



Making use of the REST api's around me...

CCdug - May 13th 2024

Toine Michielse, Solutions Architect

toine.michielse@broadcom.com

Disclaimer

Certain information in this presentation may outline Broadcom's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of Broadcom or its licensees under any existing or future license agreement or services agreement relating to any Broadcom software product; or (ii) amend any product documentation or specifications for any Broadcom software product. This presentation is based on current information and resource allocations as of April 14, 2024, and is **subject to change or withdrawal by Broadcom at any time without notice. The development, release and timing of any features or functionality described in this presentation remain at Broadcom's sole discretion.**

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future Broadcom product release referenced in this presentation, Broadcom may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to Broadcom maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

Broadcom may use any feedback provided by you related to a Broadcom product or this presentation for any Broadcom business purposes (including but not limited to, preparation, reproduction, and distribution of derivative works based upon such feedback), without any obligation to you including consent or payment.

Copyright © 2024 Broadcom. All rights reserved. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries. Broadcom, the pulse logo, Connecting everything, CA Technologies and the CA Technologies logo are among the trademarks of Broadcom.

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. Broadcom assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will Broadcom be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if Broadcom is expressly advised in advance of the possibility of such damages.

| Agenda

- Introduction
- Architecture
- Three use cases

Let me quickly introduce myself (Toine)

- Toine Michielse, born in The Netherlands
 - Db2 programmer, DBA, System Engineer, Architect
- Many years as Db2 for z/OS Lab Advocate
- Before joining Broadcom in Madrid
 - Mainframe architect at SwissRe
 - Leading Cap/Performance mgmt. team
 - Leading Db2 Consultancy team

My passions:

Db2, data,
*mainframe
modernization*



Paragliding



Playing drums
with “Ciencia
Urbana”
(see Spotify
...if you like rock)

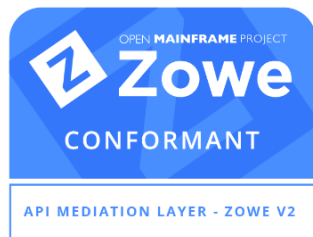


Introduction



APIs are at the heart of platform integration

- Application programming interface
- Cross product / cross platform
- Communication protocol
- Popular industry standard
- Recently adopted by mainframe vendors



Application programming interface

Type of software :



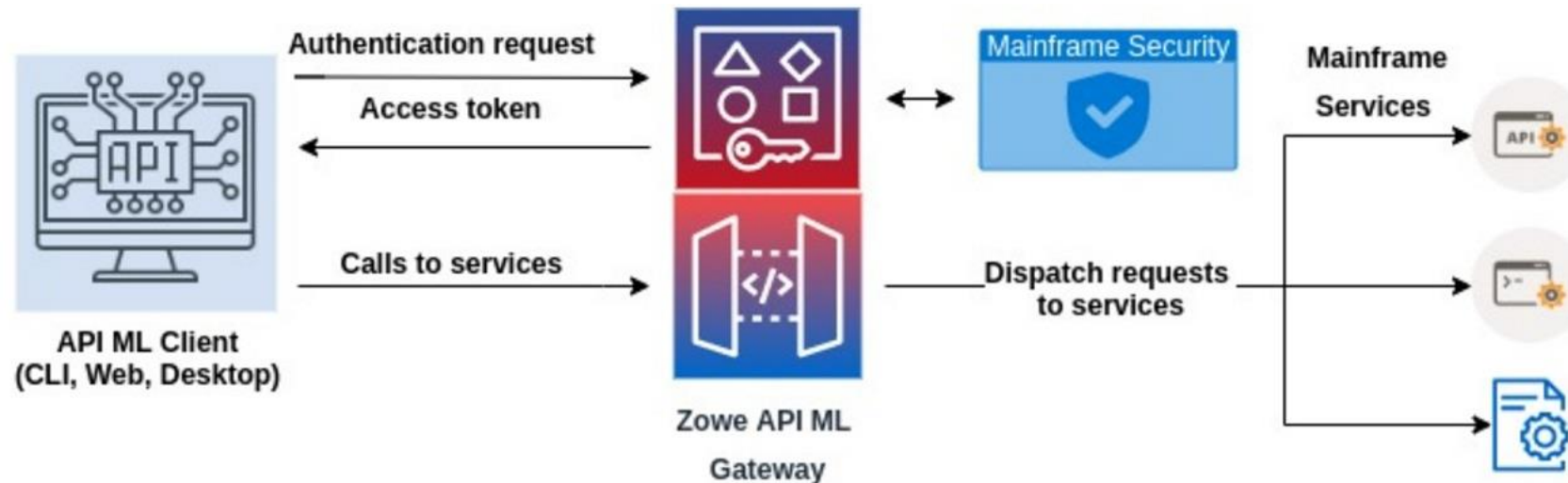
An application programming interface is a way for two or more computer programs or components to communicate with each other. It is a type of software interface, offering a service to other pieces of software. [Wikipedia](#)

