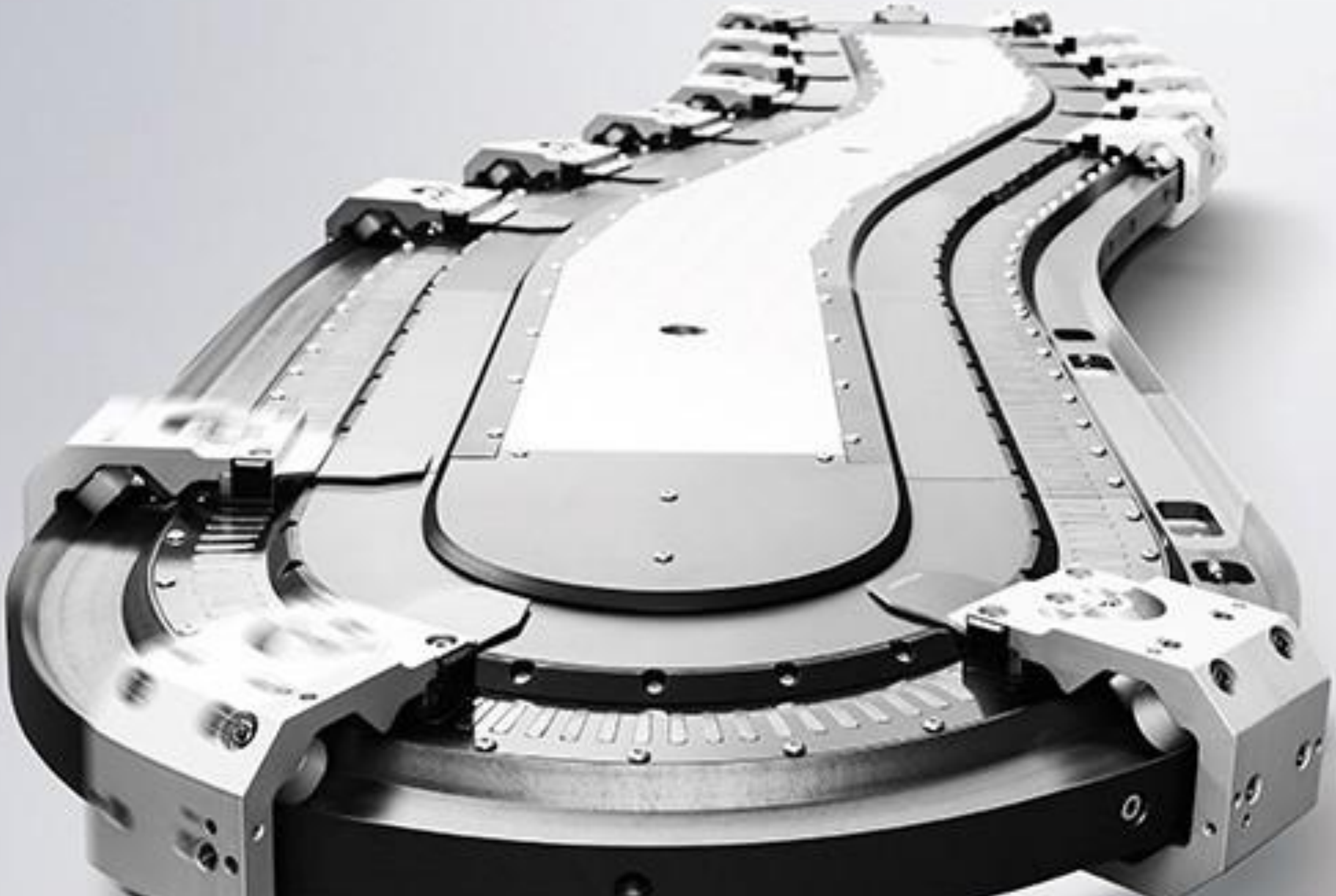
XTS TRANSPORT LAYER – DEMO APPLICATION

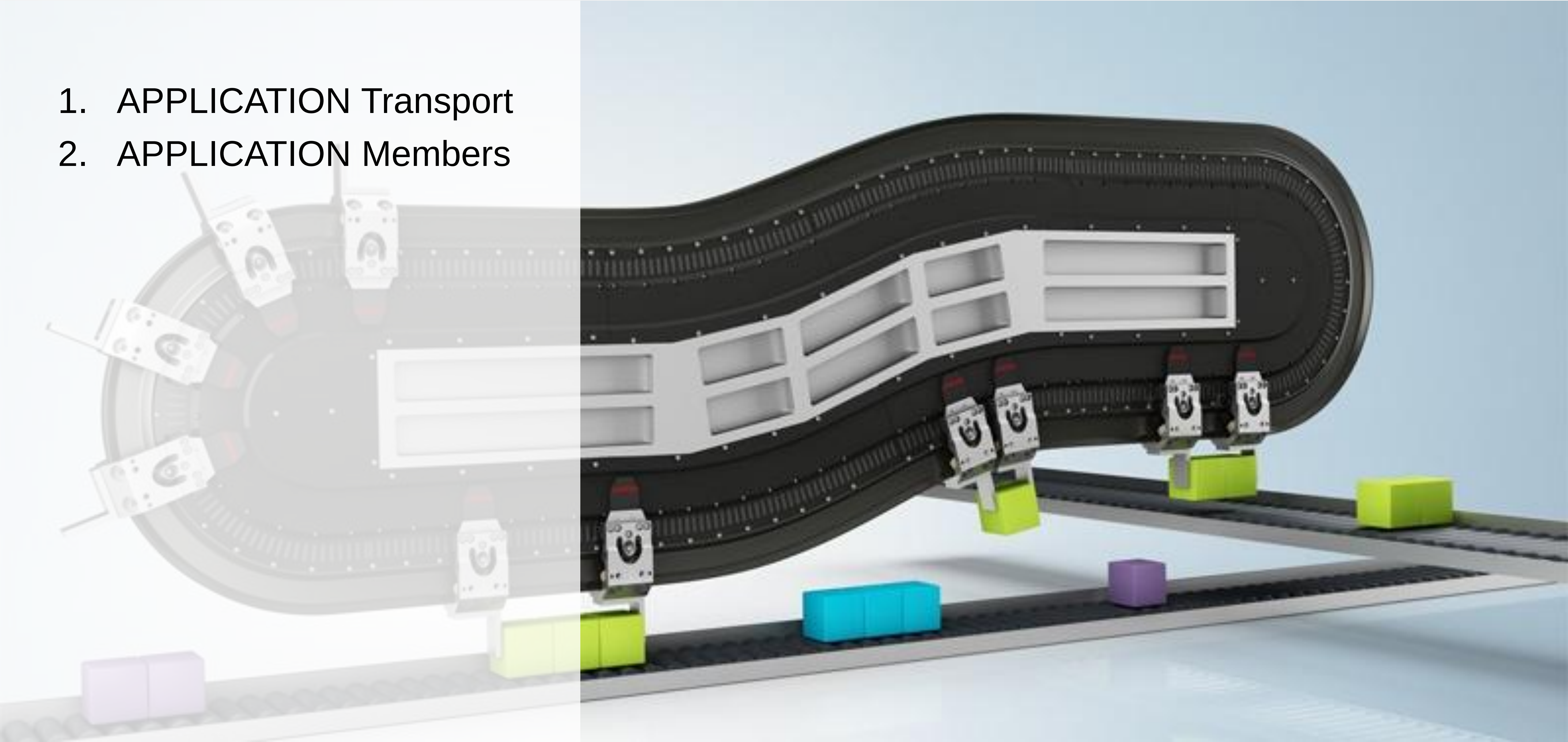
BECKHOFF



XTS TRANSPORT LAYER – DEMO APPLICATION

BECKHOFF

1. APPLICATION Transport
2. APPLICATION Members



DEMO APPLICATION

- **Transport specification:**
  - Output **500 – 550** movers/minute
  - Placing of N [1 to 12] items on N movers
    - Items are placed/dropped in INFEED
    - Transfer-System is either internal or external. (delivers parts in places)
    - N varies in place and count
  - Sending exact amount of movers for N items to INFEED
    - BUFFER_INFEED
      - placement mask is written by Transfer-System into LinkedList
      - Movers are targeted to INFEED stations according to mask.
      - BUFFER_INFEED writes mask to INFEED (LinkedList)
        - no bit → no mover (this is how gaps are closed)

DEMO APPLICATION

BECKHOFF

- **Transport specification:**
  - INFEED
    - sends 12 movers or less at a time
    - All one target: BUFFER_OUTFEED
  - BUFFER_OUTFEED
    - Works as distributor to OUTFEED
    - Requires only enable and automated handshakes

DEMO APPLICATION

- **Transport specification:**
  - OUTFEED (max 24 mover):
    - OUTFEED must work 24 mover in sync
    - OUTFEED must work 2x12 mover individually
    - → OUTFEED_R1
    - → OUTFEED_R2
  - Return track velo must be maximum
  - Movers must be accelerated to max velocity once the 180° curve is passed:
    - SENDER_FAST: sends all passing movers to SENDER_BUFFER_INFEED
  - Movers must not queue up in 180° curves:
    - SENDER_BUFFER_INFEED: only sends as many movers as requested in LinkedList entry.
