z/OS Academy

Introduction to Data Gatherer REST Services

Johannes Doerk

z/OS Data Gatherer
Software Developer

 © 2022 IBM Corporation

Disclaimer

Statements regarding IBM future direction and intent are subject to change or withdrawal and represent goals and objectives only

Agenda

Motivation

Data Gatherer REST services

In-Place Documentation

Endpoint Pattern

Schema vs. Object Model

Summary

 © 2022 IBM Corporation

Motivation

Motivation - SMF Data Consumers

- Yesterday:
  - z/OS systems programmer
    - Solid z/OS background
    - Solid native interface skills
    - The whole time of the Universe to access, process, and interpret the data
- Tomorrow additionally:
  - ML/AI Analyst & Data Scientist / Engineer
    - Solid math, statistics skills
    - Solid modern Programming Languages / Frameworks skills
    - None up to a little z/OS background
    - No time at all

 © 2022 IBM Corporation

Motivation - Proof

- This document assumes that the reader has extensive experience with z/OS, is familiar with its basic concepts, and can code JCL statements to run programs or cataloged procedures, can code in assembler language, and can read assembler, loader, and linkage editor output © MVS System Management Facilities (SMF)

- Data Science:
  - Python Foundations - Libraries: Pandas, NumPy, Arrays and Matrix handling, Visualization, Exploratory Data Analysis (EDA)
  - Statistics Foundations: Basic/Descriptive Statistics, Distributions (Binomial, Poisson, etc.), Bayes, Inferential Statistics
  - ….

    – © Massachusetts Institute of Technology, Applied Data Science Program

Motivation – Current access

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW
****** ***************************************************** Top of Data **********
000001 1z/OS V2R5 Data Gatherer ERBMFSCN Version 9 (30 Apr 2012) - SCAN SMF dataset
000002
000003  SMF dataset characteristics:
000004  RECFM    : VBS
000005  LRECL    : 32767
000006  BLKSIZE  : 32760
000007  DATASETS : 1
000008  DSNAME(S):
000009  DATE/TIME: 2022 February 23     10:59:05.732
000010
000011
000012 1Rec-Num Type    RecLn SMFDate  SMFTime  RMFDate  RMFTime  Int-Len   SMFId/Ver
000013  ------- ------- ----- -------- -------- -------- -------- --------- ---------
000014        1 002        18 2022.054 10:58:42                              SYSF
000015        2 070.001 32396 2022.054 03:59:31 2022.054 03:44:31 15:00:003 SYSF 797-
000016        3 070.001 20508 2022.054 03:59:31 2022.054 03:44:31 15:00:003 SYSF 797-
000017        4 070.001 32396 2022.054 04:14:31 2022.054 03:59:31 14:59:996 SYSF 797-
000018        5 070.001 20508 2022.054 04:14:31 2022.054 03:59:31 14:59:996 SYSF 797-
000019        6 070.001 32396 2022.054 04:29:31 2022.054 04:14:31 15:00:002 SYSF 797-
000020        7 070.001 20508 2022.054 04:29:31 2022.054 04:14:31 15:00:002 SYSF 797-
```

Motivation – Current access

```
001 Record Number 2: SMF Record Type 70(1) - RMF CPU Activity
002 ========================================================
003
004 -> SMF record header
005 ====================
006
007    SMF record length      : 32396
008    SMF segment descriptor : '0000'X
009    SMF system indicator   : '11011111'B
010    SMF record type        : 70
011    SMF record time        : 03:59:31
012    SMF record date        : 22.054
013    SMF system id          :
014    SMF subsystem id       : RMF
015    SMF record subtype     : 1
016
017 -> RMF header extension
018 ========================
019
020    Number of triplets     : 9
021
022    Section  1 offset      : '00000064'X
023    Section  1 length      : '009C'X
024    Section  1 number      : 1
025
026    Section  2 offset      : '00000100'X
027    Section  2 length      : '0158'X
```