**Not necessarily “open source” but a relevant building block*

The API Mediation Layer is Critical

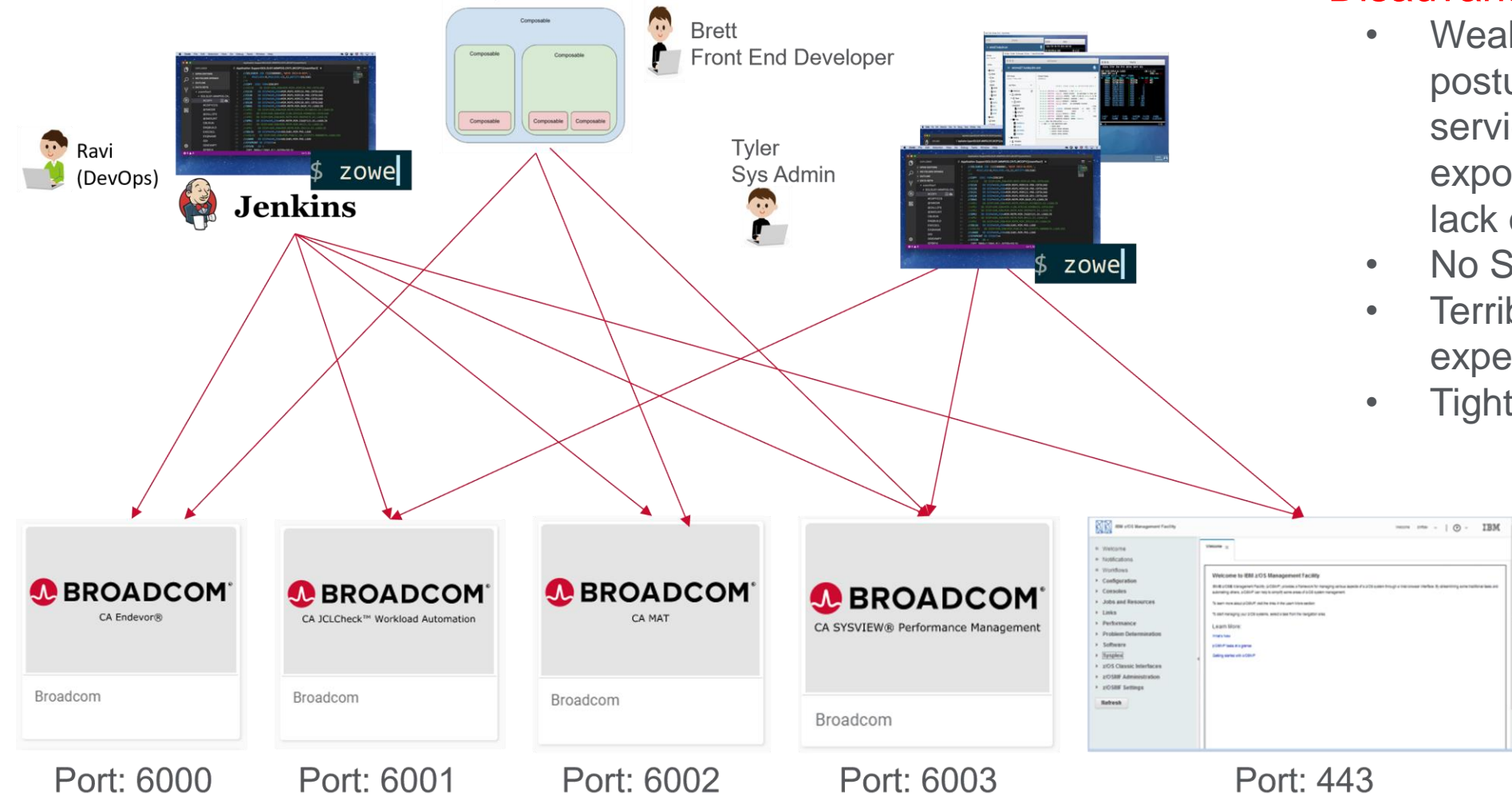
- Strengthens security posture
- Addresses a set of essential cross-cutting API concerns, including
 - ✓ Enhanced security, usability, discoverability, and resiliency
 - ✓ Improved availability, scalability, and manageability
- Superior and simpler user experience for API consumers

There are no viable alternatives to address these concerns



API Landscape *without* API Mediation Layer

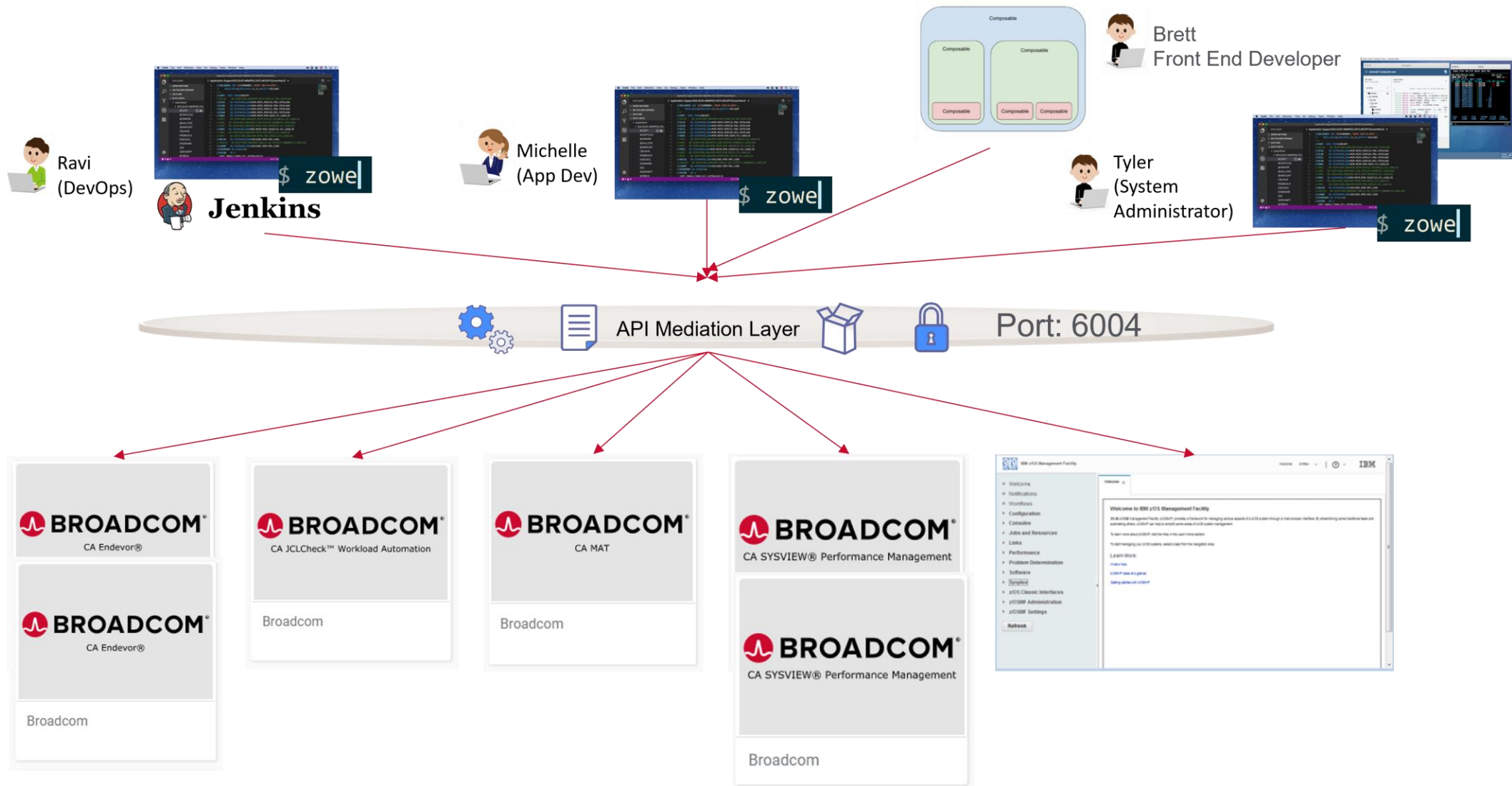
Imagine this at scale! Not for the faint of heart...



