



Modern Enterprise Application development with IBM Wazi for VS Code IBM RSE API vs z/OSMF Deep Dive

August 2024

Dr. Peter Haumer (phaumer@us.ibm.com)
David McKnight (dmcknigh@ca.ibm.com)



Disclaimer

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- The numbers are a result of adopting IBM tools and industry research data with some client verified assumptions. No warranties are made with respect to the actual improvement your organization may achieve.

CLI – Enables cloud-like access to mainframe

- Enables app developer and DevOps engineers to interact with the mainframe easily through a CLI from any terminal on Windows, MacOS, Linux
- Easily integrates with IDEs, shell commands, bash scripts, and build tools; installs using NPM

• **Interact with mainframe files**

Create, edit, download, and upload mainframe files (data sets) directly

• **Submit jobs**

Submit JCL from data sets or local storage, monitor status, view and download output automatically

• **Issue TSO and z/OS console commands**

Issue TSO and console commands to the mainframe directly

• **Integrate z/OS actions into scripts**

Build local scripts that accomplish both mainframe and local tasks

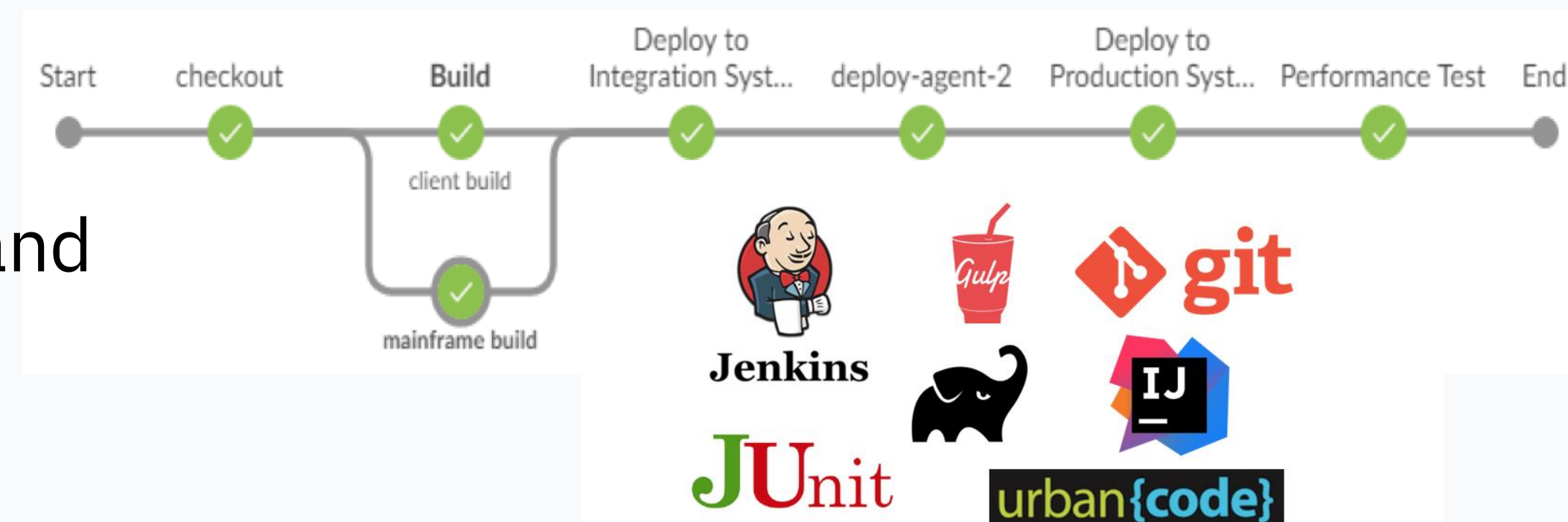
• **Produce responses as JSON documents**

Return data in JSON format on request for consumption in other programming languages

