This asset describes different scenarios related to the use of containers in Cobol CICS environment.

It includes :

* This Readme document
* A Presentation with evidence of the operation sequence and data flow in the different scenarios
* templates of involved Cobol modules to show how the scenarios can be put in place from a technical point of view.

Channels and containers provide a powerful way to pass data between programs, in amounts that far exceed the 32 KB limit that applies to COMMAREAs.

All the scenarios include a client program (COBSTART) that calls an “orchestrator” module (COBMAIN) providing the name of the service that has to be executed next (COBCHAN or COBCICS2).

A COBOL template for COBSTART is provided for debug purposes.

All the scenarios include CICS Transaction Gateway between Distributed application server and CICS.

https://w3-03.ibm.com/services/lighthouse/documents/145745

# First Scenario – Single CICS – No specific container for error data

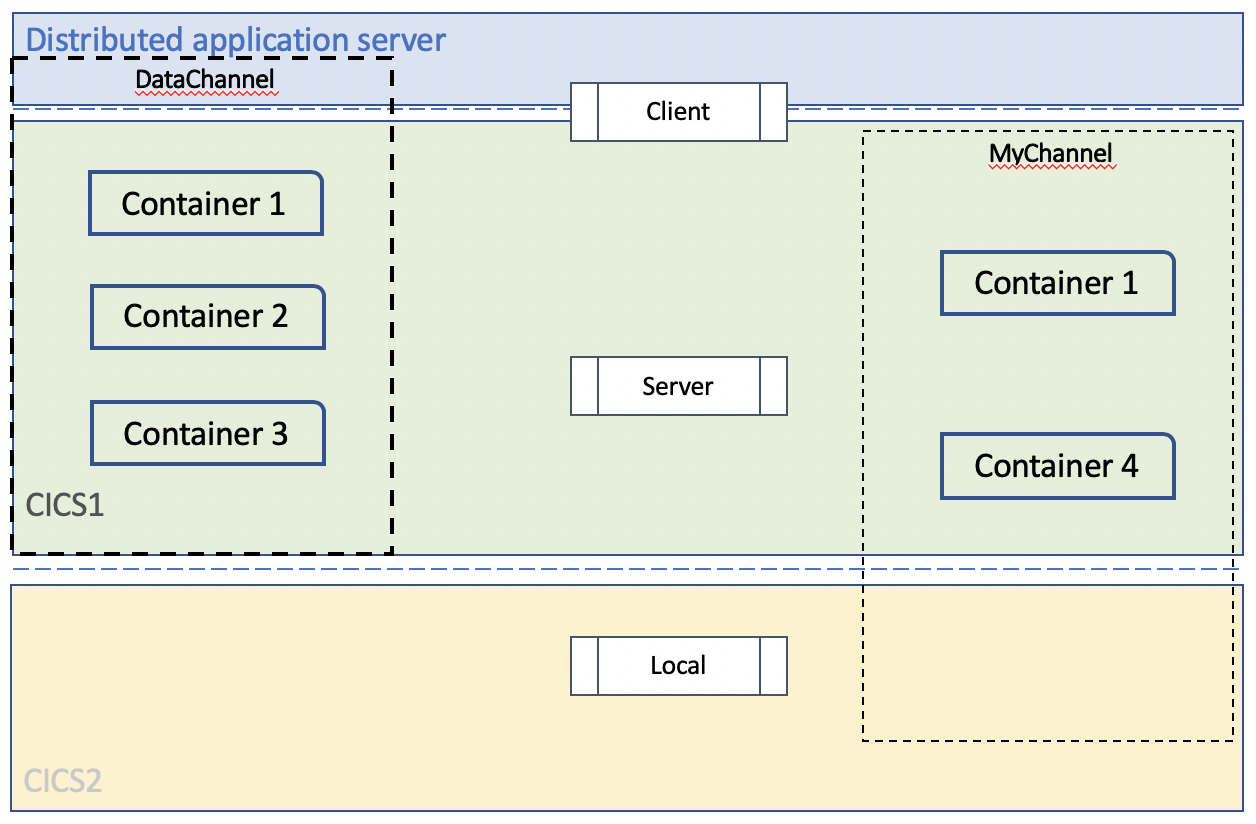
In this scenario, a client program, based on an input value, constructs a channel, passes it to a server program, and retrieves the server's output.

The server program retrieves data from the channel's containers, constructs a new channel (not available to client program) passes the new channel to a local module, retrieves local modules’ output and returns output to the client.

The local module resides in the same CICS environment as the Server module.

Two containers are globally put in the first Channel: one for technical data, one for input/output data.