Disadvantages:

- Weaker security posture: each API service needs its own exposed port and URL; lack of observability
- No SSO support
- Terrible API user experience
- Tight API/client coupling

API landscape *with* API Mediation Layer





Let's look at what we can build with existing
Broadcom APIs to modernize the way we work with
the mainframe

Use Case #1: Performance Monitoring

Db2 for z/OS systems need to be monitor for trends and exceptional situations

REST API exposes performance metrics

Time series database collects data for analysis

- Invokes the API at regular intervals
- Query language enables deep analysis

Dashboard provides visualization of the performance metrics

- Provides user customizable graphic displays
- Generates alerts based on user defined thresholds

Operations/DBA uses dashboards to monitor and analyze performance

Using open source

- Prometheus is an open-source component
 - A time series database and alerting engine



- Grafana is an open-source software
 - Provides powerful data visualization for analytics and monitoring
 - Supports various data sources
 - Allows creating, exploring, and sharing dashboards
 - Integrates well with Prometheus time-series database



Compliant versus non-compliant

Prometheus format (compliant)

Request URL

```
https://[redacted]/dbm/api/v1/idb2/prometheus/generic?function=DSAISTD&delta=true&ssid=D121
```

Server response

Code	Details
200	<p>Response body</p> <pre># HELP YEAR N/A # TYPE YEAR gauge YEAR{ssid="D121",group="D120",function="DSAISTD"} 2024.0 # HELP MONTH N/A # TYPE MONTH gauge MONTH{ssid="D121",group="D120",function="DSAISTD"} 2.0 # HELP DAY N/A # TYPE DAY gauge DAY{ssid="D121",group="D120",function="DSAISTD"} 13.0 # HELP HOUR N/A # TYPE HOUR gauge HOUR{ssid="D121",group="D120",function="DSAISTD"} 3.0 # HELP MSTR_TCB N/A # TYPE MSTR_TCB gauge MSTR_TCB{ssid="D121",group="D120",function="DSAISTD"} 0.038055 # HELP MSTR_SRB N/A # TYPE MSTR_SRB gauge MSTR_SRB{ssid="D121",group="D120",function="DSAISTD"} 0.004139 # HELP MSTR_SRB_PREEMPT N/A # TYPE MSTR_SRB_PREEMPT gauge MSTR_SRB_PREEMPT{ssid="D121",group="D120",function="DSAISTD"} 0.006368 # HELP MSTR_ZIIP N/A # TYPE MSTR_ZIIP gauge MSTR_ZIIP{ssid="D121",group="D120",function="DSAISTD"} 0.001051 # HELP MSTR_IO_INTERRUPT N/A # TYPE MSTR_IO_INTERRUPT gauge MSTR_IO_INTERRUPT{ssid="D121",group="D120",function="DSAISTD"} 3.63E-4 # HELP DBM1_TCB N/A</pre> <p>Response headers</p>

Regular JSON format (non-compliant)

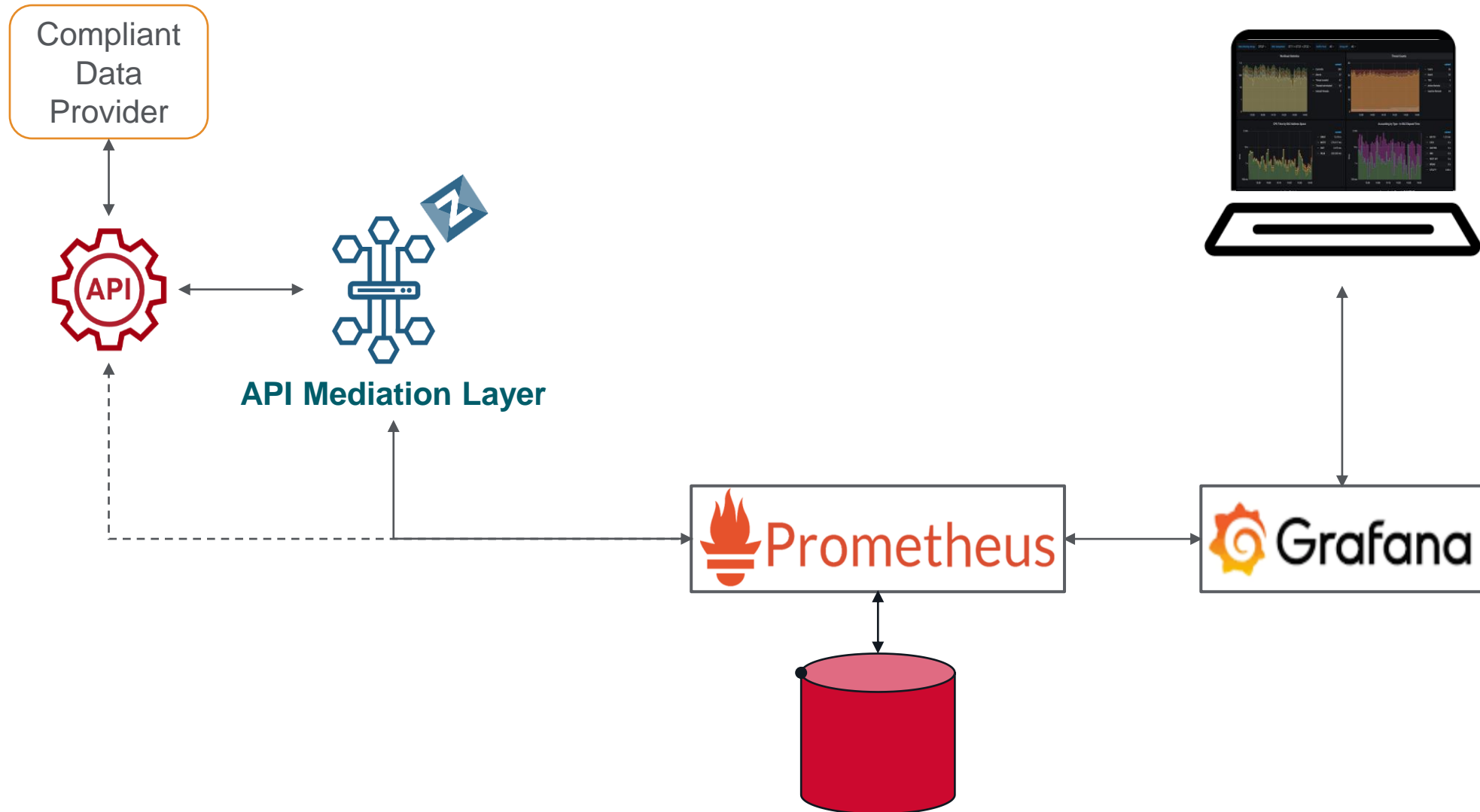
Request URL

```
https://[redacted]/api/v1/journals/IDMS
```