```
00058 -> RMF Product Section (1)
00059 ==========================
00060
00061    #1:    +0000:  797FD9D4 C6404040 40400034 431F0122  *'"RMF       C   *
00062           +0010:  054F1500 003F0000 00000384 00001001  * |          d    *
00063           +0020:  40404040 0001000F E9E5F0F2 F0F5F0F0  *         ZV020500*
00064           +0030:  03770C8E DB1B6F20 7CC00000 00000D69  * Ï l! ? @{     Z*
00065           +0040:  3A400000 00000068 00340001 03840000  *       Y     d  *
00066           +0050:  DB1B6F20 7CC00000 E2E8E2C4 D7D3C5E7  *! ? @{           *
00067           +0060:  E2E8E2C6 40404040 00020001 00000010  *SYSF             *
00068           +0070:  00040009 00000000 00010001 00010001  *                *
00069           +0080:  00050001 00010001 003C0001 01D90001  *             R  *
00070           +0090:  00060001 00000000 00060001           *                *
00071
00072 -> CPU Control Section (1)
00073 ==========================
00074
00075    #1:    +0000:  39060069 18990000 F7F5F940 40404040  *   Z r  759     *
00076           +0010:  40404040 40404040 00B00000 000001C6  *         g     F*
```

Motivation – Current access

Record mapping

Header/Self-defining Section

This section contains the common SMF record headers fields and the triplet fields (offset/length/number), if applicable, that locate the other sections on the record.

Offsets		Name	Length	Format	Description
0	0	SMF70LEN	2	binary	Record length. This field and the next field (total of four bytes) form the RDW (record descriptor word). See “Standard and Extended SMF record headers” on page 162 for a detailed description.
2	2	SMF70SEG	2	binary	Segment descriptor (see record length field).
4	4	SMF70FLG	1	binary	System indicator: **Bit** **Meaning when set** **0** New record format **1** Subtypes used **2** Reserved. **3-6** Version indicators* **7** System is running in PR/SM mode. *See “Standard and Extended SMF record headers” on page 162 for a detailed description.

Motivation – Current access

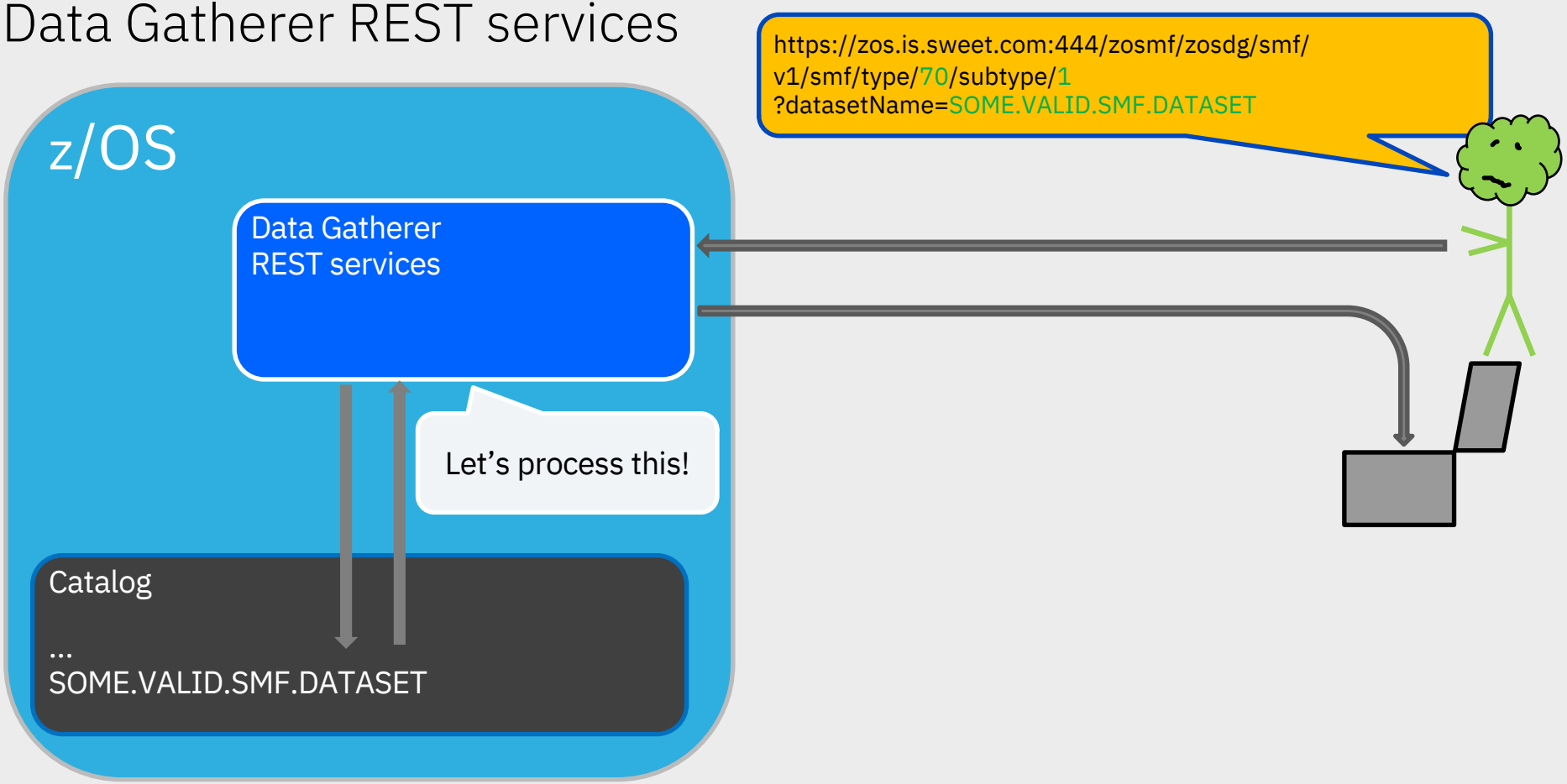
```
SMF&R.HDR DSECT
SMF&R.HDC DS    CL36     COMPLETE HEADER+PRODUCT SECTION TRIPL
          ORG   SMF&R.HDC
SMF&R.LEN DS    BL2      RECORD LENGTH
SMF&R.SEG DS    BL2      SEGMENT DESCRIPTOR
SMF&R.FLG DS    BL1      HEADER FLAG BYTE
SMF&R.RRF EQU   X'80'    NEW RECORD FORMAT
SMF&R.SUT EQU   X'40'    SUBTYPES USED
SMF&R.V4  EQU   X'10'    MVS/ESA Version 4
SMF&R.ESA EQU   X'08'    MVS/ESA VERSION
SMF&R.VXA EQU   X'04'    MVS/XA VERSION
SMF&R.OS  EQU   X'02'    OPERATING SYSTEM IS OS/VS2
SMF&R.BFY EQU   X'01'    System running in PR/SM mode
SMF&R.RTY DS    BL1      RECORD TYPE
SMF&R.TME DS    BL4      TOD RECORD WRITTEN
SMF&R.DTE DS    PL4      DATE RECORD WRITTEN
SMF&R.SID DS    CL4      SYSTEM ID FROM INSTALLATION
SMF&R.SSI DS    CL4      SUBSYSTEM ID (RMF)
SMF&R.STY DS    BL2      SUBTYPE
SMF&R.TRN DS    BL2      NUMBER OF TRIPLETS IN THIS RECORD
          DS    BL2      RESERVED
SMF&R.PRS DS    BL4      OFFSET TO RMF PRODUCT SECTION
SMF&R.PRL DS    BL2      LENGTH OF RMF PRODUCT SECTION
SMF&R.PRN DS    BL2      NUMBER OF RMF PRODUCT SECTIONS
```

Goal

- Describe all the possible Data Models and their access methods:
  - Without proprietary specifications
  - Relying on industry standards
  - Staying outside any programming language(s) skillset
  - Being scalable & adjustable

Data Gatherer REST services

Data Gatherer REST services



Result: JSON

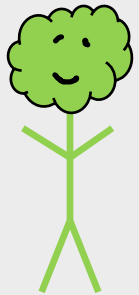
Indexing

Formatting

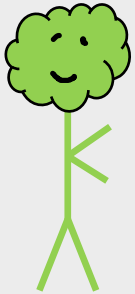
```
0:                                {…}
1:                                {…}
2:
  SMF70LEN:                       1520
  SMF70SEG:                       0
  SMF70FLG:                       "11011110"
  SMF70RRF:                       true
  SMF70SUT:                       true
  SMF70V4:                        true
  SMF70ESA:                       true
  SMF70VXA:                       true
  SMF70OS:                        true
  SMF70BFY:                       false
  SMF70RTY:                       70
  SMF70TME:                       "14:59:00.01"
  SMF70DTE:                       "2017-06-21"
  SMF70SID:
  SMF70SSI:                       "RMF "
  SMF70STY:                       1
  SMF70TRN:                       9
  SMF70PRS:                       100
  SMF70PRL:                       104
  SMF70PRN:                       1
  SMF70CCS:                       204
  SMF70CCL:                       344
  SMF70CCN:                       1
  SMF70CPS:                       548
  SMF70CPL:                       92
  SMF70CPN:                       2
  SMF70ASS:                       732
  SMF70ASL:                       788
  SMF70ASN:                       1
  SMF70BCS:                       1520
  SMF70BCL:                       84
  SMF70BCN:                       0
  SMF70BVS:                       1520
```

