

The Use of EAST in an Operational EO Archive

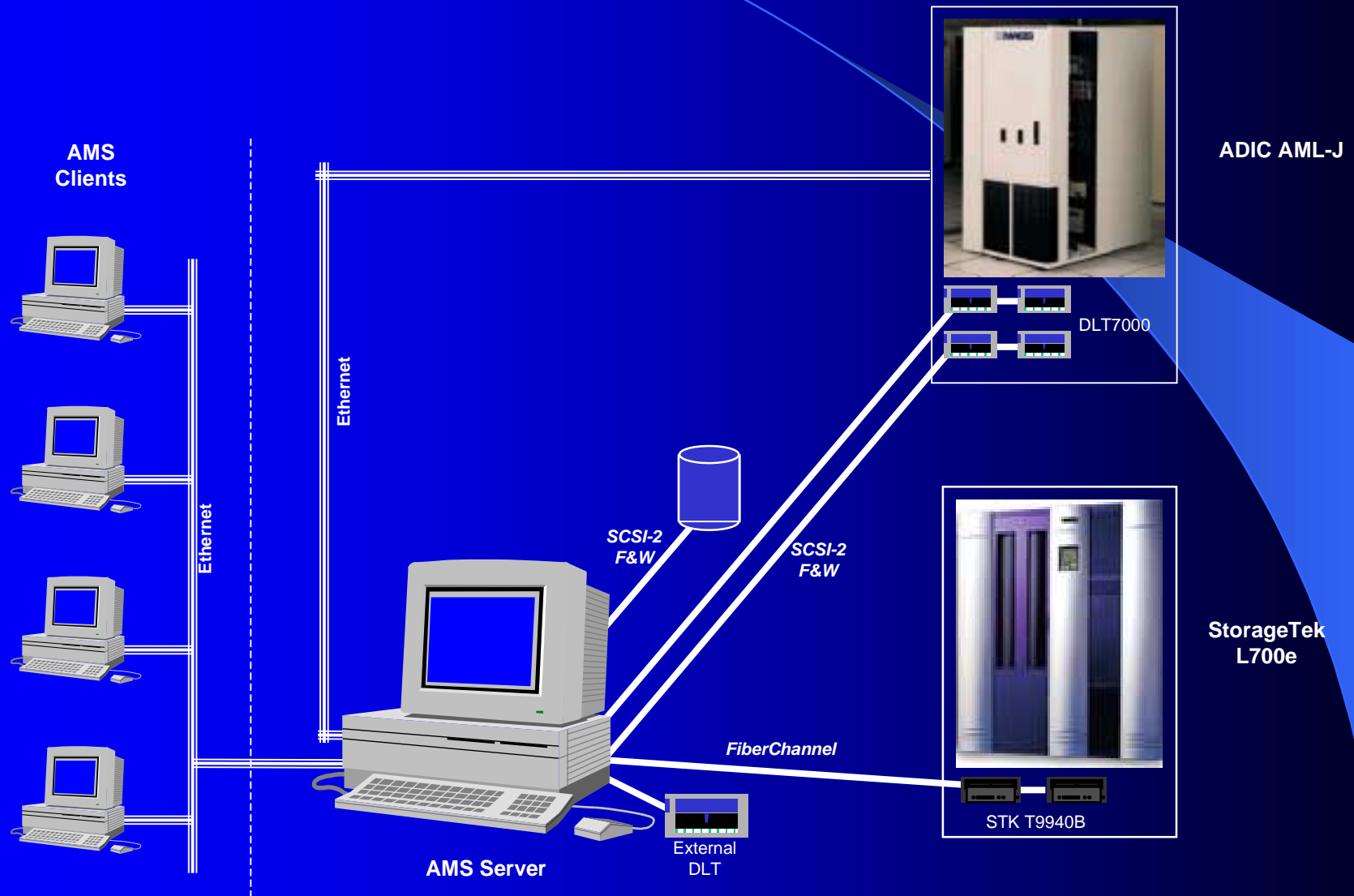
Presentation for the EAST/DEBAT Workshop
ESTEC 19 February 2003