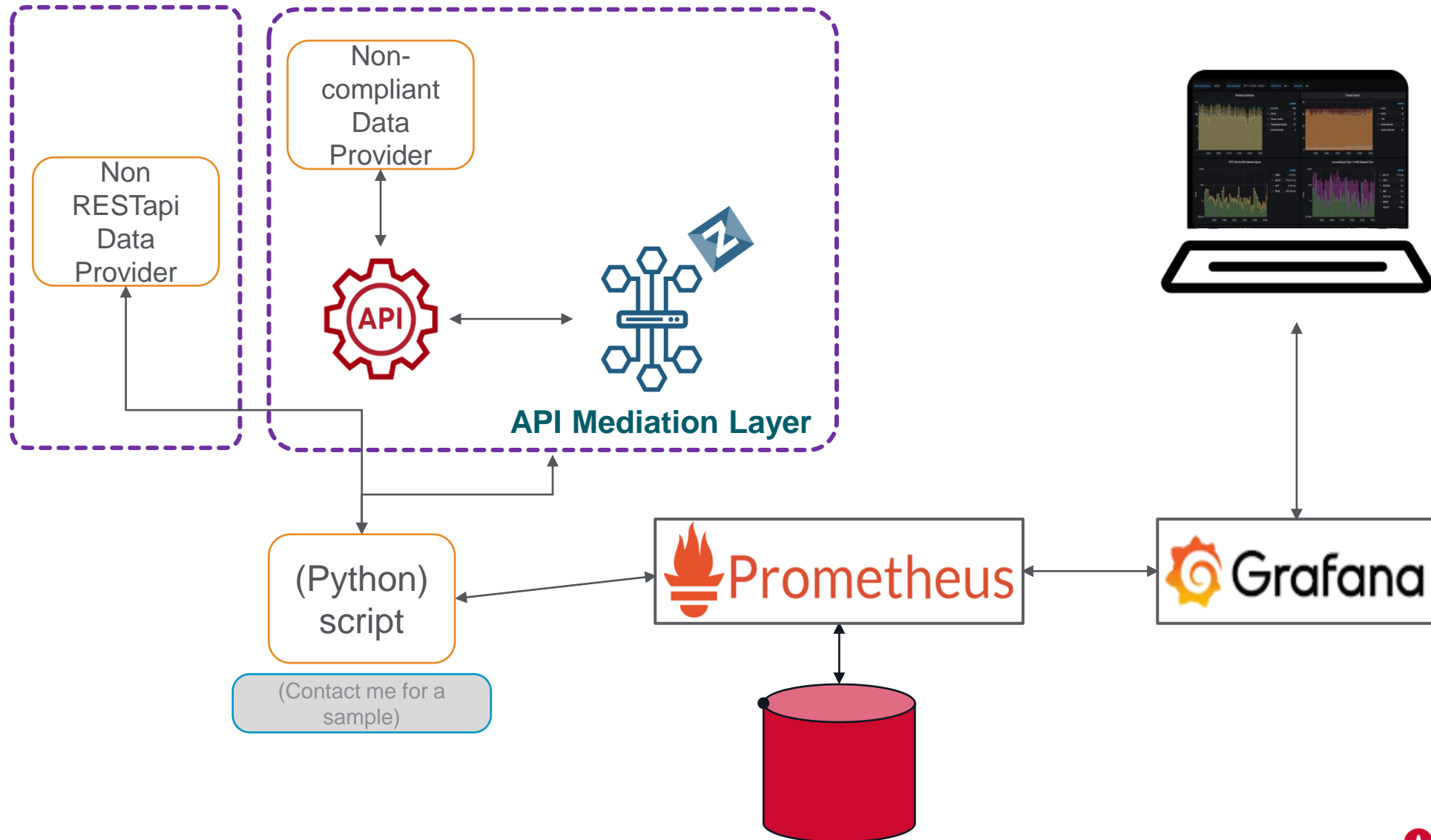
Server response

Code	Details
200	<p>Response body</p> <pre>[{ "journalName": "J1JRNL", "currentSegmentNumber": 21, "lowRelativeBlockNumber": 10, "highRelativeBlockNumber": 5000, "currentRelativeBlockNumber": 2298, "dsegRelativeBlockNumber": 5020, "dsegInterval": 0, "runUnitLevel": 3, "runUnitsWaitingRecoveryCount": 0, "offloadStatus": "", "contentStatus": "", "currentStatus": "ACTIVE" }, { "journalName": "J2JRNL", "currentSegmentNumber": 0, "lowRelativeBlockNumber": 0, "highRelativeBlockNumber": 0, "currentRelativeBlockNumber": 0, </pre>

| Pure pull model



Using push to Prometheus model



Increased demands, fewer resources, shorter on-ramps

Need for speed

- Problem resolution turnaround time, quick detection
- Problem avoidance

Need to “shift left”

- Free up time of the scarce, highly skilled resources
- Empower more users

Need for context

- Time (when an activity happened) and Persona (who is in the driver’s seat)