      - INFEED mirrors its current bitmask when pulling in movers. (writes to LinkedList of SENDER_BUFFER_INFEED)

DEMO APPLICATION

BECKHOFF

- **Transport physics:**

  - 180° curve considerations due to Newton (twice).
    - G – gravity to a point
      - vertical system configuration
        - dL/dt = $\mathbf{r} \times \mathbf{G}$
        - May a mover halt during upwards movement?
          - → **NO**, keeping the mover in such a position leads to rising module temperature, which may lead to warnings from the I^2t model in the motormodule.
            Scope temperatures (ModuleInfoData) during installation as long as possible during runs to get an estimate of how the heat dissipates.

    - T – angular momentum
      - all system configurations
        - $L = m(\mathbf{r} \times \mathbf{v})$
        - → wear and tear on the guiding rails increase
          with speed, Moment of Inertia, and radius (center of curve)

DEMO APPLICATION

- **Transport physics:**

  - 180° curve considerations due to Fourier and Joule.
    - Q – from the continuum
      - Thermal dissipation calculations may make sense in a very small subset of all the machinery in the world, and this endeavour takes quite some time and structural, chemical knowledge usually found in labs.

      - Of course it is possible to use water-cooled mounting brackets
        - Expensive: you have to deal with mechanics for cooling liquid
        - Time consuming: you have to manage flow, pump, pressure measurement, hazard scenarios
        - This you do if you have no other choice or ventilation!