```
  SMF70TNS:                       0
  SMF70TNL:                       0
  SMF70TNN:                       0
▸ smf70Subtype1ProductSection:    {…}
▸ smf70Subtype1AsidArea:          {…}
▾ smf70Subtype1CpuControl:
    SMF70MOD:                     "2964"
    SMF70VER:                     255
    SMF70BNP:                     0
    SMF70INB:                     "00000000"
    SMF70DIF:                     false
    SMF70NPC:                     false
    SMF70TSC:                     false
    SMF70PHY:                     false
    SMF70DGE:                     false
    SMF70VMG:                     false
    SMF70STF:                     "10011000"
    SMF70STS:                     true
    SMF70ADC:                     false
    SMF70WUC:                     false
    SMF70RCU:                     true
    SMF70HWV:                     true
    SMF70PTC:                     false
    SMF70PLC:                     false
    SMF70GAV:                     false
    SMF70GTS:                     0
    SMF70MDL:                     "757             "
    SMF70DSA:                     12
    SMF70IFA:                     0
    SMF70CPA:                     506
    SMF70WLA:                     1024
    SMF70LAC:                     2
    SMF70HWM:                     "N63             "
    SMF70SUP:                     0
    SMF70GJT:                     "02:00:00"
    SMF70POM:                     "02  "
```

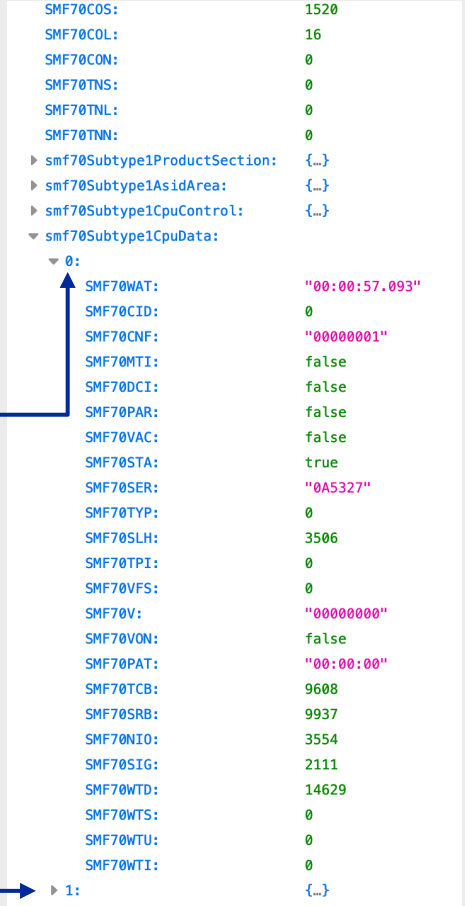
Hierarchy and 1-to-1 relationships

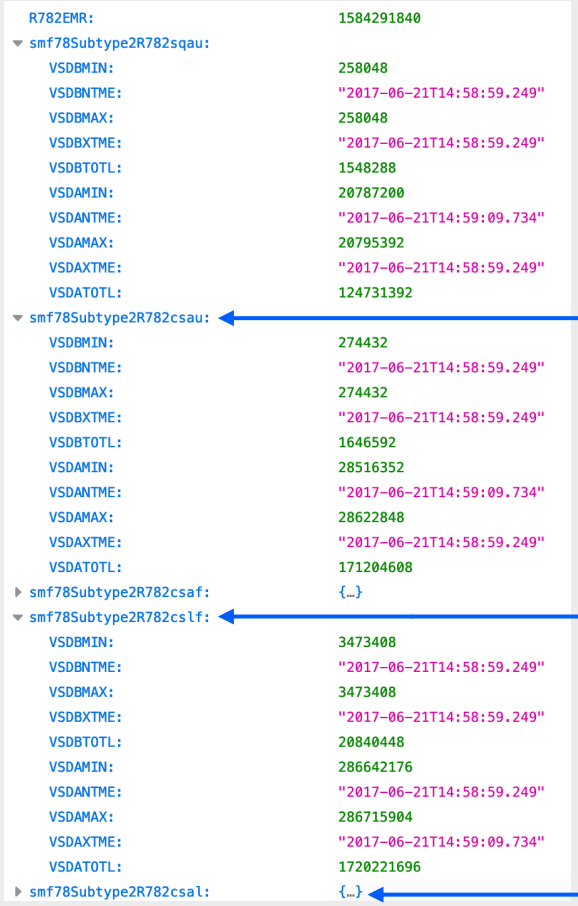


Result: JSON



1-to-many relationships





78-2 formatted fields

What is a (RESTful-)API?

- Application Programming Interface (API):
  - a set of rules that define how applications or devices can connect to and communicate with each other

- REpresentational State Transfer (REST) API:
  - an API that conforms to the design principles of the REST architectural style
  - has a goal to increase performance, scalability, simplicity, modifiability, visibility, portability, and reliability
  - provides:
    - simplicity of a uniform interface
    - modifiability of components to meet changing needs (even while the application is running)
  - …

What is the OpenAPI Specification?

- defines a standard, language-agnostic interface to RESTful APIs
- eases the discovery and understanding of the capabilities of the service
  - no source code access is required, neither documentation, nor network traffic inspection
- can then be used by:
  - documentation generation tools to display the API
  - code generation tools to generate servers and clients in various programming languages, testing tools, and many other use cases

© swagger.io

An OASv3 OpenAPI Document

- OpenAPI Document
  - document that defines or describes an API. An OpenAPI definition uses and conforms to the OAS
- Provides in-place documentation about:
  - all the possible HTTP requests, including:
    - request path, its' variables, query parameters, ingress headers etc.
  - all the possible responses, including:
    - statuses and their meanings
    - possible response models, including
      - structures, elements, parameters
      - parent-child hierarchies, as well as 1-to-1, 1-to-many relationships

 © 2022 IBM Corporation

Example: Request-Response contract

```
/v1/smf/type/70/subtype/1:
  get:
    tags:
      0:                                "SMF 70"
    summary:                            "CPU, PR/SM, and ICF Activity"
    description:                        "This end-point retrieves SMF 70 subtype 1 data from the specific SMF record using the dataset provided"