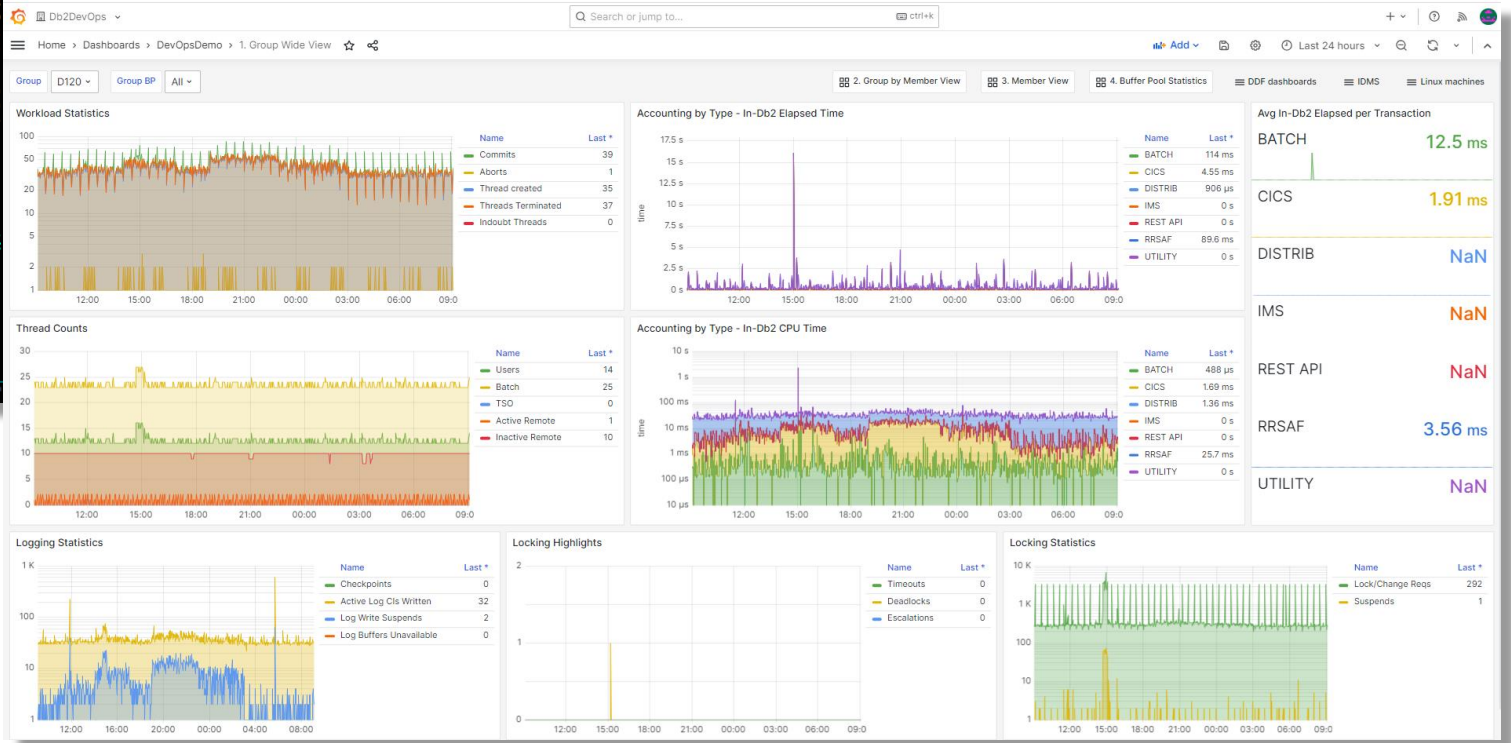
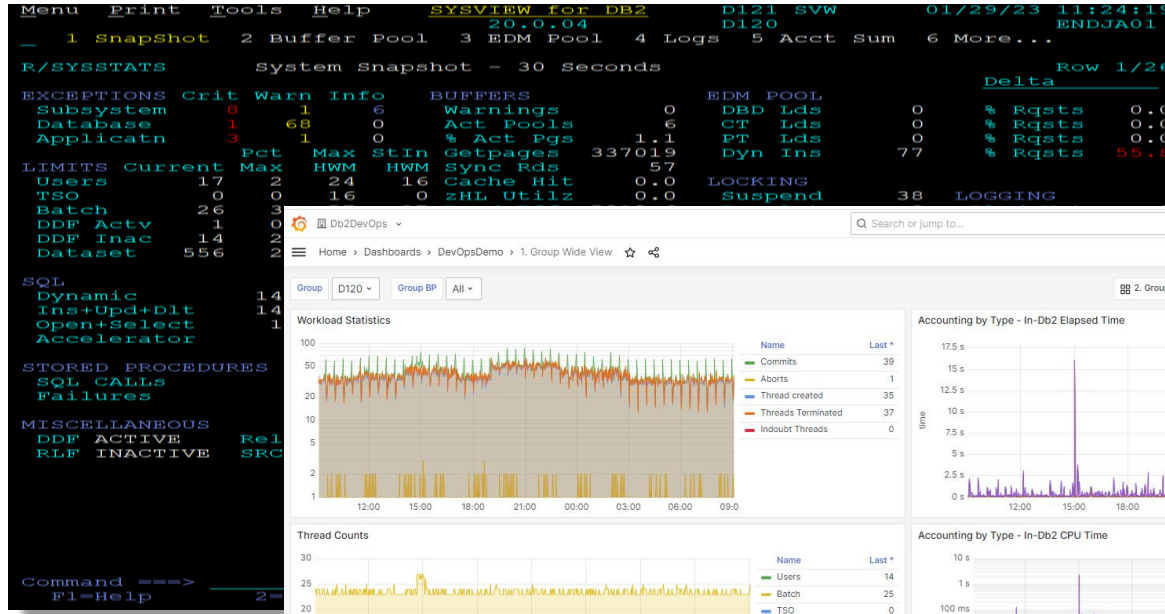


Dashboarding helps....

Need for speed

Need to “shift left”