• **CLI Plug-Ins**

Access to CICS, DB2, IMS, MQ, FTP, IDF

Build | Test | Deploy



```
GROUPS
-----

plugins          Install and manage plug-ins
profiles         Create and manage configuration profiles
provisioning | pv Perform z/OSMF provisioning tasks on Published Templates
                  in the Service Catalog and Provisioned Instances in the
                  Service Registry.

zos-console | console Issue z/OS console commands and collect responses
zos-files | files      Manage z/OS data sets
zos-jobs | jobs        Manage z/OS jobs
zos-tso | tso          Issue TSO commands and interact with TSO address spaces
zosmf            Interact with z/OSMF

OPTIONS
-----

--version | -v (boolean)

    Display the current version of CA Brightside

GLOBAL OPTIONS
-----

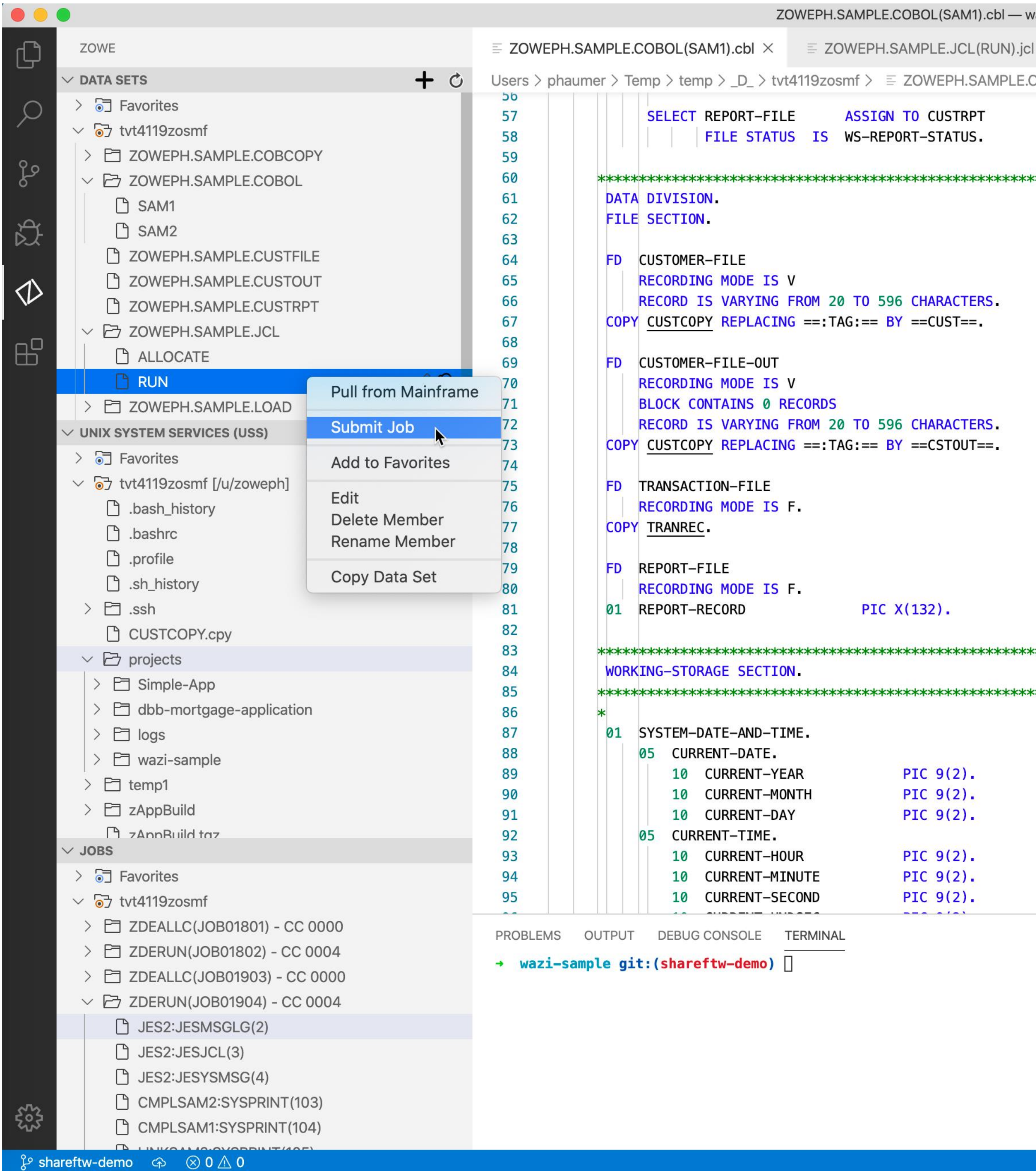
--response-format-json | --rfj (boolean)

    Produce the command response as a JSON document

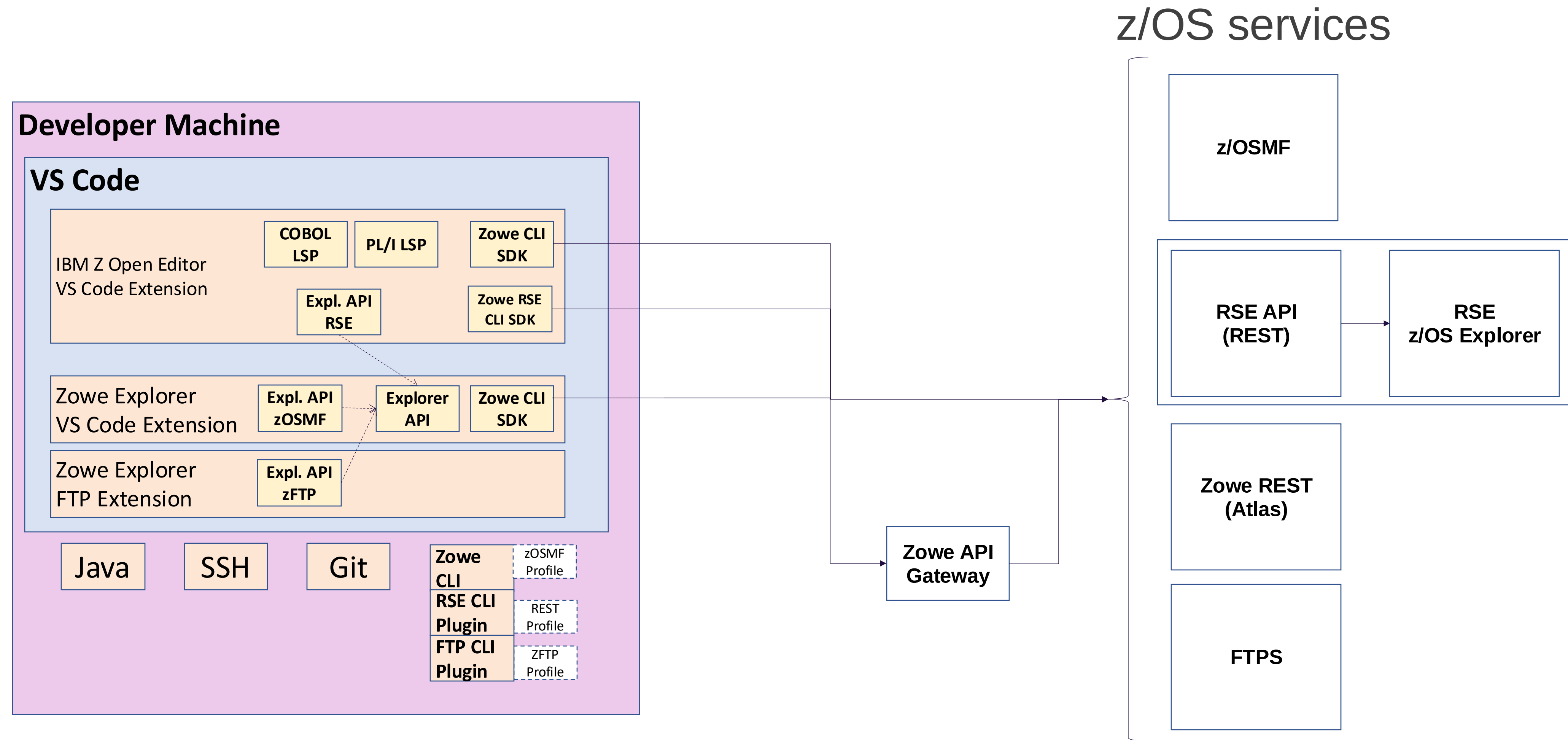
--help | -h (boolean)
```