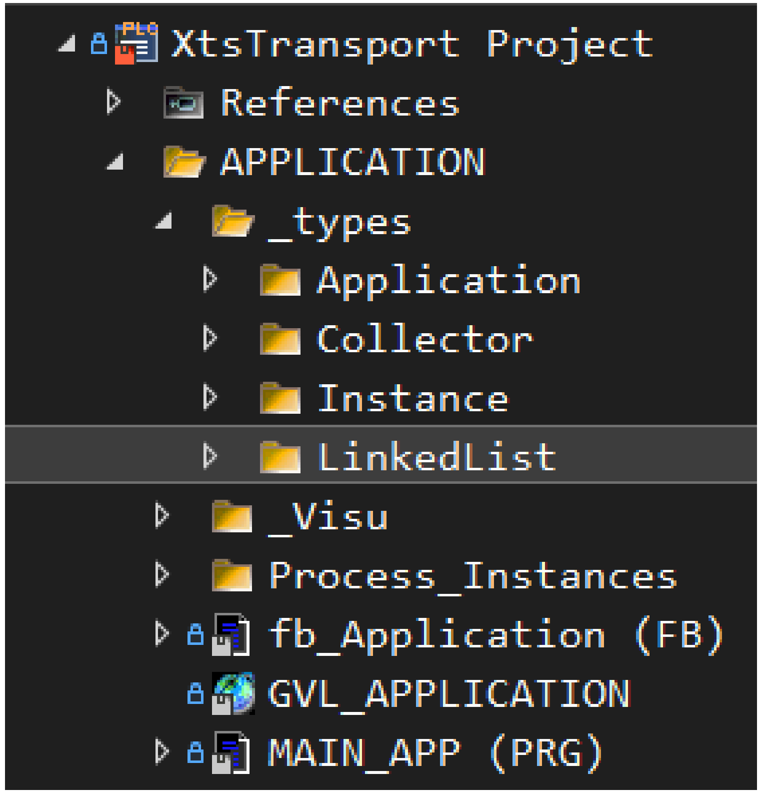
DEMO APPLICATION

BECKHOFF

- **Transport physics:**
  - 180° curve considerations due to Fourier and Joule.
    - Q – from the continuum
      - XTS_TRANSPORT_LAYER does not require any of the above
        - This application presents a very common configuration
        - For vertical systems in general: everything falls for gravity,
          - so make sure powers’ on long enough to withstand and reach level ground
        - In order to achieve stable process level handling, remember:
          - Do not cross bridges you cannot pass; Even Gandalf might succumb to heat.

DEMO APPLICATION

- **Transport Members:**
  - XTS_TRANSPORT_LAYER requires some AddOns
    - All AddOns are suggestions / examples
      - given the fact that heterogen means ‘Not all in TwinCAT
        → you’d might not be able to simply copy/paste
      - Of course I’ll continue the layered design and ctrl/state mechanics
        - The code already runs in high-speed, high-volume applications
          (this example is a real machine somewhere)
    - What is needed as AddOn?
      - ExternControl (you)
        - MAIN_APP(PRG)
        - fb_Application (does jobs similar to fb_TransportUnit)
          - Job description in the methods ;-)
        - fb_Instance (executing the transport, handshaked by fb_Application)
        - fb_ProcessCollector (grouping of stations, handshaked by assigned fb_Instance)



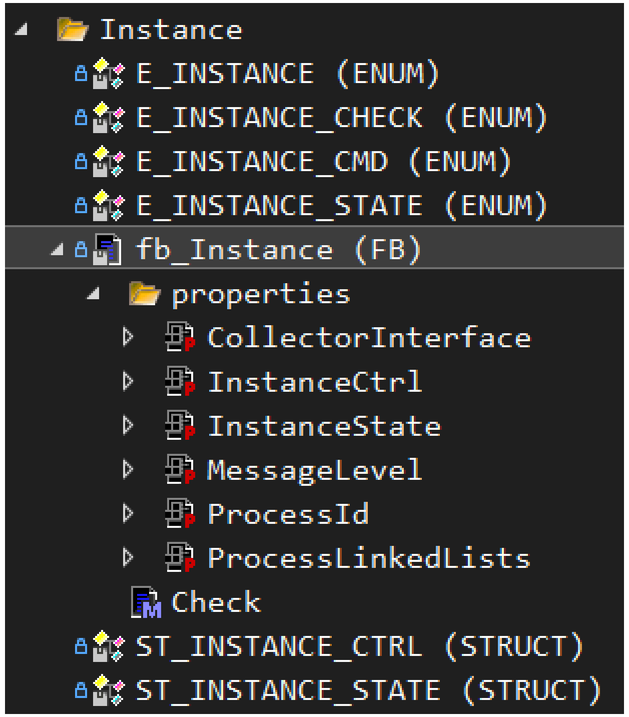
```
XtsTransport Project
    References
    APPLICATION
        _types
            Application
            Collector
            Instance
            LinkedList
        _Visu
        Process_Instances
        fb_Application (FB)
        GVL_APPLICATION
        MAIN_APP (PRG)
```