Need for context



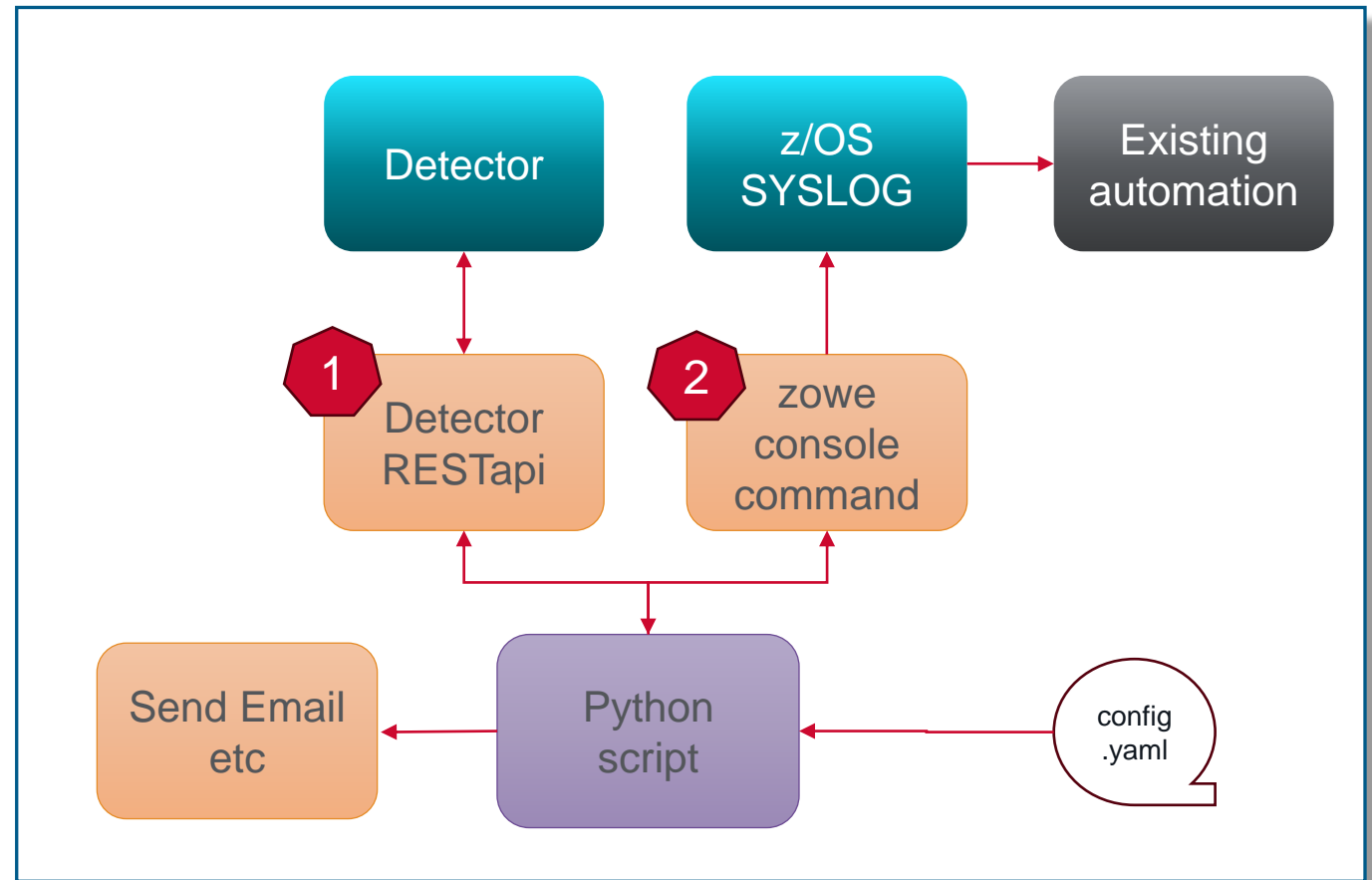
Use Case #2:

Augmenting product capabilities

- Customer was looking to trigger automation/actions when certain SQL codes appear
 - Integrate best of two platforms (Linux and Mainframe)
 - The answer: RESTapi
- Detector is excellent SQL statement level monitor
 - Performance
 - SQL codes

Use case 2: augmenting capabilities

- Implementation:
 - Python scripts polls Detector at predefined intervals
 - To trigger existing automation
 - Issue a z/OS SEND command
 - Implement other action triggers



Augmenting capabilities: code snippets

2

```
import json
import subprocess

def action_wto(wtoMsg):
    cmd = "\"SE 'RESTapi Demo SQLcodeMonitor: {}',OPERATOR=11\"".format(wtoMsg)
    zowe_output_str = subprocess.check_output(
        ['zowe', 'zos-console', 'issue', 'command', cmd,
         '--zosmf-profile', 'SVW.zosmf', '--response-format-json'])
    zowe_output = json.loads(zowe_output_str) # parse the json response

    if not zowe_output["success"]:
        print("Console command failed!")
        print(str(zowe_output["stderr"]))
        exit(1)
    print("\n\t A wto has been issued")
    return
```

```
import json
import urllib
```

```
base_uri="https://<your host>"
base_path="dbm/api/v1/pdt"
api_port="<your port>"
credentials="xxxx" (base64 encoded for basic authentication)
```

```
def get_endpoint(endpoint,parms):
    url = "{}:/{}/{}?{}".format(base_uri, api_port, base_path, endpoint,parms)
    request = urllib.request.Request(url, headers={'Authorization': 'Basic ' + credentials})
    response = json.loads(urllib.request.urlopen(request, timeout=30, context=context).read())
    return response
```

```
import getpass
import base64
```

```
user = input("Userid: ")
pswd = getpass.getpass('Password: ')
```

```
credentials = base64.b64encode("{}:{}".format(user,
pswd).encode('ascii')).decode('ascii')
print("credential:",credentials)
```

Augmenting capabilities: sample output

```
SQLcode monitor started
  Monitor interval: 30
  Generate SQL      : True
  Config file       : sqlMonitorConfig.yml
```

Checking for sqlcodes at 2024-03-26 16:57:30

New instances of sqlcode -950

Error text:

```
SQLCODE = -950, ERROR: THE LOCATION NAME SPECIFIED IN THE CONNECT STATEMENT IS
INVALID OR NOT LISTED IN THE COMMUNICATIONS DATABASE
```

SQL text:

```
SELECT NAME,CREATOR FROM DB2DOESNOTEXIST.SYSIBM.SYSTABLES
```

New instances of sqlcode -206

An email has been sent to dba@broadcom.com

subject: SQLCODE -206 occurred, take action

body: Time 2024-03-26 11:12:55;

Connection TSO; Corrid MICT001; Authid MICT001, Plan DSNESPCS;

Package DSNESPCS.DSNESM68.1A0D8BD811DAADD8(UI36064)

Error text:

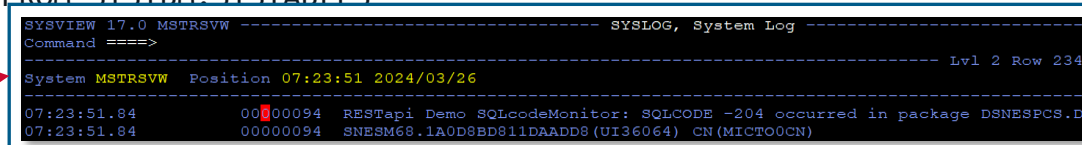
```
SQLCODE = -206, ERROR: GEKKECOLUMN IS NOT VALID IN THE CONTEXT WHERE IT IS USED
```

SQL text:

```
SELECT GEKKECOLUMN FROM SYSIBM.SYSTABLES
```

New instances of sqlcode -204

A wto has been issued



```
SYSVIEW 17.0 MSTRSVW ----- SYSLOG, System Log -----
Command =====
----- Lvl 2 Row 234
System MSTRSVW Position 07:23:51 2024/03/26
-----
07:23:51.84      00000094  REStapi Demo SQLcodeMonitor: SQLCODE -204 occurred in package DSNESPCS.D
07:23:51.84      00000094  SNESM68.1A0D8BD811DAADD8 (UI36064) CN(MICT00CN)
```