Gian Maria Pinna
ESA-ESRIN

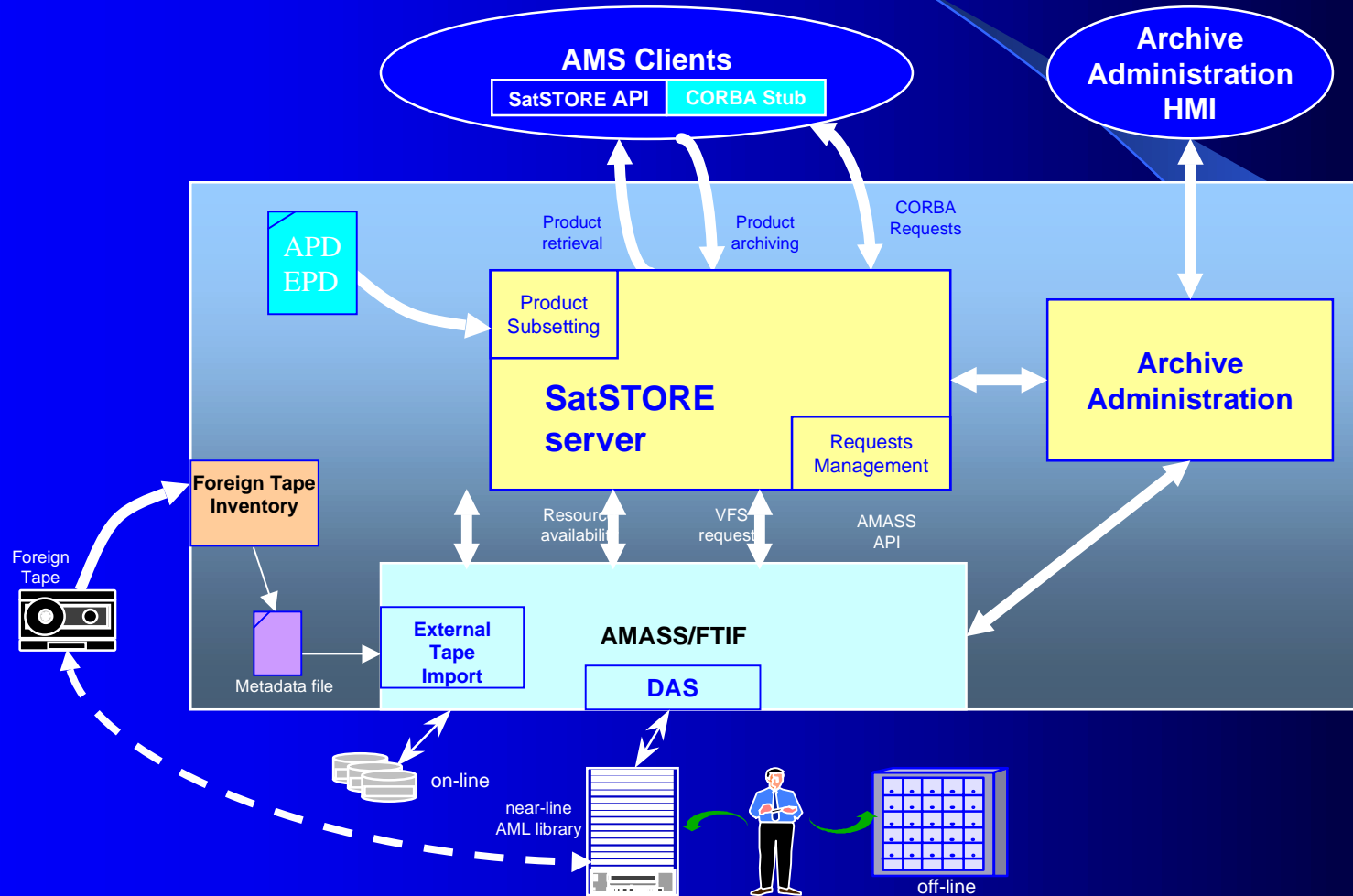
ESA Archive Management System

- Dedicated to long term data archiving
- On-line access via client-server services
- API and CORBA services for data archiving and retrieval
- Internal storage organization hidden to clients
- Data security with access rights management
- Based on Storage Management SW
- **Archived products sub-setting**

ESRIN AMS configuration



AMS architecture



AMS components

AMASS

- COTS responsible for the data and volumes handling in the archive
- Implements the concept of Virtual File System
- Files organized in Volume Groups with one or more path roots associated
- Volumes metadata on disk database

AMS components

SatSTORE

- Provides the interface to AMASS (VFS interface) and Clients
- Interface with Clients via exchange of command & answer files (API available to be linked with client application)
- Interface with Clients via CORBA
- Products (Single File or Directory) transfer via FTP or NFS
- Transfers of products from/to Clients always initiated by AMS