Zowe Explorer VS Code Extension

- Exposes a large set of Zowe CLI operation to a user-friendly tree browser UI in Microsoft VS Code
 - Three unique views for Data Sets, USS and Jobs.
 - Executes against Zowe CLI APIs that use z/OSMF.
-
- Maintained by Zowe CLI Squad.
 - Broadcom and IBM main contributors.
 - Freely available in VS Code Marketplace.



Wazi for VS Code extending Zowe CLI and Explorer: Transparent support of different z/OS connectivity APIs



High-Level Comparison

z/OSMF

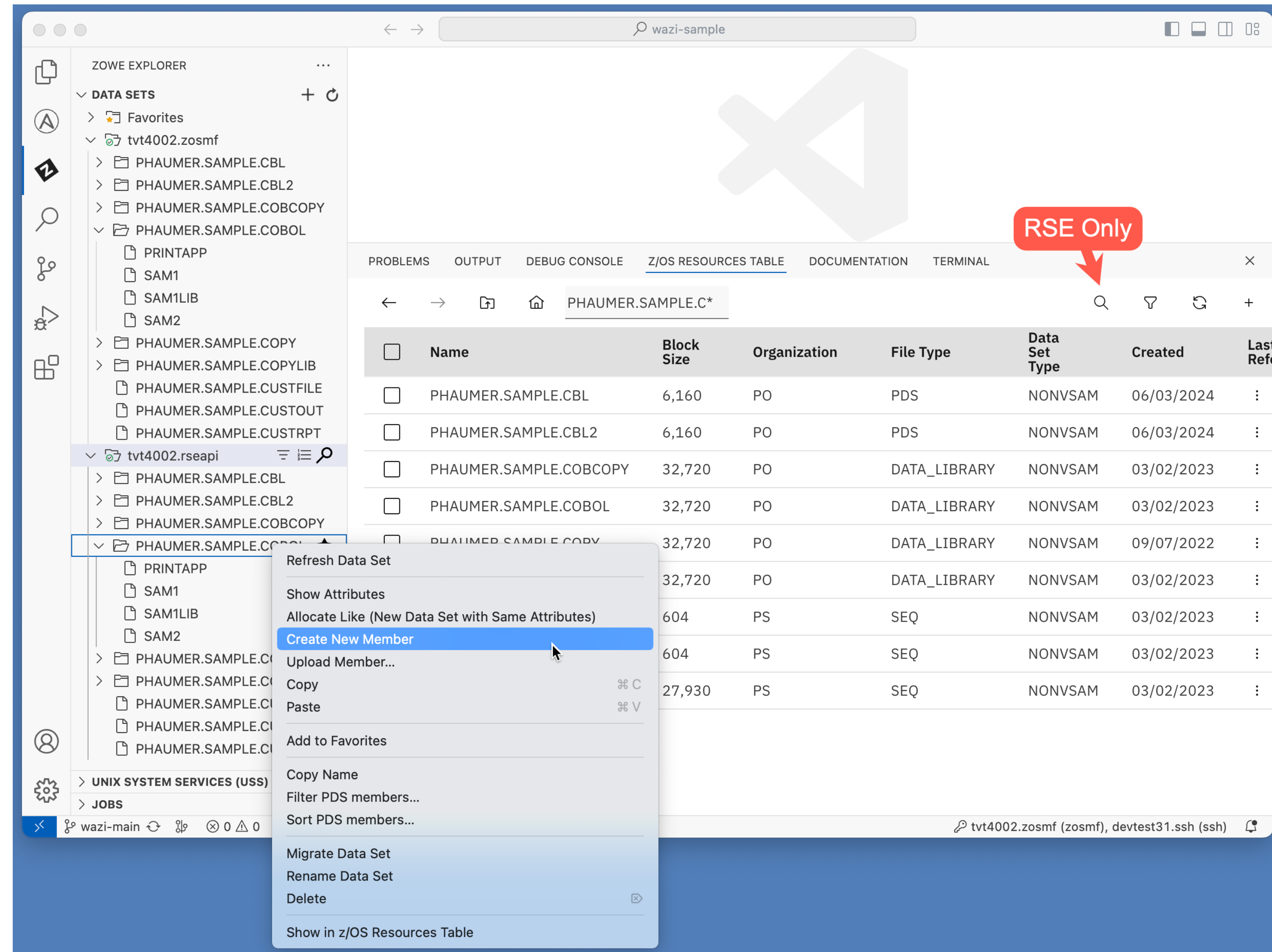
- Part of z/OS
- Focus on system management use cases
- Default for Zowe CLI, Explorer and other tools
- Zowe utilizes data sets, UNIX files, Jobs, TSO, workflows, logs
- SSH used directly without z/OSMF by Zowe