DEMO APPLICATION

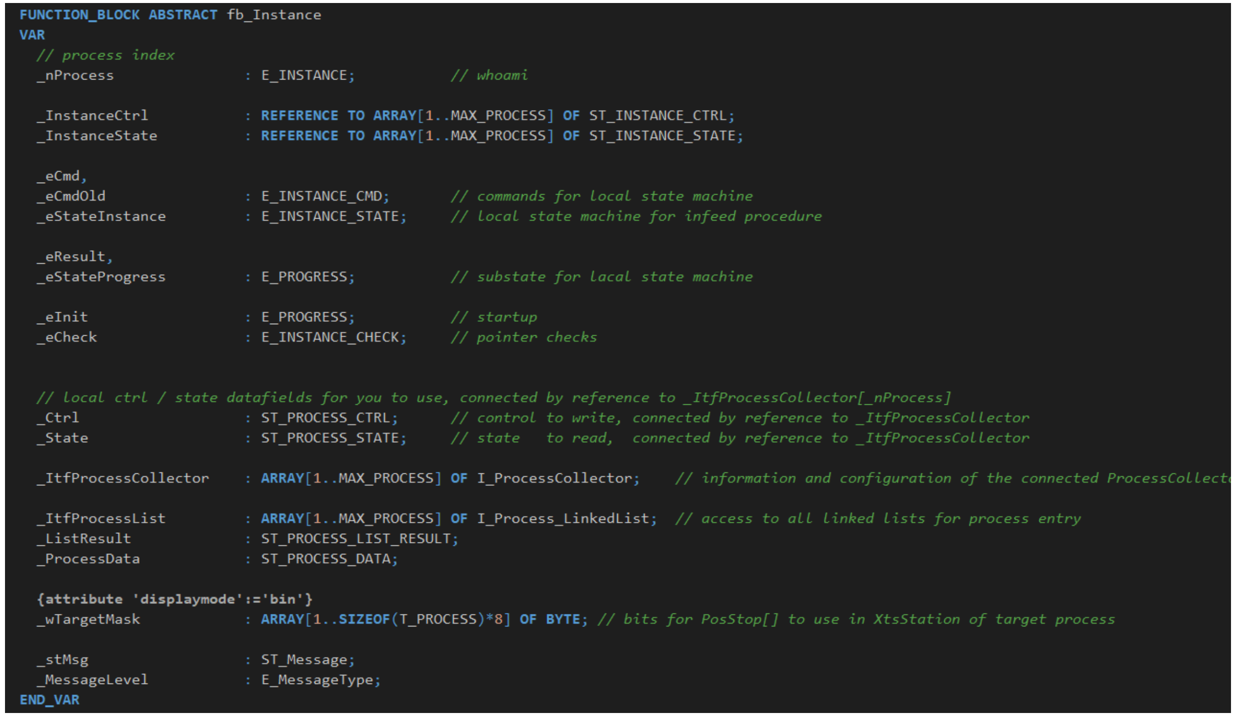
- **Transport Members:**
  - XTS_TRANSPORT_LAYER requires some AddOns (see GVL_APPLICATION)
    - How to write dedicated process transport procedures?
      - fb_Instance
        - ABSTRACT base class
        - Requires ID (index of fb_ProcessCollector[ ])
        - top level ctrl/state interface for transport of movers
        - Has access
        - Transport procedures of instances
        - handshaked by fb_Application.Work() (ST_INSTANCE_CTRL / ST_INSTANCE_STATE)
    - How to forward ListEntries from Transfer-System and/or processes?
      - fb_Process_LinkedListCtrl
      - Transfer-System uses BUFFER_INFEED’s LinkedList (AddTail) (ST_PROCESS_DATA)

DEMO APPLICATION

- **Transport Members:**
  - XTS_TRANSPORT_LAYER requires some AddOns (see GVL_APPLICATION)
    - To see you have to look (below)



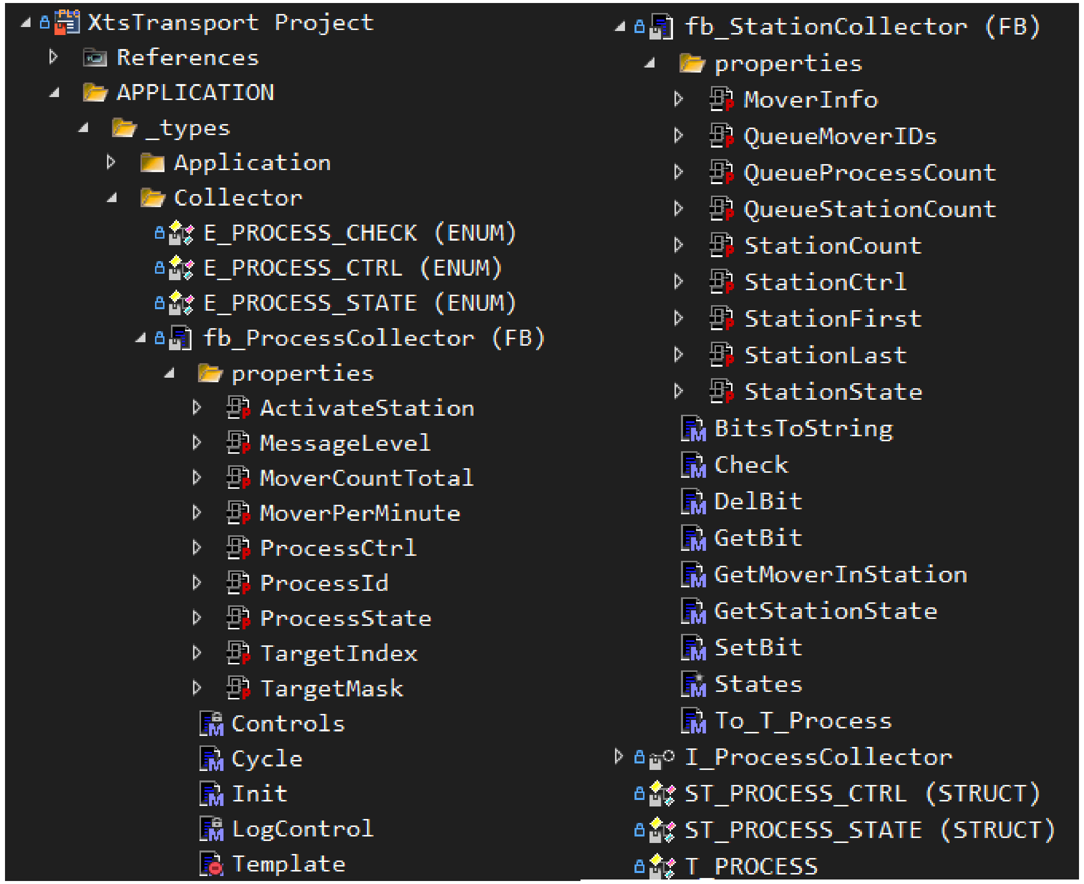
```
◢ Instance
    E_INSTANCE (ENUM)
    E_INSTANCE_CHECK (ENUM)
    E_INSTANCE_CMD (ENUM)
    E_INSTANCE_STATE (ENUM)
  ◢ fb_Instance (FB)
      ◢ properties
          ▷ CollectorInterface
          ▷ InstanceCtrl
          ▷ InstanceState
          ▷ MessageLevel
          ▷ ProcessId
          ▷ ProcessLinkedLists
        Check
    ST_INSTANCE_CTRL (STRUCT)
    ST_INSTANCE_STATE (STRUCT)
```



```
FUNCTION_BLOCK ABSTRACT fb_Instance
VAR
  // process index
  _nProcess               : E_INSTANCE;           // whoami

  _InstanceCtrl           : REFERENCE TO ARRAY[1..MAX_PROCESS] OF ST_INSTANCE_CTRL;
  _InstanceState          : REFERENCE TO ARRAY[1..MAX_PROCESS] OF ST_INSTANCE_STATE;

  _eCmd,
  _eCmdOld                : E_INSTANCE_CMD;       // commands for local state machine
  _eStateInstance         : E_INSTANCE_STATE;     // local state machine for infeed procedure

  _eResult,
  _eStateProgress         : E_PROGRESS;           // substate for lacal state machine

  _eInit                  : E_PROGRESS;           // startup
  _eCheck                 : E_INSTANCE_CHECK;     // pointer checks


  // local ctrl / state datafields for you to use, connected by reference to _ItfProcessCollector[_nProcess]
  _Ctrl                   : ST_PROCESS_CTRL;      // control to write, connected by reference to _ItfProcessCollector
  _State                  : ST_PROCESS_STATE;     // state   to read,  connected by reference to _ItfProcessCollector

  _ItfProcessCollector    : ARRAY[1..MAX_PROCESS] OF I_ProcessCollector;    // information and configuration of the connected ProcessCollecto

  _ItfProcessList         : ARRAY[1..MAX_PROCESS] OF I_Process_LinkedList;  // access to all linked lists for process entry
  _ListResult             : ST_PROCESS_LIST_RESULT;
  _ProcessData            : ST_PROCESS_DATA;

  {attribute 'displaymode':='bin'}
  _wTargetMask            : ARRAY[1..SIZEOF(T_PROCESS)*8] OF BYTE; // bits for PosStop[] to use in XtsStation of target process

  _stMsg                  : ST_Message;
  _MessageLevel           : E_MessageType;
END_VAR
```