AMS API and CORBA services

- Product Archiving
- Product Retrieval (with Sub-setting)
- Product Status
- Product Removal
- Product Move
- List of Products
- Request cancellation

AMS components

Archive Administration HMI

- JAVA client (Applet & Web Browser)
- HTTP server with CGI/Servlets interface to AMS internal services
- Dynamic (Ex. List of Requests) and Static (Ex. Statistics) information

AMS Clients configuration

- All clients are known and configured on the AMS in terms of host name, account and password to be used for FTP transfer
- Access rights granted per client and per dataset
- Files in the AMASS VFS stored read-only

AMS Products Identification

- Products are uniquely identified by a Rule of n keys

RULE_NAME : NOAA

KEY1 : Mission Identifier

KEY2 : Mission number

KEY3 : Sensor

KEY4 : Product Type

KEY5 : Acquisition Start Date Year

KEY6 : Acquisition Start Date Month

KEY7 : Acquisition Start Date Day

KEY8 : Acquisition Start Date Hours

KEY9 : Acquisition Start Date Minutes

KEY10 : Acquisition Start Date Seconds

KEY11 : Number of Lines

KEY12 : Orbit Number

KEY13 : Ascending/Descending Flag

KEY14 : Acquisition Station

KEY15 : Origin Product Name

archive_product_path : /[1]_[2]/[3]/[4]/[5]/[6]/

archive_product_name : [15]_[5]_[6]_[7]_[8]_[9]_[10]_[11]_[12]_[13]_[14].[4]

client_product_name : [5][6][7][8][9][14][1][2]

END_RULE_NAME

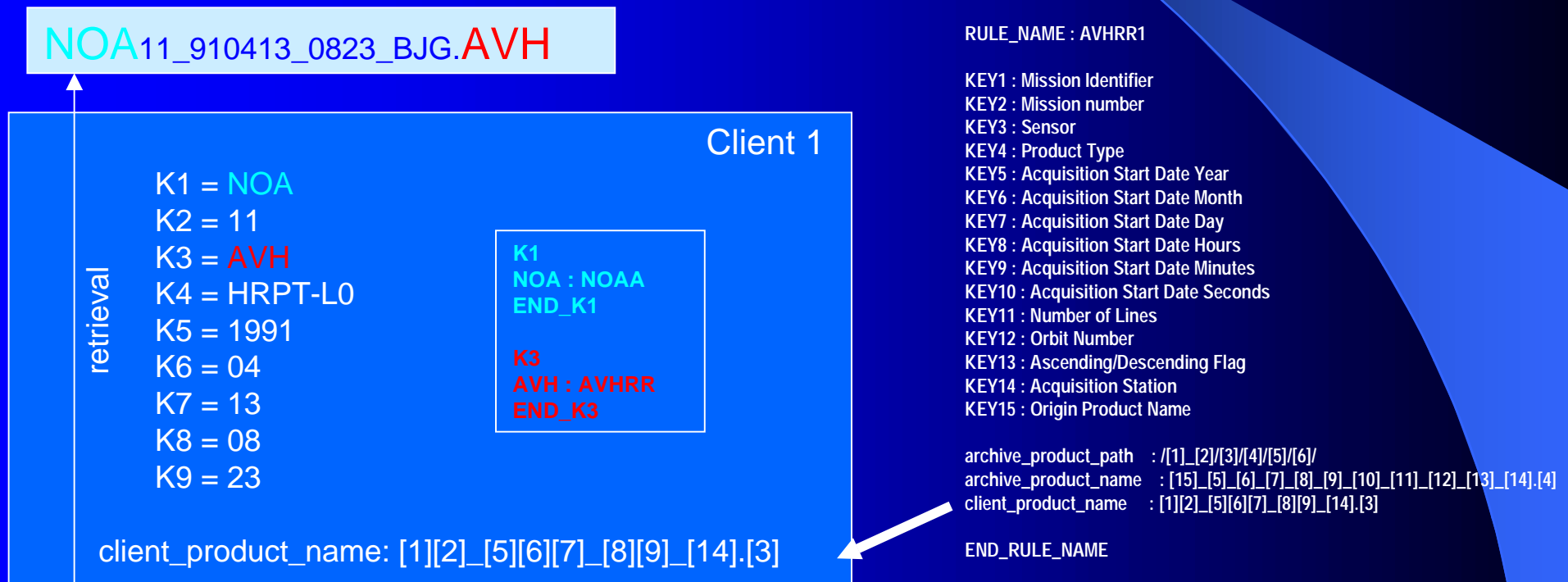
Product archiving example:

Product Path: /archive/NOAA_11/AVHRR/HRPT-L0/1991/04

Product Name: 9104130823BJGN1_1991_04_13_08_23_44_2378_23487_A_BJG.HRPT-L0

Enhanced Services for Clients

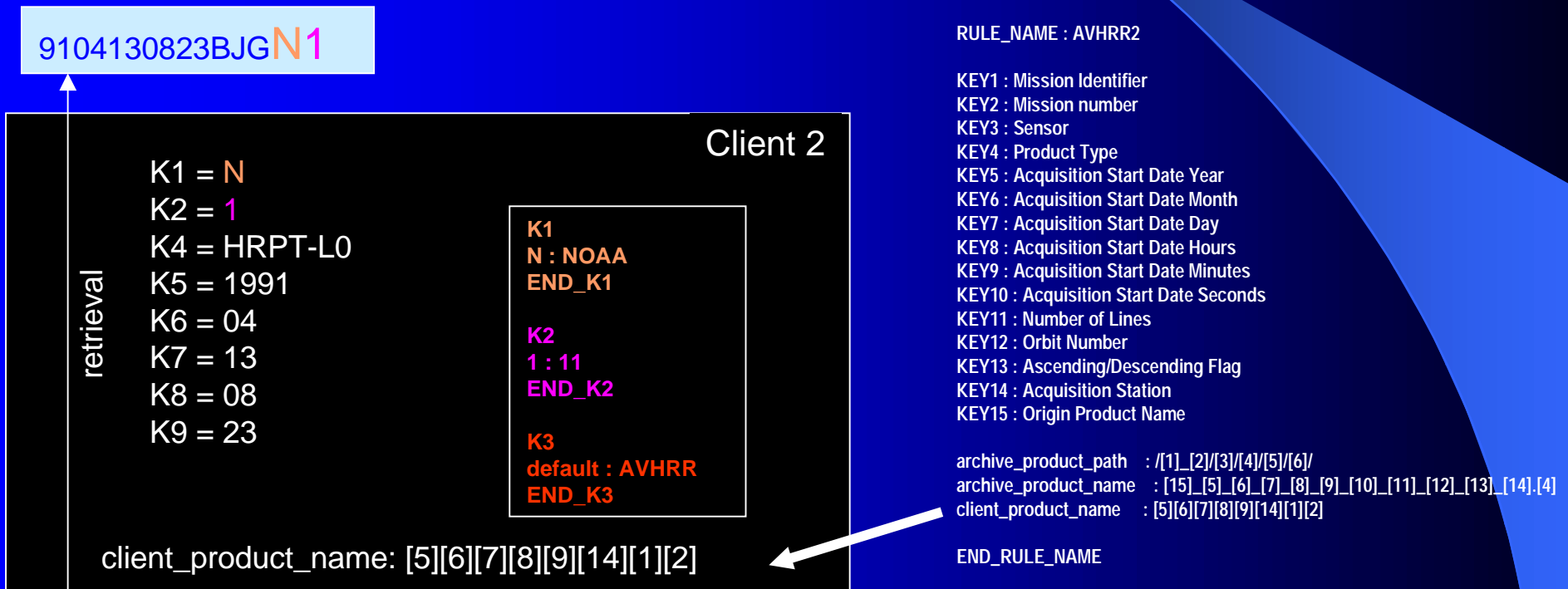
- Configuration files with Keys translation rules



/archive/NOAA_11/AVHRR/HRPT-L0/1991/04/9104130823BJGN1_1991_04_13_08_23_44_2378_23487_A_BJG.HRPT-L0

Enhanced Services for Clients

- Configuration files with Keys translation rules



/archive/NOAA_11/AVHRR/HRPT-L0/1991/04/9104130823BJGN1_1991_04_13_08_23_44_2378_23487_A_BJG.HRPT-L0

AMS components

Product Formatting & Sub-Setting (FSS)