Use Case #3:

Productivity boost for IDMS operations

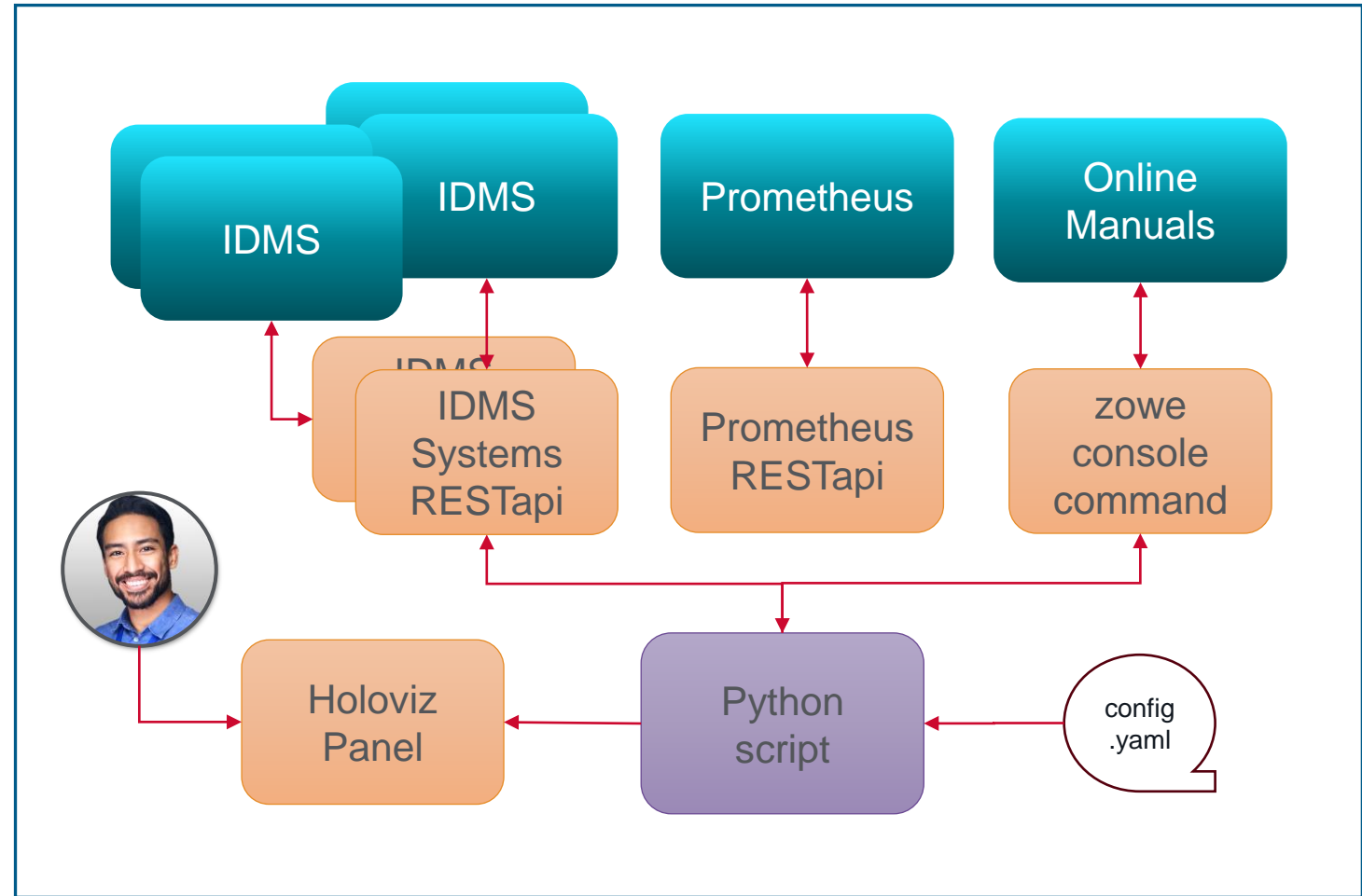
The operator needs to log into the appropriate IDMS system to perform

1. Cancel task
2. View logs

Other views and application configuration options

Use case 3: productivity boost

- Implementation:
 - Python scripts uses Holoviz Panel as a GUI
 - Gets list of all IDMS CV's it can reach through configured providers
 - From selected CV, gets all configured endpoints data
 - Implements controlled user task cancellation
 - Insights in the log usage from Prometheus
 - Manual supported operator command execution



Productivity boost: configurable UI

≡ IDMS System RESTapi Sample Application (user : ██████████)

Environment: Prod(TCDE) | System: IDMS

Refresh: 2024-03-27 07:48:29

Config | Cancel | Apply

Transaction (selected) | UserTask | Journal | SysTask

taskNumber, taskCode, currentProgram, taskStatus, subschemaName, databaseProcessingStatus, databaseWaitStatus, databaseLockStatus, databaseCallsCount, pagesWrittenCount, pagesReadCount, pagesRequestedCount, recordsRequestedCount, recordCurrentOfTransactionCount, recordFragmentCount, nonShareLocksHeldCount, shareLocksHeldCount, totalLocksAcquiredCount, journalBeforeImageCount, journalAfterImageCount, calcRecordsWithoutOverflowCount, calcRecordsWithOverflowCount, viaRecordsWithoutOverflowCount, viaRecordsWithOverflowCount, sr7RecordCount, sr8RecordCount, sr8RecordDeleteCount, orphanedRecordsAdoptionCount, sr8SpawnCount, sr8SplitCount, fewestLevelsSearchedCount, greatestLevelsSearchedCount, lastDatabaseVerbNumber, currentAreaName, currentRecordName

System | Log | Journal | Commands | Config

Transactions

Task	taskCode	currentProgram	taskStatus	subschemaName	DB ProcSt	Db LockSt	Db Calls	PG Req	Recs req	Calc ovflw	Via ovrf
2		RHDCRUAL	WAIT	IDMSNWK7		H	39	8	8	0	
3		RHDCRUAL	WAIT	IDMSNWKL		H	177,603	6	6	0	
4		RHDCRUAL	WAIT	IDMSNWK6		A	279	140	214	0	
5		RHDCRUAL	WAIT	IDMSSECU		H	58	12	12	0	

User tasks

Task	taskCode	Status	taskPriority	currentProgram	SysTime	UserTime	ecb1	ecb2	ecb3
53	OPER	WAIT	100	RHDCOPER	0	0	*TIMER*	*TIMER*	PTERM