DEMO APPLICATION

- **Transport Members:**
  - XTS_TRANSPORT_LAYER requires some AddOns (see GVL_APPLICATION)
    - Look over there →
    - fb_ProcessCollector **extends** fb_StationCollector



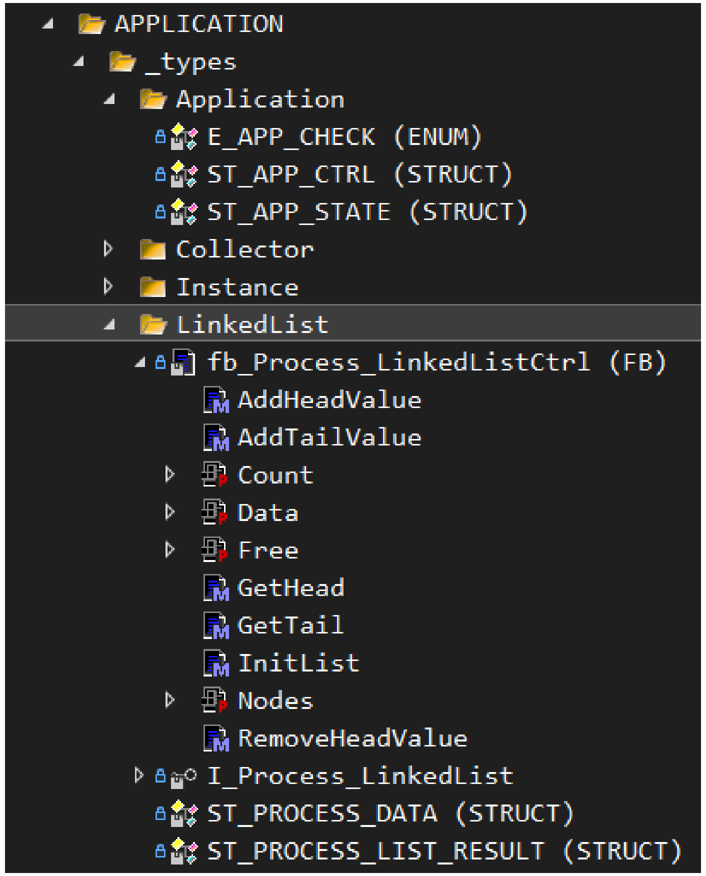
```
◢ XtsTransport Project
   ▷ References
   ◢ APPLICATION
      ◢ _types
         ▷ Application
         ◢ Collector
              E_PROCESS_CHECK (ENUM)
              E_PROCESS_CTRL (ENUM)
              E_PROCESS_STATE (ENUM)
           ◢ fb_ProcessCollector (FB)
              ◢ properties
                 ▷ ActivateStation
                 ▷ MessageLevel
                 ▷ MoverCountTotal
                 ▷ MoverPerMinute
                 ▷ ProcessCtrl
                 ▷ ProcessId
                 ▷ ProcessState
                 ▷ TargetIndex
                 ▷ TargetMask
                Controls
                Cycle
                Init
                LogControl
                Template
```

```
◢ fb_StationCollector (FB)
   ◢ properties
      ▷ MoverInfo
      ▷ QueueMoverIDs
      ▷ QueueProcessCount
      ▷ QueueStationCount
      ▷ StationCount
      ▷ StationCtrl
      ▷ StationFirst
      ▷ StationLast
      ▷ StationState
     BitsToString
     Check
     DelBit
     GetBit
     GetMoverInStation
     GetStationState
     SetBit
     States
     To_T_Process
▷ I_ProcessCollector
  ST_PROCESS_CTRL (STRUCT)
  ST_PROCESS_STATE (STRUCT)
  T_PROCESS
```