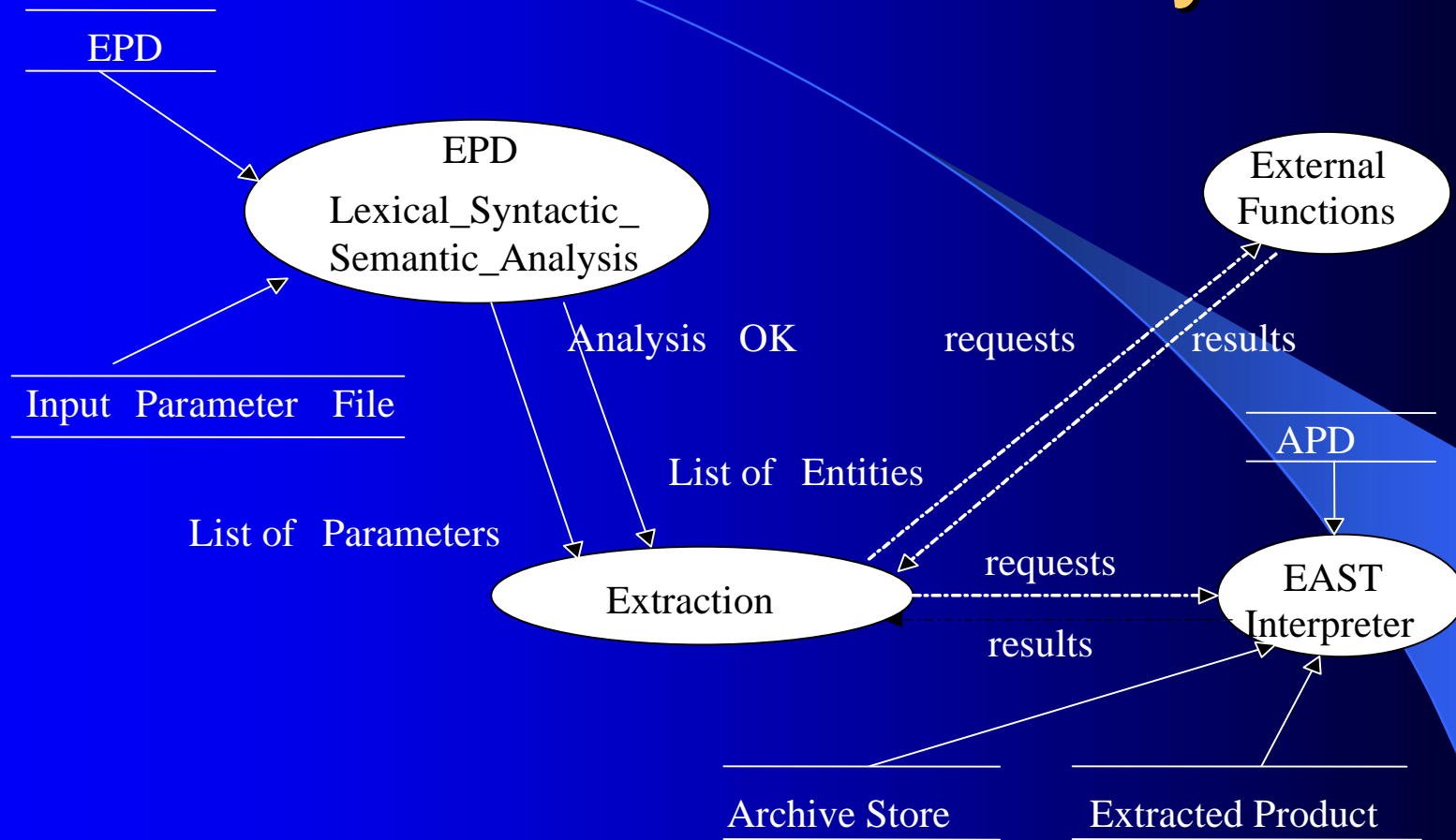
- Performs the extraction of a portion of product by specifying start and end parameters (UTC time, SBT, line, etc.)
- Needs description of products: Archive Product Description (APD)
- Needs definition of sub-setting algorithm: Extraction Process Definition (EPD)

AMS components

Product Formatting & Sub-Setting (FSS)

- APD: ASCII file based on the standard (CCSDS/ISO) data description language EAST
- EPD: ASCII file based on the standard (CCSDS) language PVL (Parameter Value Language)

FSS Extraction Facility



The subprogram 'EPD_Lexical_Syntactic_Semantic_Analysis' is responsible for the analysis of the EPD and the Input_Parameter_File and the storage of the different entities defined in the EPD.

The subprogram 'Extraction' is responsible for the application of the selected kind of extraction upon the selected data files by using the entity previously analysed and extracted. It extracts parameters on request or generates an output data file, called the Extracted Product, which can be used for further processing.