Journals

Name	Segment	LowRBN	HighRBN	CurRBN	RULevel	RUWait
J1JRNL	21	10	5,000	2,300	3	
J2JRNL	0	0	0	0	0	
J3JRNL	0	0	0	0	0	
J4JRNL	0	0	0	0	0	

System tasks

Task	taskCode	currentProgram	Status	ecb1	ecb2	ecb3	SysTime	UserTime
0	*SYSTEM*	MASTER	WAIT	PLE	LTTMSECB	SRVT ECB	0	0
1	*SYSTEM*	*DBRC*	WAIT	DBRCWTOR	ESEECB	CCEECB	0	0
2	SRVCDVR	RHDCRUSD	WAIT	SDCSECB	*TIMER*		0	0
3	SRVCDVR	RHDCRUSD	WAIT	SDCSECB	*TIMER*		0	0
4	SRVCDVR	RHDCRUSD	WAIT	SDCSECB	*TIMER*		0	0

```

IDMS-DC Release 1900 Display Active Tasks
Task Id Task Cd Program Terminal Pri Stat User Id
0000000053 OPER RHDCOPER LVTM001 100 ACTV
0000000000 *SYSTEM* *MASTER* 255 WAIT
0000000001 *SYSTEM* *DBRC* 255 WAIT
0000000014 *DRIVER* CCILINE 254 WAIT
0000000015 *DRIVER* TCPIP 254 WAIT
0000000016 *DRIVER* UCFLINE 254 WAIT
0000000017 *DRIVER* VTAM01 254 WAIT
  
```


Productivity boost: user task cancellation

The screenshot displays a web interface for managing user tasks. At the top, there is a section titled "User tasks" with a dropdown arrow. Below this is a table with columns: Task, taskCode, Status, taskPriority, currentProgram, SysTime, UserTime, ecb1, ecb2, and ecb3. A single row is visible with the following values: 53, OPER, WAIT, 100, RHDCOPER, 0, 0, *TIMER*, *TIMER*, PTERM. Below the "User tasks" section is another section titled "Journals" with a dropdown arrow. It contains a table with columns: Name, Segment, LowRBN, HighRBN, CurRBN, RULevel, and RUWaiting. The rows are: J1JRNL (21, 10, 5,000, 2,300, 3), J2JRNL (0, 0, 0, 0, 0), J3JRNL (0, 0, 0, 0, 0), and J4JRNL (0, 0, 0, 0, 0). A modal dialog box is overlaid on the interface, titled "Usertask selected for deletion IDMS in environment Prod(TCDE)". The dialog contains the text: "Taskcode: OPER", "Type: Operator session", and "This task is Cancellable". At the bottom of the dialog are two buttons: "Close" and "Terminate". A red arrow points from the "Terminate" button to a terminal window below.

```
PREVIOUS TASK ABENDED WITH ABEND CODE MTTA  
V190 ENTER NEXT TASK CODE: CA IDMS release 19.0 tape GJJ04I node SYST0190
```

Productivity boost: OPS – SME communication

The image displays two screenshots of the IDMS System RESTapi interface, illustrating a productivity boost through log condensation.

Left Screenshot: Shows the 'System log for IDMS in environment Prod(TCDE)'. The log data is displayed as a list of entries:

Date	Time	Message
2024-03-21	22:59:59.144432	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-22	15:39:58.543885	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-23	08:20:00.300210	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-24	01:00:02.178463	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-24	17:39:59.747781	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-25	10:19:58.793874	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-26	03:00:03.653388	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-26	19:39:59.556827	IDMS DC050001 V190 T11 DCLOG IS 37% FULL

Right Screenshot: Shows the same log data after condensation. The log data is displayed as a condensed view:

Date	Time	Message
2024-03-21	22:59:59.144432	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-24	01:00:02.178463	IDMS DC050001 V190 T11 DCLOG IS 37% FULL
2024-03-26	03:00:03.653388	IDMS DC050001 V190 T11 DCLOG IS 37% FULL

Red arrows and boxes highlight the transition from the full log view to the condensed view, demonstrating the productivity boost.

Productivity boost: integrate data sources

The screenshot displays the IDMS System RESTapi Sample Application interface. The main area shows the command 'DCMT DISPLAY ACTIVE TASKS' being executed. The output is as follows:

```

Output for DCMT DISPLAY ACTIVE TASKS
using system/datasource IDMS/EMPLDEMO
Returned: 52 records

DCMT DISPLAY ACTIVE TASKS :

      Current max tasks      40
      Times at max tasks     0
      Allocated DCE/TCE      40
      Number of tasks abended 1
      Number of tasks processed 73
      Number of tasks active  19

Taskid Taskcd Prog  LTERM Pri Stat Stim A(ECB) ECB Type
000000000 *SYSTEM* *MASTER* 255 WAIT NOST 0006504C PLESECB
                                135242A0 LTTMSECB
                                00164A2C Service Task ECB
000000001 *SYSTEM* *DBRC* 255 WAIT NOST 00070988 DBRC WTOR ECB
                                0F02BF90 ESEECB
                                0003A4E4 CCEECB
                                0003A604 CCEECB
                                0003A6C8 CCEECB
                                0003A754 CCEECB
                                0003A720 CCEECB
000000002 *DRIVER* RHDCRUSD 253 WAIT NOST 13546390 SERVICE DRIVER ECB
                                1413158C TIMER ECB
                                13546410 SERVICE DRIVER ECB
  
```

The interface also includes a 'Syntax help' section for the command, which provides the following information:

Syntax

```

DCMT broadcast-params
Display Active Tasks
  
```

Parameters

broadcast-params
Indicates to execute the DCMT command on all or a list of data sharing group members.

Usage

Global Task Statistics

DCMT DISPLAY ACTIVE TASKS displays global task statistics and information on each active task thread. The following global task statistics are provided:

Field	Value
Current max tasks	Maximum number of task threads that can be active concurrently
Times at max tasks	Number of times a maximum tasks condition occurred
Allocated DCE/TCE	Number of dispatch control elements (DCEs) and task control

| Questions



Broadcom Mainframe Technical Exchanges

- ✓ European in-person event in Prague: April 16-18
- North American in-person event in Plano, TX: September 10-12 ([registration open](#))
- Global virtual event: October 8-10 ([save the date](#))

Make plans to attend

- Network with peers and Mainframe technical experts
- Technical education, product update, how-to and roundtable sessions
- No registration fee! Open to all Broadcom customers
- Learn more: <https://bit.ly/MainframeTechEx>