DEMO APPLICATION

- **Transport Members:**
  - XTS_TRANSPORT_LAYER requires some AddOns (see GVL_APPLICATION)
    - This one looks familiar →

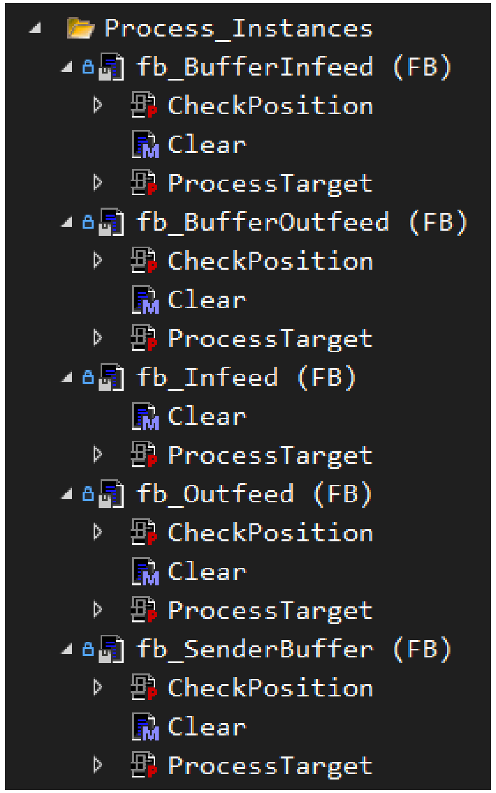
(I was just too lazy for further abstraction,
hence an adjusted copy of fb_Station_LinkedListCtrl)



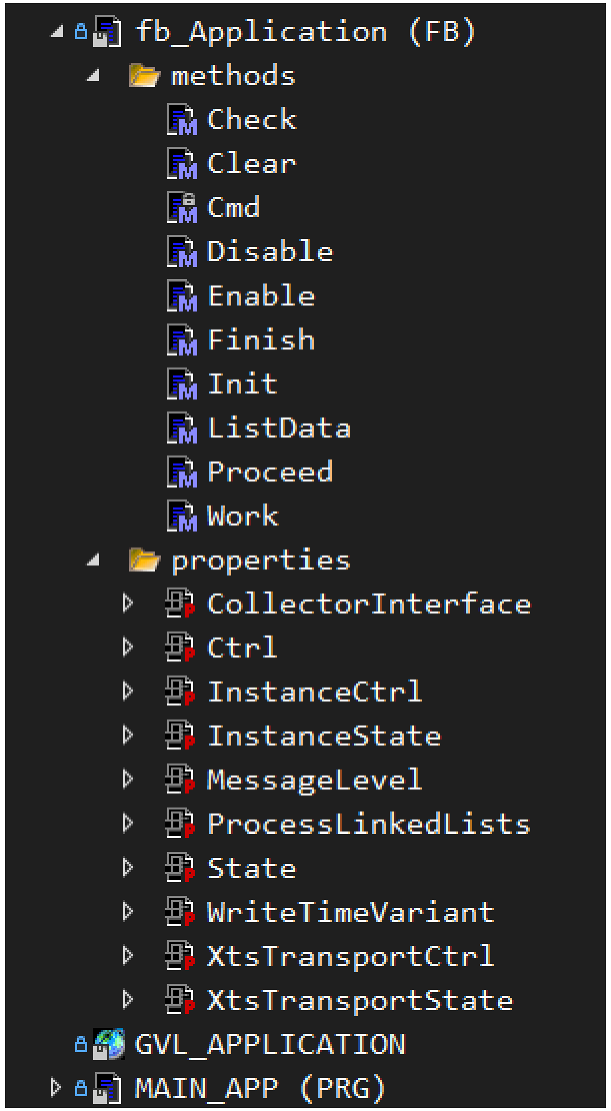
```
APPLICATION
  _types
    Application
      E_APP_CHECK (ENUM)
      ST_APP_CTRL (STRUCT)
      ST_APP_STATE (STRUCT)
    Collector
    Instance
    LinkedList
      fb_Process_LinkedListCtrl (FB)
        AddHeadValue
        AddTailValue
        Count
        Data
        Free
        GetHead
        GetTail
        InitList
        Nodes
        RemoveHeadValue
      I_Process_LinkedList
      ST_PROCESS_DATA (STRUCT)
      ST_PROCESS_LIST_RESULT (STRUCT)
```

DEMO APPLICATION

- **Transport Members:**
  - XTS_TRANSPORT_LAYER requires some AddOns (see GVL_APPLICATION)
    - Run application run →



```
◢ Process_Instances
  ◢ fb_BufferInfeed (FB)
    ▷ CheckPosition
      Clear
    ▷ ProcessTarget
  ◢ fb_BufferOutfeed (FB)
    ▷ CheckPosition
      Clear
    ▷ ProcessTarget
  ◢ fb_Infeed (FB)
      Clear
    ▷ ProcessTarget
  ◢ fb_Outfeed (FB)
    ▷ CheckPosition
      Clear
    ▷ ProcessTarget
  ◢ fb_SenderBuffer (FB)
    ▷ CheckPosition
      Clear
    ▷ ProcessTarget
```