Lexical, Syntactic & Semantic Analysis

- ‘Lexical_Analysis’ is responsible for the lexical analysis of the EPD and provides the words read to the ‘Syntactic_Analysis’ upon its request
- ‘Syntactic_Analysis’ is responsible for the syntactic analysis of the EPD and analyses the words provided by the ‘Lexical_Analysis’ upon its request, in order to identify rules
- ‘Semantic_Analysis’ is responsible for the semantic analysis of the EPD and verifies and creates the entities structures as well as the List of Parameters. It prepares all necessary information for the extraction phase. It is responsible for the definition of the input files and descriptor files paths. It then associates the input file keys to values, manage them, and uses them to build the descriptor files, APD or output files names

Keywords and Operators

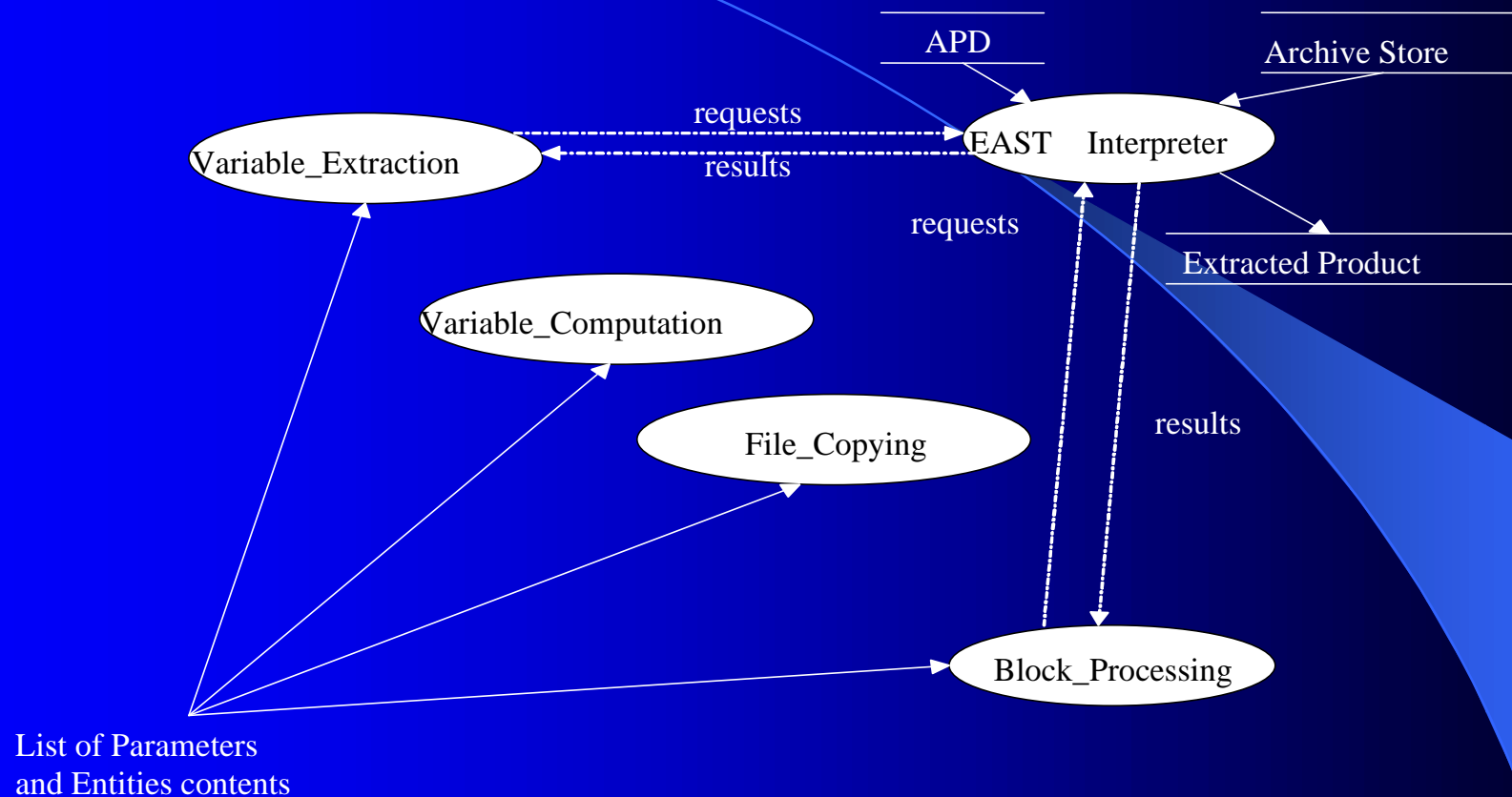
List of the possible keywords:

BEGIN_GROUP, END_GROUP, VARIABLES_EXTRACTION, COMMENTS,
INPUT_FILE_NUMBER, EPD_DESCRIPTION, EPD_NAME, OBJECTIVES,
INPUT_PARAMETERS_DESCRIPTION, PARAMETERS_FILE_PATH,
INPUT_FILES_WITH_EAST_DESCRIPTION,
INPUT_FILES_WITHOUT_EAST_DESCRIPTION, OUTPUT_FILES,
INPUT_PARAMETERS, PARAMETER_EXTRACTION_DEFINITION,
PARAMETER_NAME, PARAMETER_PATH, PARAMETER_TYPE,
PARAMETER_CONDITIONAL_EXTRACTION_DEFINITION, CONDITION,
VARIABLE_COMPUTATION, INTEGER, REAL, PARAMETER_VALUE,
RESULT_LOCATION, TEMPORARY_MEMORY_AREA, STOP_ADDRESS,
START_ADDRESS, MEMORY, BLOCK_LOCATION,
CONDITION_PARAMETER_TYPE, OUTPUT_FILE_NUMBER, FIELD_NAME,
RECORDS_EXTRACTION.

List of possible operators:

“==”, “=”, “,”, “<”, “>”, “<=”, “>=”, “+”, “-“, “*”, “/”, “(“, “)”, “.”

Data Extraction



The subprogram 'Variable_Extraction' is responsible for the extraction of the different variables that have to be extracted from a particular Data File and for the storage of their associated extracted values (such as for example SWATH_SIZE in the User Header File).

The subprogram 'Variable_Computation' is responsible for the computation of the different variables that have to be computed according to formulas.

The subprogram 'File_Copying' is responsible for the file copy from the input product directory to the output directory.

The subprogram 'Block_Processing' is responsible for the extraction of the blocks corresponding to the defined criterion (a date, a block number, a line number, ...).

EPD Example

```
BEGIN_GROUP = EPD_DESCRIPTION;  
    EPD_NAME = "LANDSAT_TM_RETRIEVAL";  
    OBJECTIVES = "Test";  
END_GROUP = EPD_DESCRIPTION;  
  
BEGIN_GROUP = CRITERIA_DESCRIPTION;  
    PARAM1 = "TIME1";  
    PARAM2 = "TIME2";  
END_GROUP = CRITERIA_DESCRIPTION;  
.....
```

EPD Example

.....

```
BEGIN_GROUP = VARIABLES_EXTRACTION;
```

```
FILENAME = "/ARCHIVE/[1]/[2]/[3]/DESC_*/*.HEA";
```

```
APD_FILENAME = "$SVA_APD_FILE_SPACE/user_header.eas";
```

```
BEGIN_GROUP = PARAMETER_EXTRACTION_DEFINITION;
```

```
    PARAMETER_NAME = "SWATH_SIZE";
```

```
    PARAMETER_PATH = "POINTERS_TO_TAPE_DATA.SWATH_SIZE";
```

```
    PARAMETER_TYPE = "INTEGER";
```

```
END_GROUP = PARAMETER_EXTRACTION_DEFINITION;
```

```
BEGIN_GROUP = PARAMETER_EXTRACTION_DEFINITION;
```

```
    PARAMETER_NAME = "SWATH_PER_BLOCK";
```

```
    PARAMETER_PATH = "POINTERS_TO_TAPE_DATA.SWATH_PER_BLOCK";
```

```
    PARAMETER_TYPE = "INTEGER";
```

```
END_GROUP = PARAMETER_EXTRACTION_DEFINITION;
```

```
END_GROUP = VARIABLES_EXTRACTION;
```

.....