    parameters:                         […]
    responses:
      200:
        description:                    "No error occurred and the response contains all the data requested"
        content:
          application/json:
            schema:
              $ref:                     "#/components/schemas/IndexedMapOfSmf70Subtype1View"
      404:
        description:                    "The specified resource cannot be found"
        content:
          application/json:
            schema:
              $ref:                     "#/components/schemas/Error"
      500:
        description:                    "Internal Server Error"
        content:
          application/json:
            schema:
              $ref:                     "#/components/schemas/Error"
```

Application of those concepts to our use case?

- Every supported SMF (sub-)type:
  - is an OpenAPI model
  - is accessible via its own end-point on demand
  - consists of the full Data Model Composition with parent-child hierarchies, aka
    - Header ->
      - (1:1) Product section
      - (1:1) ASID_AREA
      - (1:1) CPU_CONTROL
      - (1:n) CPU_DATA
      - …

 © 2022 IBM Corporation

Infrastructure

- Java application
  - zIIPs benefits are available
  - uses own z/OS Liberty server instance provided and pre-configured by z/OSMF
    - does NOT depend on z/OSMF Primary server instance
    - does NOT require z/OSMF Primary server instance during runtime
    - may be configured separately (i.e., memory limits, JVM options etc.)
      - the memory required is correlated with the amount of data to be processed and directly influences the performance
    - additional application-specific configuration is available

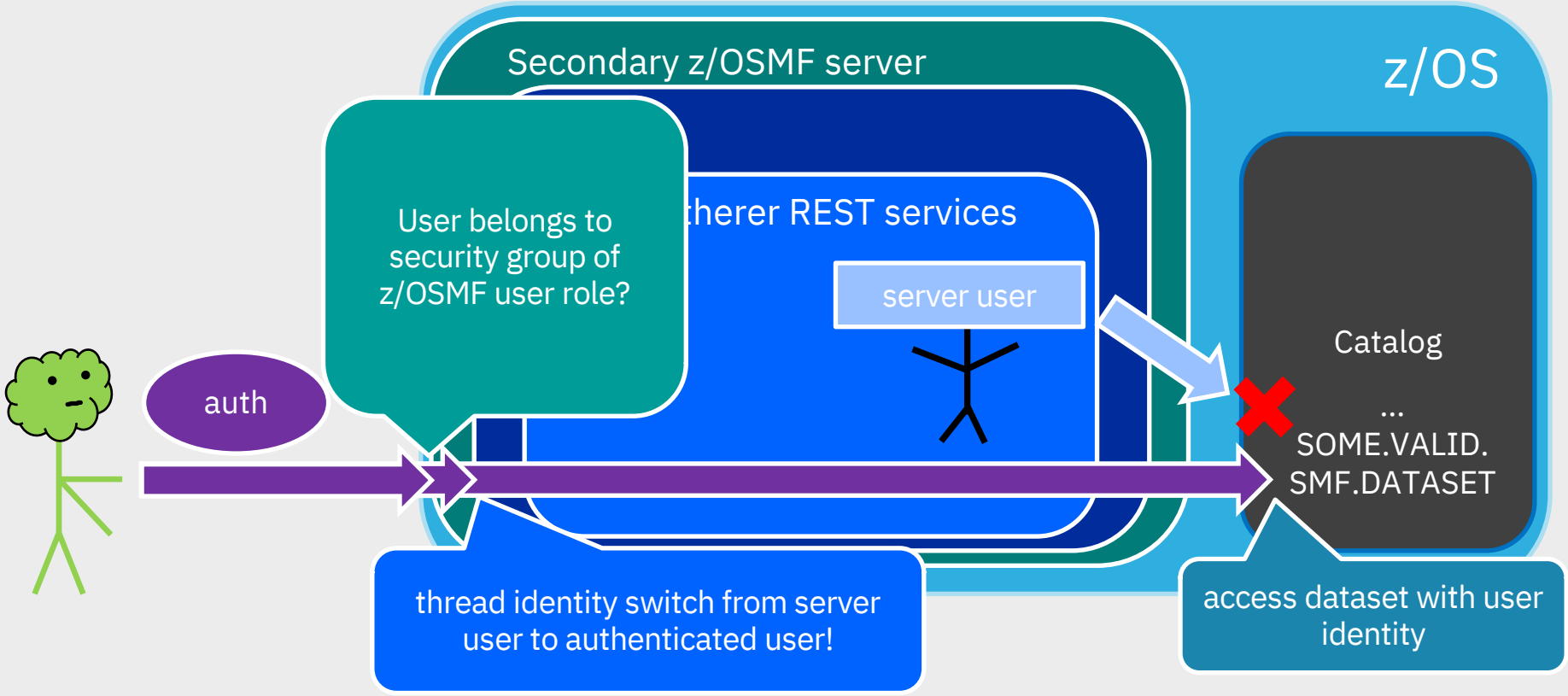


Authentication

- server-managed (provided by Liberty):
  - basic
    - header-based
    - enabled by default for OpenAPI Schema
  - token-based
    - LTPA2
      - cookie-based
      - enabled by default for OpenAPI Schema
    - JWT
      - cookie-based
      - disabled by default for OpenAPI Schema

 © 2022 IBM Corporation

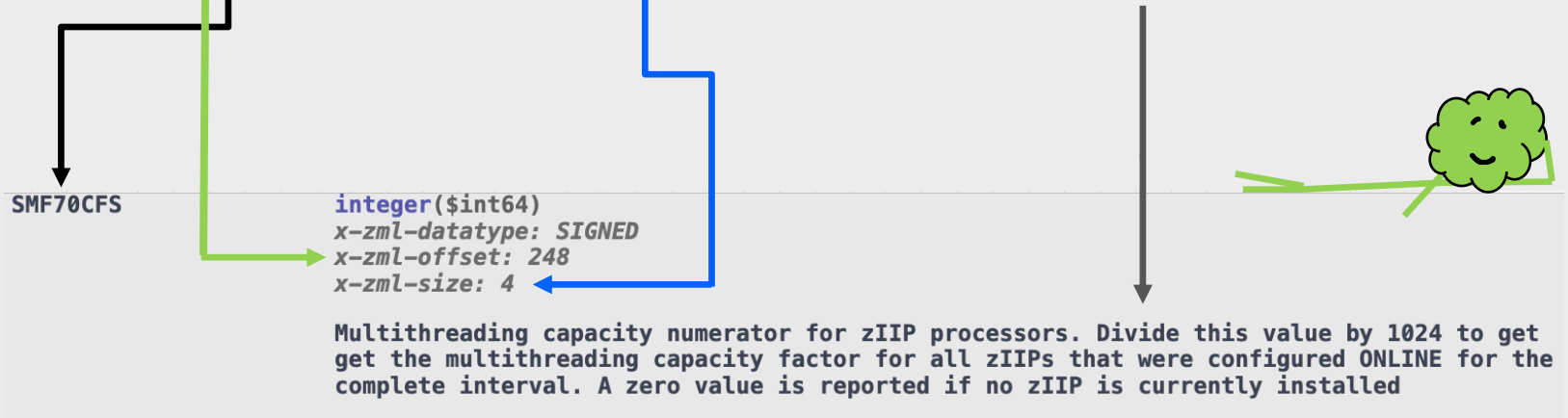
Request flow security



In-Place Documentation

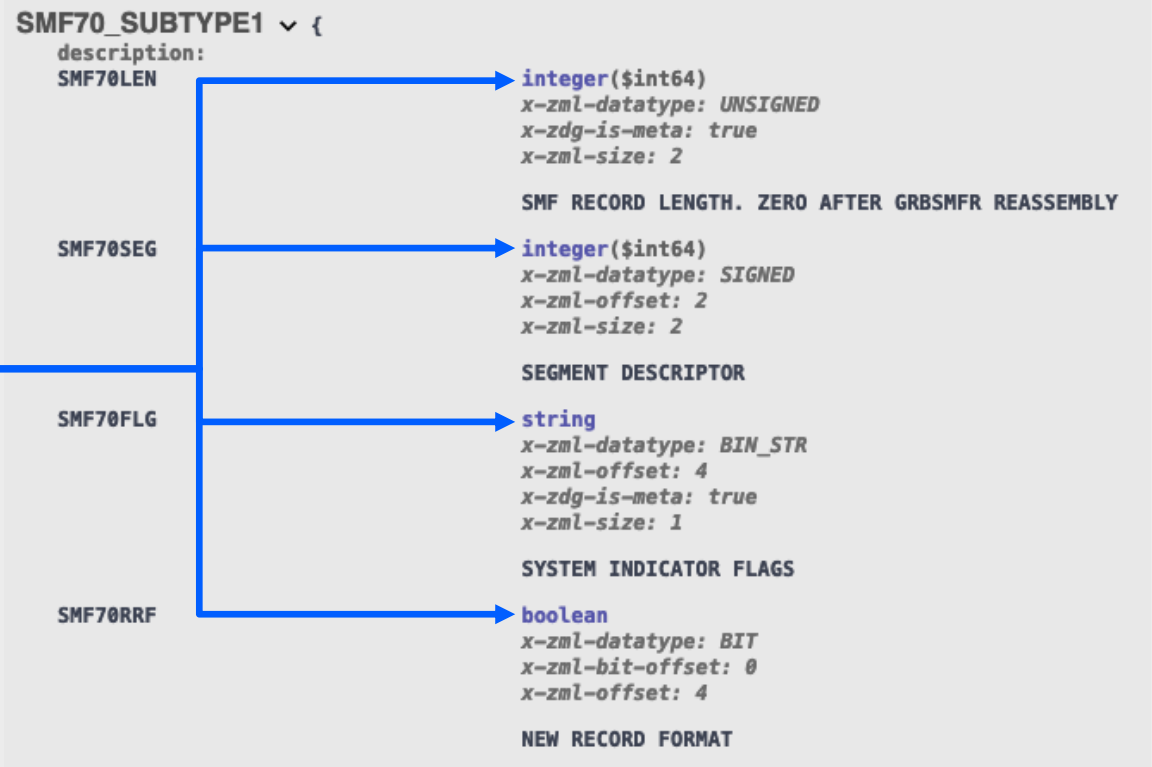
In-Place Documentation (field-level description)

Offsets		Name	Length	Format	Description
248	F8	SMF70CFS	4	binary	Multithreading capacity numerator for zIIP. Divide this value by 1024 to get the multithreading capacity factor for all zIIPs that were configured ONLINE for the complete interval. A zero value is reported if no zIIP is currently installed.