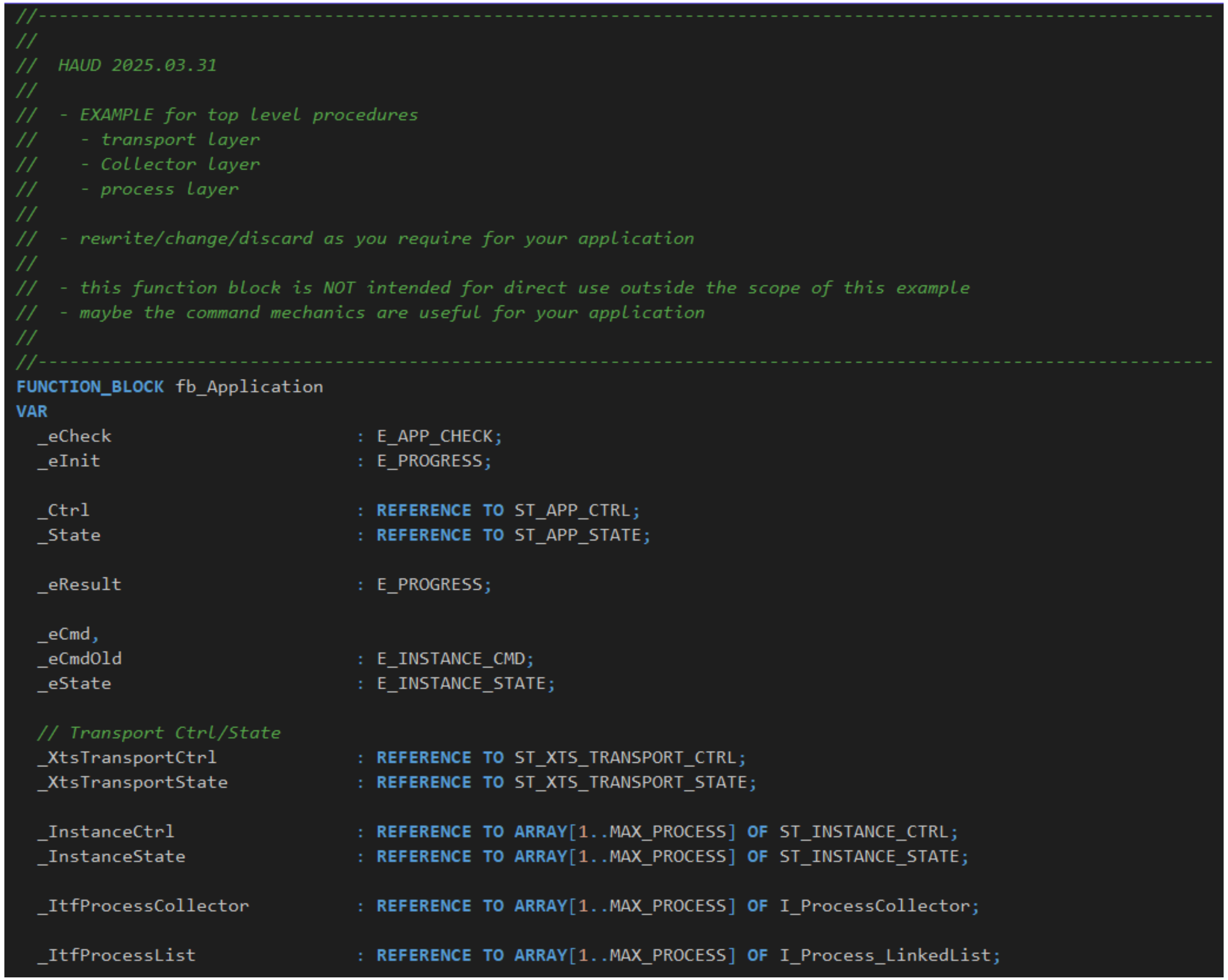


```
◢ fb_Application (FB)
  ◢ methods
      Check
      Clear
      Cmd
      Disable
      Enable
      Finish
      Init
      ListData
      Proceed
      Work
  ◢ properties
    ▷ CollectorInterface
    ▷ Ctrl
    ▷ InstanceCtrl
    ▷ InstanceState
    ▷ MessageLevel
    ▷ ProcessLinkedLists
    ▷ State
    ▷ WriteTimeVariant
    ▷ XtsTransportCtrl
    ▷ XtsTransportState
  GVL_APPLICATION
▷ MAIN_APP (PRG)
```