One container is put in the new Channel. For both input and output data



Containers :

|  |  |
| --- | --- |
| **Container id** | **Data** |
| Container 1 | Technical Data |
| Container 2 | Input/Output Data |
| Container 3 | Input/Output Local Data |

Related Templates provided:

|  |  |  |
| --- | --- | --- |
| **Module type** | **Template** | **Option** |
| Client | COBSTART | ActionType = 1 (CHAN) |
| Server | COBMAIN | ActionType = 1 (CHAN) |
| Local | COBCHAN |  |

# Second Scenario – Multiple CICS – No specific container for error

In this scenario, a client program constructs a channel, passes it to a server program, and retrieves the server’s output.

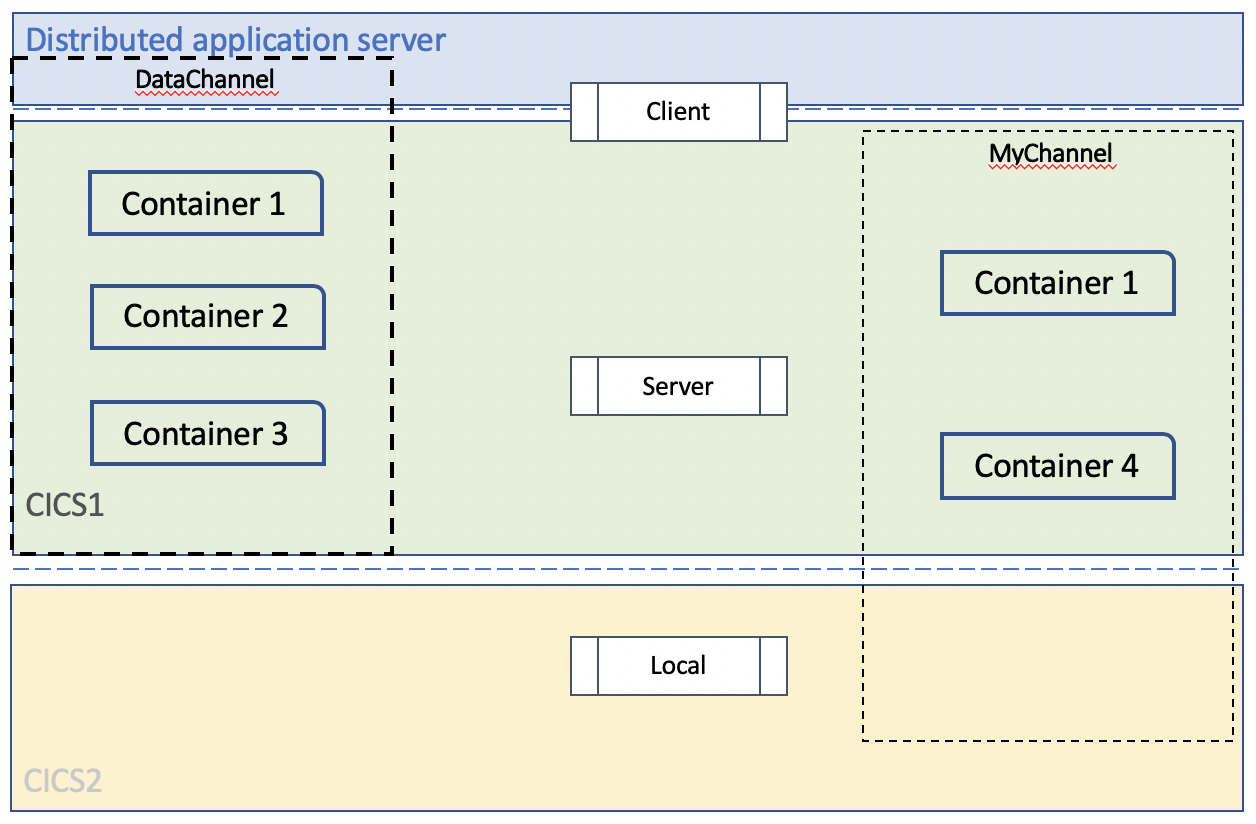
The server program retrieves data from the channel’s containers, constructs a new channel (not available to client program) passes it to a local module, retrieves local modules’ output and returns output to the client.

The local module retrieves data from the new channel’s containers and returns output to the server program.

The local module resides in a different CICS than the Server module.

Three containers are globally put in the first Channel: one for technical data, one for input data and one for output data.

One container is put in the new Channel. For both input and output data



Containers :

|  |  |
| --- | --- |
| **Container id** | **Data** |
| Container 1 | Technical Data |
| Container 2 | Input Data |
| Container 3 | Output Data |
| Container 4 | Input/Output Local Data |

Related Templates provided:

|  |  |  |
| --- | --- | --- |
| **Module type** | **Template** | **Option** |
| Client | COBSTART | ActionType = 2 (CICS2) |
| Server | COBMAIN | ActionType = 2 (CICS2) |
| Local | COBCICS2 |  |

# Third Scenario – Multiple CICS – Specific Error container

This scenario is similar to the second, but with use of a specific container for error management: a client program constructs a channel, passes it to a server program, and retrieves the server’s output.