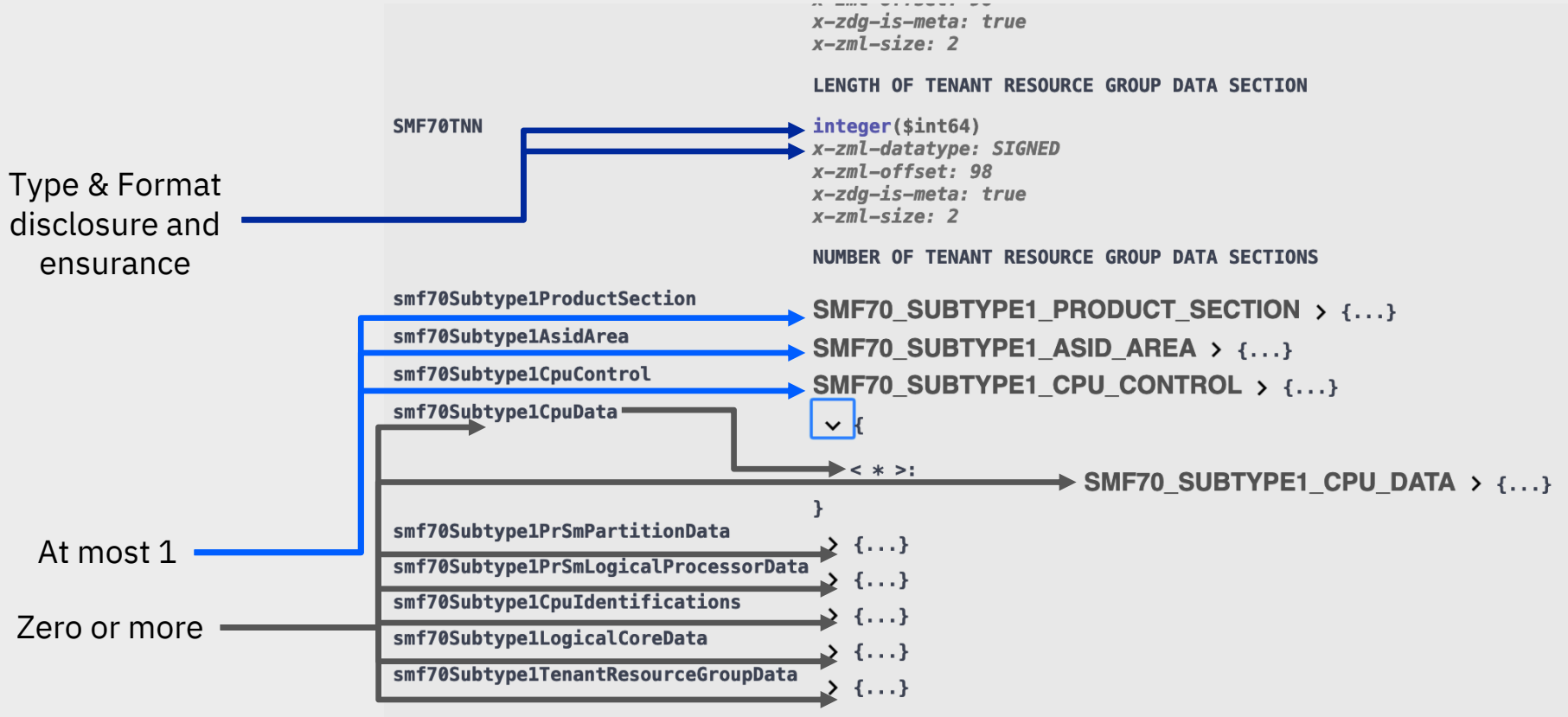


In-Place Documentation

Type & Format disclosure and ensurance

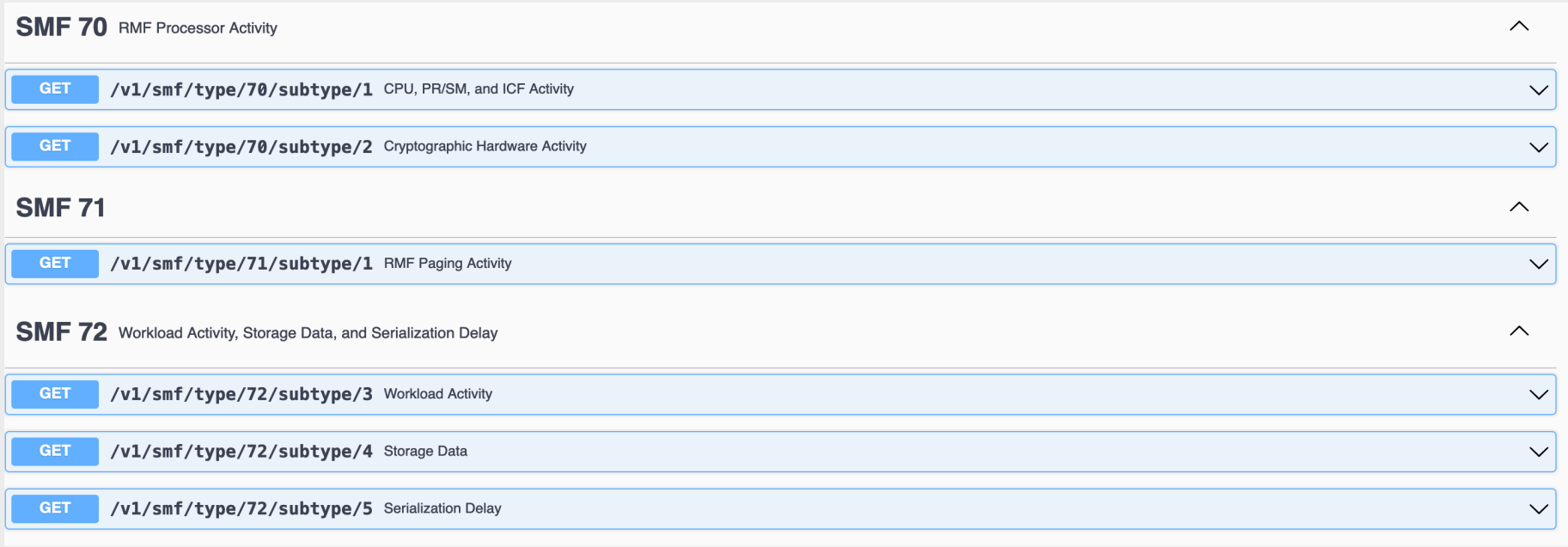


Parent-child hierarchies 1-to... relationships



Endpoint Pattern

Specific endpoint per (sub-)type



... up to SMF 79

SMF 99 subtypes 1, 2, 6, 12 and 14

SMF 113 subtypes 1 and 2

Endpoint pattern (required)

{host}:{port}/{context-path}/v1/smf/type/{#type}/subtype/{#subtype}?datasetName={smf.dataset.name}&{opt}

Element	Description	Example(s)
Host	Machine’s network identifier	ibm.example.com
Port	The port assigned to the Web Application Server the application is deployed at	444
Context-path	The the prefix of a URL-path	zosmf/zosdg/smf/
#type	The SMF type the application shall process	70 / 71 / 78
#subtype	The subtype of the SMF type the application shall process	1 / 2 / 11
smf.dataset.name	The name of a dataset to process	SOME.VALID.SMF.DATASET

 © 2022 IBM Corporation

Endpoint pattern (required) - Example

https://ibm.example.com:444/zosmf/zosdg/smf/v1/smf/type/70/subtype/1

?datasetName=SOME.VALID.SMF.DATASET

Endpoint pattern (common optional)

Element	Description	Example(s)
systemName	Four-character SMF system ID that identifies the z/OS system from which the data is gathered	ASYS
startTime	Data gathering begin time in ISO8601 format	2022-03-28T10:00:00 2022-03-28T10:00 2022-03-28
endTime	Data gathering end time in ISO8601 format	2022-03-28T11:00:00 2022-03-28T11:00 2022-03-28

 © 2022 IBM Corporation

Endpoint pattern (common optional) - Example

https://ibm.example.com:444/zosmf/zosdg/smf/v1/smf/type/70/subtype/1

?datasetName=SOME.VALID.SMF.DATASET

&systemName=ASYS

&startTime=2022-03-28T10:00:00

&endTime=2022-03-28T11:00:00

Endpoint pattern (specific optional)

Element	Description	Example(s)
SMF70_SUBTYPE1	A set of the fields that are to be returned for SMF70_SUBTYPE1-structure	SMF70FLG / SMF70SSI & ~40 other fields
SMF70_SUBTYPE1_CPU_DATA	A set of the fields that are to be returned for this structure	SMF70TYP / SMF70PAT & ~20 other fields
...		

 © 2022 IBM Corporation

Endpoint pattern (specific optional) - Example

https://ibm.example.com:444/zosmf/zosdg/smf/v1/smf/type/70/subtype/1

?datasetName=SOME.VALID.SMF.DATASET

&systemName=ASYS

&startTime=2022-03-28T10:00:00

&endTime=2022-03-28T11:00:00

&SMF70_SUBTYPE1=SMF70TME

&SMF70_SUBTYPE1_CPU_DATA=SMF70WAT