DEMO APPLICATION

BECKHOFF

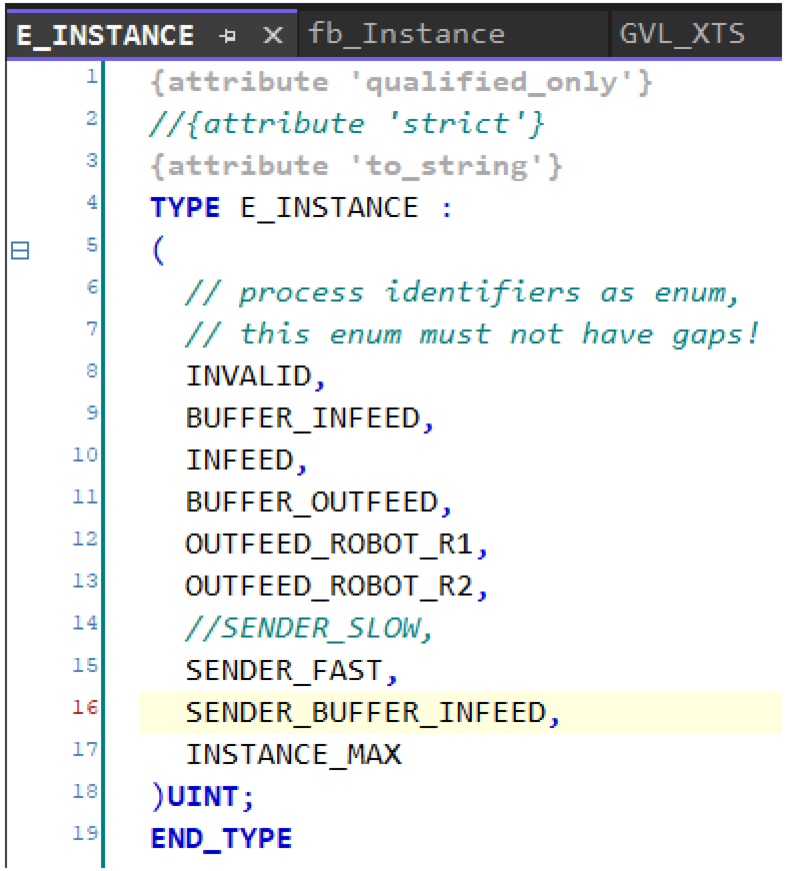
- **Transport Members:**
  - Do not blink



```
//----------------------------------------------------------------------------------------------------------------
//
//  HAUD 2025.03.31
//
//  - EXAMPLE for top level procedures
//    - transport layer
//    - Collector layer
//    - process layer
//
//  - rewrite/change/discard as you require for your application
//
//  - this function block is NOT intended for direct use outside the scope of this example
//  - maybe the command mechanics are useful for your application
//
//----------------------------------------------------------------------------------------------------------------
FUNCTION_BLOCK fb_Application
VAR
  _eCheck                        : E_APP_CHECK;
  _eInit                         : E_PROGRESS;

  _Ctrl                          : REFERENCE TO ST_APP_CTRL;
  _State                         : REFERENCE TO ST_APP_STATE;

  _eResult                       : E_PROGRESS;

  _eCmd,
  _eCmdOld                       : E_INSTANCE_CMD;
  _eState                        : E_INSTANCE_STATE;

  // Transport Ctrl/State
  _XtsTransportCtrl              : REFERENCE TO ST_XTS_TRANSPORT_CTRL;
  _XtsTransportState             : REFERENCE TO ST_XTS_TRANSPORT_STATE;

  _InstanceCtrl                  : REFERENCE TO ARRAY[1..MAX_PROCESS] OF ST_INSTANCE_CTRL;
  _InstanceState                 : REFERENCE TO ARRAY[1..MAX_PROCESS] OF ST_INSTANCE_STATE;

  _ItfProcessCollector           : REFERENCE TO ARRAY[1..MAX_PROCESS] OF I_ProcessCollector;

  _ItfProcessList                : REFERENCE TO ARRAY[1..MAX_PROCESS] OF I_Process_LinkedList;
```

DEMO APPLICATION

- **Transport Members:**
  - How many process instances do I need? (see E_INSTANCE)
    - **TransferSystem (fb_Application.ListData())**
      - Not a process per se, but required to get things going
      - simulated list entries for bitmask of XtsStations
      - writes into LinkedList of BUFFER_INFEED (bitmask with used bits 0-11)
    - **BUFFER_INFEED (fb_BufferInfeed)**
      - nProcessId as Input property
      - Checks linked list for new entries
      - sends required amount of movers to INFEED
      - Writes into LinkedList of INFEED



```
{attribute 'qualified_only'}
//{attribute 'strict'}
{attribute 'to_string'}
TYPE E_INSTANCE :
(
  // process identifiers as enum,
  // this enum must not have gaps!
  INVALID,
  BUFFER_INFEED,
  INFEED,
  BUFFER_OUTFEED,
  OUTFEED_ROBOT_R1,
  OUTFEED_ROBOT_R2,
  //SENDER_SLOW,
  SENDER_FAST,
  SENDER_BUFFER_INFEED,
  INSTANCE_MAX
)UINT;
END_TYPE
```

DEMO APPLICATION

BECKHOFF

- **Transport Members:**

    - How many process instances do I need?
        - **INFEED (fb_Infeed)**
            - Checks LinkedList for new entries
            - Starts infeed of movers according to bitmask in list entry.
            - Writes a copy of the list entry into LinkedList of SENDER_BUFFER_INFEED
            - WORK, starts process
            - FINISH, ends process
            - Sends out movers to BUFFER_OUTFEED

        - **BUFFER_OUTFEED (fb_BufferOutfeed)**
            - checks the assigned XtsStation for new mover
            - Sends single mover to output
            - Internal counter is keeping track of the 24 possible target XtsStations

DEMO APPLICATION

- **Transport Members:**
  - How many process instances do I need?
    - **OUTFEED_R1 (fb_Outfeed)**
      - Checks if all 12 XtsStations have a mover waiting
      - Infeed of 12 movers
      - Work 12 movers
      - Send 12 movers to SENDER_FAST
    - **OUTFEED_R2 (fb_Outfeed)**
      - Checks if all 12 XtsStations have a mover waiting
      - Infeed of 12 movers
      - Work 12 movers
      - Send 12 movers to SENDER_FAST

DEMO APPLICATION

- **Transport Members:**
  - How many process instances do I need?
    - **SENDER_FAST (fb_SenderBuffer)**
      - checks the assigned XtsStation for new mover
      - Sends mover to SENDER_BUFFER_INFEED
    - **SENDER_BUFFER_INFEED (fb_SenderBuffer)**
      - Checks ProcessList for new entry from INFEED
      - Sends the amount (written in ListEntry) of movers to BUFFER_INFEED