Remote System Explorer API

- Provided with z/OS Explorer (free) and Developer for z/OS
- Focus on developer use cases
- Wazi for VS Code extends Zowe CLI/Explorer
- RSE API CLI Plugin utilizes data set, UNIX files, Jobs, TSO
- RSE API CLI Plugin add additional features for encoding maps, searching content, commands (ssh replacement), and custom name/value store

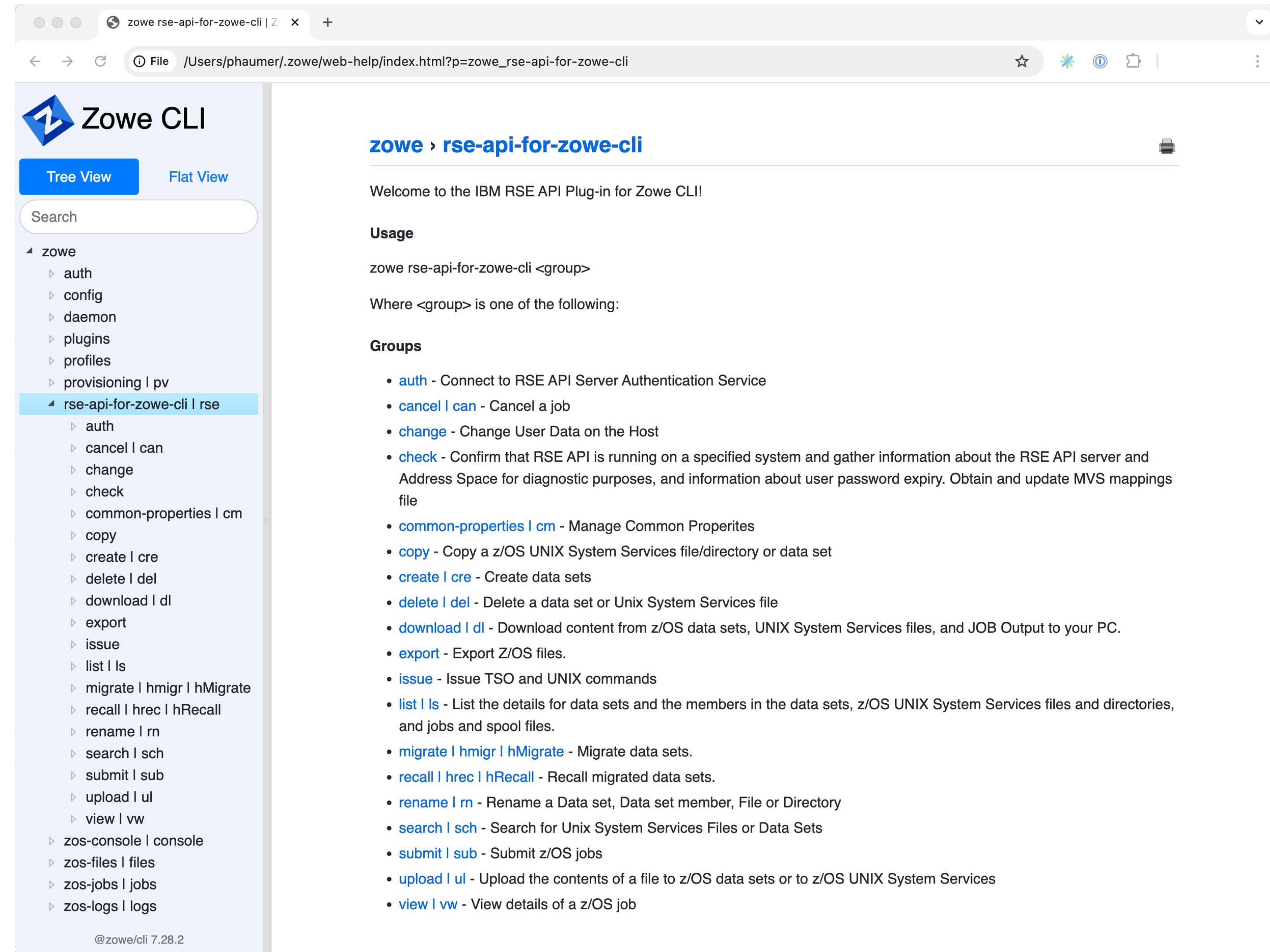
Zowe Explorer can use both

- Z Open Editor utilizes Zowe Explorer's extensibility API to inject RSE API as an additional protocol.
- All protocols can be used at the same time giving the user choices.
- Differences between protocols result into different menu and button actions.
- E.g. only RSE has a content search function in the Z Open Editor's z/OS Resources table.



Full CLI support of all APIs

- RSE API not only fully integrated with Zowe Explorer, but also with Zowe CLI via free Plugin provided with Wazi for VS Code.
- Consistent command line parameters with equivalent z/OSMF commands.
- Wazi for VS Code samples show how to write automation that can use either protocol interchangeably.
- Additional commands for added value features as shown in this presentation.



The screenshot shows a web browser window displaying the Zowe CLI help page for the `rse-api-for-zowe-cli` plugin. The browser's address bar shows the file path `/Users/phaumer/.zowe/web-help/index.html?p=zowe_rse-api-for-zowe-cli`. The page has a sidebar on the left with a tree view of the Zowe CLI command structure. The `rse-api-for-zowe-cli` command is selected, showing its sub-commands: `auth`, `cancel | can`, `change`, `check`, `common-properties | cm`, `copy`, `create | cre`, `delete | del`, `download | dl`, `export`, `issue`, `list | ls`, `migrate | hmigr | hMigrate`, `recall | hrec | hRecall`, `rename | rn`, `search | sch`, `submit | sub`, `upload | ul`, `view | vw`, `zos-console | console`, `zos-files | files`, `zos-jobs | jobs`, and `zos-logs | logs`. The main content area on the right is titled `zowe > rse-api-for-zowe-cli` and contains the following text:

Welcome to the IBM RSE API Plug-in for Zowe CLI!

Usage

`zowe rse-api-for-zowe-cli <group>`

Where <group> is one of the following:

Groups

- [auth](#) - Connect to RSE API Server Authentication Service
- [cancel | can](#) - Cancel a job
- [change](#) - Change User Data on the Host
- [check](#) - Confirm that RSE API is running on a specified system and gather information about the RSE API server and Address Space for diagnostic purposes, and information about user password expiry. Obtain and update MVS mappings file
- [common-properties | cm](#) - Manage Common Properites
- [copy](#) - Copy a z/OS UNIX System Services file/directory or data set
- [create | cre](#) - Create data sets
- [delete | del](#) - Delete a data set or Unix System Services file
- [download | dl](#) - Download content from z/OS data sets, UNIX System Services files, and JOB Output to your PC.
- [export](#) - Export Z/OS files.
- [issue](#) - Issue TSO and UNIX commands
- [list | ls](#) - List the details for data sets and the members in the data sets, z/OS UNIX System Services files and directories, and jobs and spool files.
- [migrate | hmigr | hMigrate](#) - Migrate data sets.
- [recall | hrec | hRecall](#) - Recall migrated data sets.
- [rename | rn](#) - Rename a Data set, Data set member, File or Directory
- [search | sch](#) - Search for Unix System Services Files or Data Sets
- [submit | sub](#) - Submit z/OS jobs
- [upload | ul](#) - Upload the contents of a file to z/OS data sets or to z/OS UNIX System Services
- [view | vw](#) - View details of a z/OS job

@zowe/cli 7.28.2

Content Search with Z Open Editor

z/OS Resources table and RSE

- Z Open Editor’s z/OS Resources table can be used with z/OSMF and RSE API, but special features with RSE such as Search
- Filter and Search for content in your Sequential or Partitioned Data Sets
- RSE API Plugin for Zowe CLI also provides additional Search on UNIX files
- Docs: https://ibm.github.io/zopeneditor-about/Docs/advanced_resource_table.htm