&SMF70_SUBTYPE1_CPU_DATA=SMF70CID

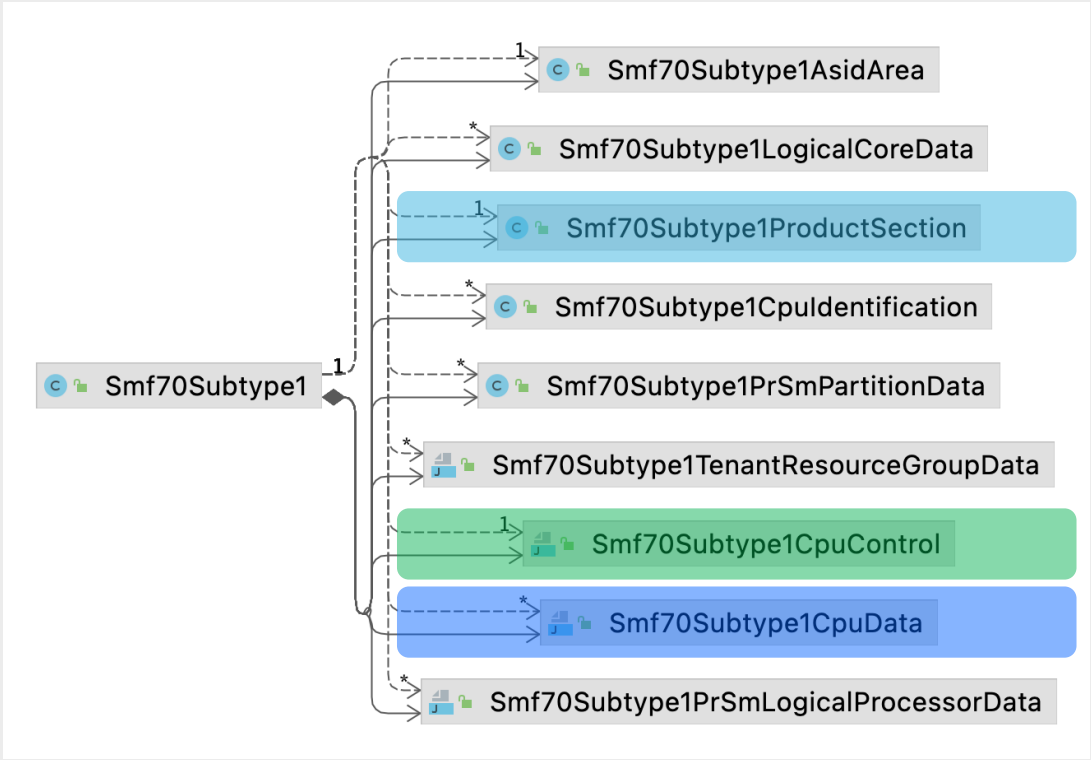
&SMF70_SUBTYPE1_CPU_IDENTIFICATION=*

Schema vs. Object Model

How does the data look like?

26	1A		2		Reserved.
28	1C	SMF70PRS	4	binary	Offset to RMF product section from the RDW.
32	20	SMF70PRL	2	binary	Length of RMF product section.
34	22	SMF70PRN	2	binary	Number of RMF product sections.
Individual header extension for subtype 1:					
36	24	SMF70CCS	4	binary	Offset to CPU control section from RDW.
40	28	SMF70CCL	2	binary	Length of CPU control section.
42	2A	SMF70CCN	2	binary	Number of CPU control section.
44	2C	SMF70CPS	4	binary	Offset to CPU data section from RDW.
48	30	SMF70CPL	2	binary	Length of CPU data section.
50	32	SMF70CPN	2	binary	Number of CPU data sections in this record.
52	34	SMF70ASS	4	binary	Offset to ASID Data Area section from RDW.
56	38	SMF70ASL	2	binary	Length of ASID Data Area section.
58	3A	SMF70ASN	2	binary	Number of ASID Data Area sections.
60	3C	SMF70BCS	4	binary	Offset to PR/SM Partition data section from RDW.
64	40	SMF70BCL	2	binary	Length of PR/SM Partition data section.
66	42	SMF70BCN	2	binary	Number of PR/SM Partition data sections.
68	44	SMF70BVS	4	binary	Offset to PR/SM Logical Processor data section from RDW.
72	48	SMF70BVL	2	binary	Length of PR/SM Logical Processor data section.
74	4A	SMF70BVN	2	binary	Number of PR/SM Logical Processor data sections.

SMF 70-1

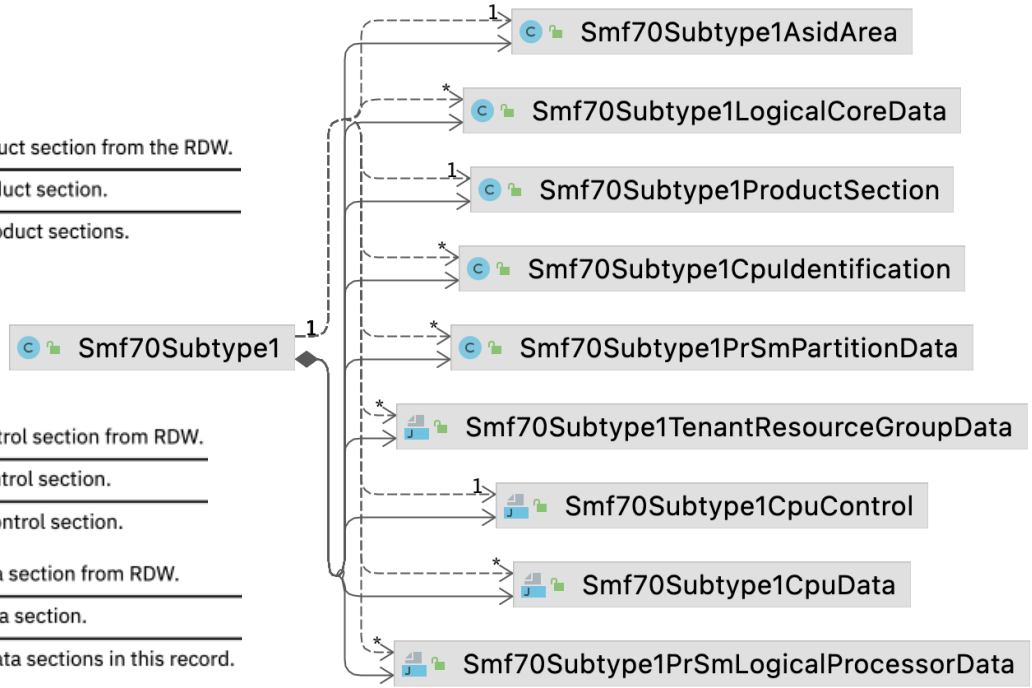


 © 2022 IBM Corporation

SMF 70-1

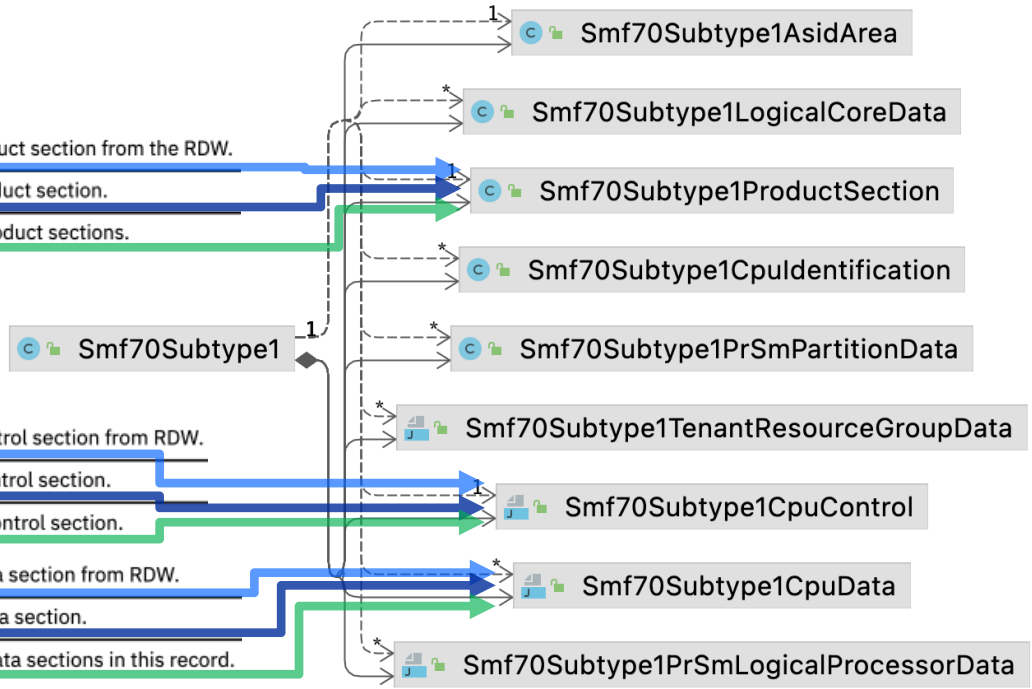
28	1C	SMF70PRS	4	binary	Offset to RMF product section from the RDW.
32	20	SMF70PRL	2	binary	Length of RMF product section.
34	22	SMF70PRN	2	binary	Number of RMF product sections.

36	24	SMF70CCS	4	binary	Offset to CPU control section from RDW.
40	28	SMF70CCL	2	binary	Length of CPU control section.
42	2A	SMF70CCN	2	binary	Number of CPU control section.
44	2C	SMF70CPS	4	binary	Offset to CPU data section from RDW.
48	30	SMF70CPL	2	binary	Length of CPU data section.
50	32	SMF70CPN	2	binary	Number of CPU data sections in this record.



SMF 70-1

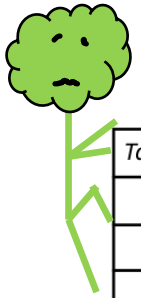
28	1C	SMF70PRS	4	binary	Offset to RMF product section from the RDW.
32	20	SMF70PRL	2	binary	Length of RMF product section.
34	22	SMF70PRN	2	binary	Number of RMF product sections.
36	24	SMF70CCS	4	binary	Offset to CPU control section from RDW.
40	28	SMF70CCL	2	binary	Length of CPU control section.
42	2A	SMF70CCN	2	binary	Number of CPU control section.