EPD Example

.....

```
BEGIN_GROUP = VARIABLES_EXTRACTION;  
  FILENAME = "/ARCHIVE/[1]/[2]/[3]/DESC_*/*.BDA";  
  APD_FILENAME = "$SVA_APD_FILE_SPACE/block_adr_descriptor.eas";  
  BEGIN_GROUP = PARAMETER_CONDITIONAL_EXTRACTION_DEFINITION;  
    PARAMETER_NAME = "BLOCK_NUMBER_BEGIN";  
    PARAMETER_PATH = "VIDEO_START_ADDR";  
    PARAMETER_TYPE = "INTEGER";  
    CONDITION = "(SATELLITE_TIME < PARAM1)";  
    CONDITION_PARAMETER_TYPE = "TIME_IN_MILLISECOND";  
  END_GROUP = PARAMETER_CONDITIONAL_EXTRACTION_DEFINITION;  
  BEGIN_GROUP = PARAMETER_CONDITIONAL_EXTRACTION_DEFINITION;  
    PARAMETER_NAME = "BLOCK_NUMBER_END";  
    PARAMETER_PATH = "VIDEO_START_ADDR";  
    PARAMETER_TYPE = "INTEGER";  
    CONDITION = "(SATELLITE_TIME == PARAM2)";  
    CONDITION_PARAMETER_TYPE = "TIME_IN_MILLISECOND";  
  END_GROUP = PARAMETER_CONDITIONAL_EXTRACTION_DEFINITION;  
END_GROUP = VARIABLES_EXTRACTION;
```

.....

EPD Example

```
.....  
BEGIN_GROUP = VARIABLES_COMPUTATION;  
    PARAMETER_NAME = "BLOCK_LENGTH";  
    PARAMETER_VALUE = "(SWATH_PER_BLOCK * SWATH_SIZE)";  
    PARAMETER_TYPE = "INTEGER";  
END_GROUP = VARIABLES_COMPUTATION;  
  
BEGIN_GROUP = VARIABLES_COMPUTATION;  
    PARAMETER_NAME = "NB_BLOCKS";  
    PARAMETER_VALUE = "(BLOCK_NUMBER_END - BLOCK_NUMBER_BEGIN)";  
    PARAMETER_TYPE = "INTEGER";  
END_GROUP = VARIABLES_COMPUTATION;  
.....
```

EPD Example

.....

```
BEGIN_GROUP = VARIABLES_COMPUTATION;
```

```
    PARAMETER_NAME = "SIZE_BLOCKS";
```

```
    PARAMETER_VALUE = "(NB_BLOCKS * BLOCK_LENGTH)";
```

```
    PARAMETER_TYPE = "INTEGER";
```

```
END_GROUP = VARIABLES_COMPUTATION;
```

```
BEGIN_GROUP = VARIABLES_COMPUTATION;
```

```
    PARAMETER_NAME = "START_BYTE";
```

```
    PARAMETER_VALUE = "(BLOCK_NUMBER_BEGIN * BLOCK_LENGTH)";
```

```
    PARAMETER_TYPE = "LONG";
```

```
END_GROUP = VARIABLES_COMPUTATION;
```

```
BEGIN_GROUP = VARIABLES_COMPUTATION;
```

```
    PARAMETER_NAME = "STOP_BYTE";
```

```
    PARAMETER_VALUE = "(START_BYTE + SIZE_BLOCKS)";
```

```
    PARAMETER_TYPE = "LONG";
```

```
END_GROUP = VARIABLES_COMPUTATION;
```

.....

EPD Example

.....

```
BEGIN_GROUP = RECORDS_EXTRACTION;
```

COMMENTS= "This statements group allows to extract records from a set of blocks located in a file. These blocks are repetition of records. All blocks are analysed on a record by record basis, and those verifying the conditions are extracted;

```
FILENAME = "/TEST/SatSTORE_V4_1_VALID/[1]/[2]/[3]/[4]_[5]/*.IMA";
```

```
START_ADDRESS = "START_BYTE";
```

```
STOP_ADDRESS = "STOP_BYTE";
```

```
APD_FILENAME = "$SVA_APD_FILE_SPACE/sensor_TM_data.eas";
```

```
BLOCK_DUMPING = "FALSE";
```

```
LENGTH = "BLOCK_LENGTH";
```

```
CONDITION = "(LANDSAT_TM_MAJOR_FRAME.SATELLITE_AUXILIARY_DATA.DATE  
>= PARAM1) AND (LANDSAT_TM_MAJOR_FRAME.SATELLITE_AUXILIARY_DATA.DATE <=  
PARAM2) ";
```

```
CONDITION_PARAMETER_TYPE = "TIME_LANDSAT";
```

```
OUTPUT_FILE = "TEST.IMA";
```

```
END_GROUP = RECORDS_EXTRACTION;
```

Conclusions

- The use of EAST in the AMS was introduced to permit the subsetting of large data files in the AMS
- The requirement was fully met and any AMS user is able today to have a special subsetting configured
- EAST has been very efficient to achieve the goal in an operational system
- The performance of the FSS are satisfactory even in an operational use of the feature
- Special characteristics of the format of the archived product can be exploited to increase the subsetting speed