RSE API CLI Plugin UNIX Search example.

```
1 $ zowe rse search uss /u/user "*.txt" --t "hello world!"
2
3 Found 2 results for search:
4
5 /u/user/test/examples/example.txt -- 2 matches:
6
7 1) "hello world! ..."
8   Line: 1 -- Start Offset: 1 -- end Offset: 17
9
10 1) ".. hello world!"
11   Line: 5 -- Start Offset: 23 -- end Offset: 40
12
13 /u/user/test/examples/example.txt -- 1 match:
14
15 1) "hello world! ..."
16   Line: 13 -- Start Offset: 12 -- end Offset: 30
```

Searching in Sequential Data Sets

← → ↗ 🏠 IBMUSER.* 🔍 🔄 +

^ ☐	Name	Block Size	Organization	Search for sequential data set content (search scope is limited by active filters)	Set Type
^ ☐	IBMUSER.SAMPLE.CUSTFILE	604	PS	RALEIGH	/SAM ⋮
3 matches for 'RALEIGH'					
	1 00001AIBM	000445599000254205	S MIAMI BLVD--	RALEIGH-----	NCUSA-----
	2 00002AWIDGET MAKERS	000002599000064206	S MIAMI BLVD--	RALEIGH-----	NCUSA-----
	4 00003ACAFE RTP	000003399000064208	S MIAMI BLVD--	RALEIGH-----	NCUSA-----
^ ☐	IBMUSER.SAMPLE.CUSTOUT	604	PS	SEQ	NONVSAM ⋮
2 matches for 'RALEIGH'					
	1 00001AIBM-RTP	000445599000254205	S MIAMI BLVD--	RALEIGH-----	NCUSA-----
	2 00002AWIDGET MAKERS	000002599000064206	S MIAMI BLVD--	RALEIGH-----	NCUSA-----

Submit 🔍

Search for sequential data set content
(search scope is limited by active filters)

Searching in Partitioned Data Sets

← → ↗ 🏠 BSANTOS.* 🔍 🔄 +

^ ☐	Name	Search type:
^ ☐	BSANTOS.SAMPLE.COBOL.NEW(SAM1)	PDS Member ⌵
Search for members matching pattern		
4 matches for 'Terminating'		
	401 DISPLAY 'Terminating Progi	
	408 DISPLAY 'Terminating Progi	
	414 DISPLAY 'Terminating Progi	
	420 DISPLAY 'Terminating Progi	
Search for data set members containing content		
Terminating Ra		
Submit 🔍		
^ ☐	BSANTOS.SAMPLE.COBOL.NEW(SAM1LIB)	
4 matches for 'Terminating'		
	347 DISPLAY 'Terminating Program due to File Error'	
	354 DISPLAY 'Terminating Program due to File Error'	
	360 DISPLAY 'Terminating Program due to File Error'	
	366 DISPLAY 'Terminating Program due to File Error'	
^ ☐	BSANTOS.SAMPLE.COPYLIB(SAM1)	

Search type:

PDS Member ⌵

Search for members matching pattern

SAM*

Search for data set members containing content

Terminating Ra

Submit 🔍



Fine granular RSE EBCDIC mappings

- Retrieving MVS files via Zowe for VS Code converts them UTF-8.
- Zowe profile provide one central way to define MVS EBCDIC encoding for all files.
- RSE API provides fine-granular mappings using data set name patterns.
- Mappings can be defined
 - Globally on RSE API server
 - Project-level using ZAPP files
 - User-level managing personal mapping files
- Available in Z Open Editor and Developer for z/OS.
- Docs: https://ibm.github.io/zopeneditor-about/Docs/ebcdic_encoding.html

Zowe CLI Profile with a single default

```
1 {
2   "profiles": {
3     "rse": {
4       "type": "rse",
5       "properties": {
6         "port": 6802,
7         "basePath": "rseapi",
8         "protocol": "https",
9         "encoding": "IBM-930"
10      },
11      "secure": ["user", "password"]
12    }
13  }
14 }
```