44	2C	SMF70CPS	4	binary	Offset to CPU data section from RDW.
48	30	SMF70CPL	2	binary	Length of CPU data section.
50	32	SMF70CPN	2	binary	Number of CPU data sections in this record.



Documentation for SMF 78-2

Virtual Storage Private Area Data Section

72	48	R782LSFP	40	Mixed(1)[1]	LSQA/SWA/229/230/249 free pages both above and below 16 megabytes.



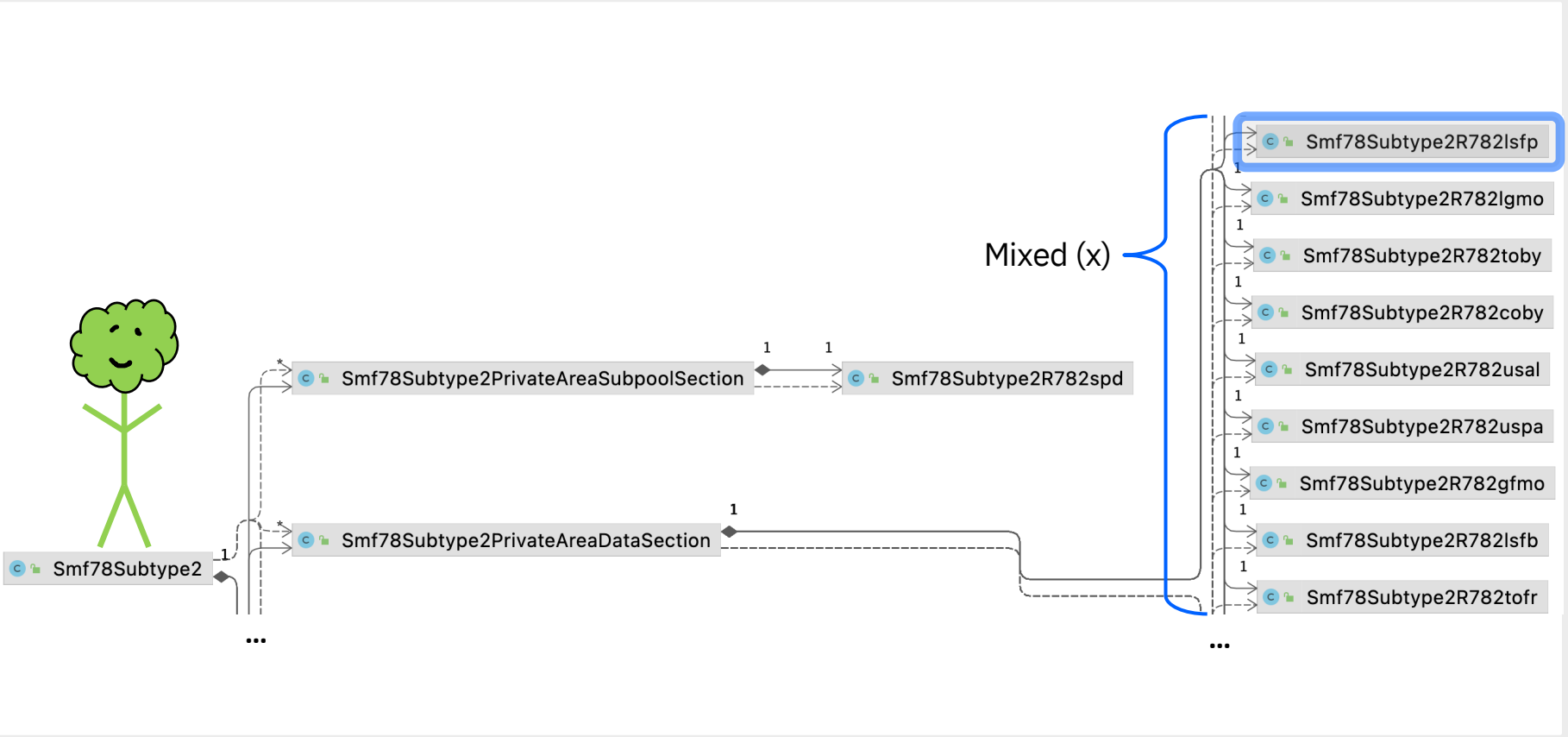
Format descriptions

For fields of format Mixed(1) containing data from below 16 megabytes, use only the first 20 bytes. For fields of this format containing data from both below and above 16 megabytes, use all 40 bytes.

Table 14. Byte structure for data collected below 16 megabytes - Mixed(1)

Offsets		Name	Length	Format	Description
0	0	VSDBMIN	4	binary	Minimum value for below 16 megabytes.
4	4	VSDBNTME	4	binary	Time stamp for minimum. Format is high-order bytes of time-of-day (TOD) clock.
8	8	VSDBMAX	4	binary	Maximum value for below 16 megabytes.
12	C	VSDBXTME	4	binary	Time stamp for maximum. Format is high-order bytes of time-of-day (TOD) clock.
16	10	VSDBTOTL	4	floating	Total for all samples below 16 megabytes (used to calculate average). See SMF78SAM to calculate averages for Common Storage data section fields, and R782SAMP to calculate averages for private area data and Private Subpool section fields.

Data Model SMF 78-2



Types & Formats

Type	Description	Common type	Output
10010010	Usually flags	Mostly boolean	true / false
[0-9]+D?	Packed numeric	Numeric	0 / 2 / 300
0YYYYDDD	Date format	ISO-date	2022-03-28
0CYYDDDF	C: Centuries since 1900 C = 0 - for range 1900-1999 C = 1 - for range 2000-2099		
TOD Clock	Date and time format	ISO-date & time ISO-date ISO-time Integer (duration)	2022-03-28T10:00:00.001 2022-03-28 10:00:00 30
HHMMSSth 0HHMMSSF MMSSTTTF 000TTTTF	Time formats -no hours -no hours / minutes / seconds	ISO-time	10:00:00.001 00:10:12.456 00:00:00.123 10:00:00 10:00

Something else?..

- ~40 distinct models at the (sub-)type level
- ~45 functional endpoints
- All OASv3 documented & conformant
- Sole use of standard types and formats:
  - Time-of-day, packed time, etc. -> ISO-8601:
    - YYYY-MM-DD`T`HH:mm:ss.fff
    - YYYY-MM-DD
    - HH:mm:ss.fff
  - Decimal / binary / packed -> Numeric
  - Sub-byte fields level -> mostly Boolean

 © 2022 IBM Corporation

Summary

Summary

- Part of z/OS-base infrastructure
- Data on demand using client-initiated synchronous Request-Response calls
- Fully documented endpoints in-place