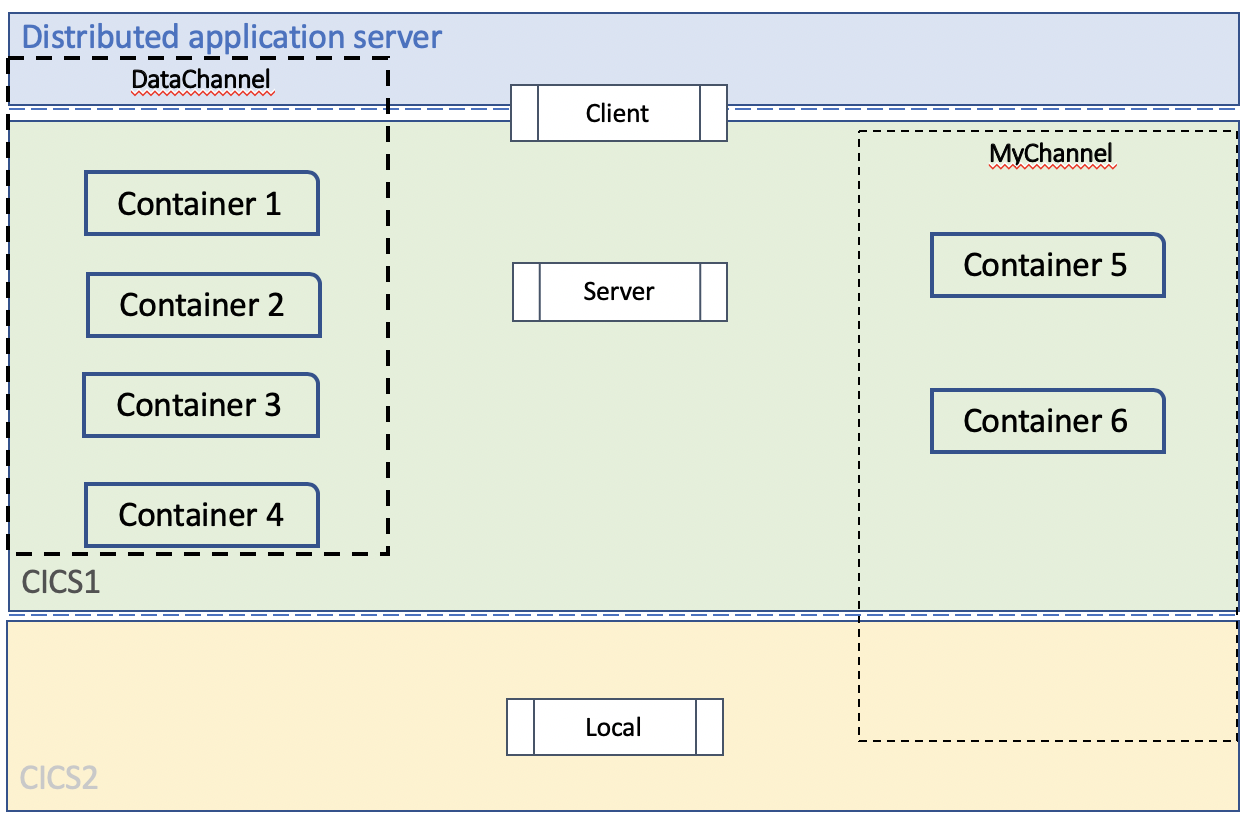
The server program retrieves data from the channel’s containers, constructs a new channel (not available to client program) passes it to a local module, retrieves local modules’ output and returns output to the client.

The local module retrieves data from the new channel’s containers and returns output to the server program.

Based on an input parameter, the local module resides in the same CICS as the Server module or in a different one.

Four containers are globally put in the first Channel: one for technical data, one for input data, one for output data and one for error data. In case of severe error, only the Error container is returned. In case of warning, both error and output containers are returned, otherwise only the output container is returned.

Two containers are put in the new Channel. One for input and one output data



Containers :

|  |  |
| --- | --- |
| **Container id** | **Data** |
| Container 1 | Technical Data |
| Container 2 | Input Data |
| Container 3 | Output Data |
| Container 4 | Error Data |
| Container 5 | Input Local Data |
| Container 6 | Output Local Data |

Related Templates provided:

|  |  |  |
| --- | --- | --- |
| **Module type** | **Template** | **Option** |
| Client | COBSTART | ActionType = 3 (BESTP) |
| Server | COBMAIN | ActionType = 3 (BESTP) |
| Local | COBCICS2 |  |