- Fully documented Data Models in-place
- Full Data Model Composition including parent-child hierarchies and 1-to-1 & 1-to-many relationships
- Full field-level data representation in common formats
- No proprietary Types & Formats

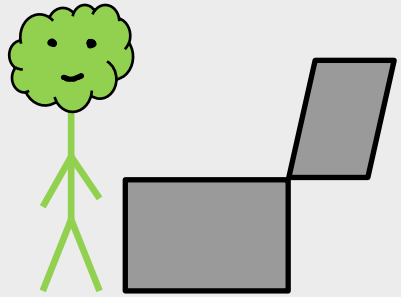
Summary - Usage

- Requirements:
  - a caller
    - web-browser
    - front-end application
    - API consuming platform
    - programmatic caller (i.e., in-code caller)
      - including generated client, e.g., via OpenAPI Generator
    - command line caller (e.g., CURL)
    - …
  - an SMF Dataset
- Result
  - modelled and formatted raw data in JSON as it is in the Dataset

Summary - Input

- Minimal input required
  - an SMF dataset name
- Additional input available
  - time-range – limits processing to the records within the specific start and / or end time
  - LPAR name – limits processing to the records for specific LPAR only
  - (sub)type specific filters – limits the processing and output to specific fields and / or sections only

Summary - Output



- Output
  - JSON with an array of Objects
  - each Object represents an SMF record of a specific (sub)type
  - each Object includes all the hierarchy requested and / or available
  - each Object includes all the fields requested and / or available

Requirements and Shipment

- Part of z/OS-base infrastructure
- z/OS 2.5 is required
  - z/OS Data Gatherer ADG feature is NOT required
- Shipping vehicles for z/OS 2.5 are
  - OA62062 (setup)
    - PTF UJ09235
  - OA62081 (support)
    - PTF UJ07630
  - PH41855 (z/OSMF support)
    - PTF UI82355

Example Consumer Use Case: IBM SMF Explorer

- Also uses OA62081 as shipment vehicle
- Runs on your workstation or a cloud instance
- Jupyter Notebooks are the main deliverables besides the IBM SMF Explorer Python package:

We want to build a community around Notebooks

Using GitHub to share our Notebooks and to give you the chance to contribute

Check out today's lab right after the coffee break ;)

DG REST Services and IBM SMF Explorer
11:15 - 12:45

Further information

- z/OS Data Gatherer Programmer's Guide
- z/OS Data Gatherer User's Guide
- z/OS Management Facility Configuration Guide
- MVS System Management Facilities (SMF)
- https://github.com/IBM/IBM-Z-zOS/tree/main/zOS-DataGatherer

Thank you

Johannes Doerk
z/OS Data Gatherer Software Developer

Contact

E-Mail johannes.doerk@ibm.com
LinkedIn / Xing Johannes Doerk

 © 2022 IBM Corporation