Z Open Editor ZAPP file specifying mappings

```
1 name: sam
2 version: 1.1.0
3
4 profiles:
5   - name: RSE-Mappings
6     type: rseapi
7     settings:
8       mappings:
9         - resource: "**.SAMPLE.COB*"
10           extension: cbl
11           transfer: text
12           encoding: IBM-273
13           memberMappings:
14             - extension: dat
15               transfer: binary
16               resource: "*DAT"
17         - resource: "**PLI*"
18           extension: pl1
19           transfer: text
20           encoding: IBM-500
21           default.encoding: IBM-1047
```

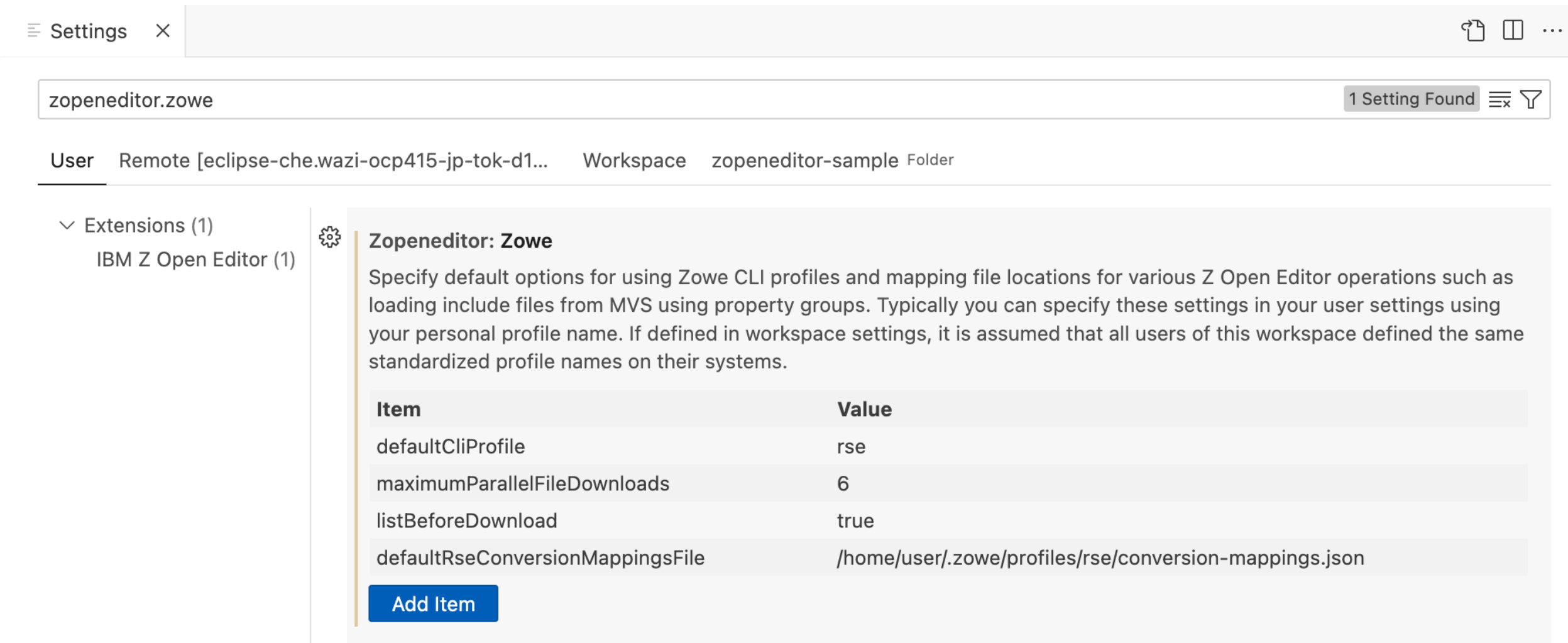
Mapping table in z/OS Explorer (Eclipse)

z/OS File System Mapping					
System: devtest1					
Mapping Criterion	Workstation File Extension	Transfer Mode	Host Code Page	Local Code Page	Priority
**MFS	mfs	text	IBM-037 (inherited)	UTF-8 (inherited)	1
**COBOL	cbl	text	IBM-037 (inherited)	UTF-8 (inherited)	2
**COBCOPY	cpy	text	IBM-037 (inherited)	UTF-8 (inherited)	3
**PLI	pli	text	IBM-037 (inherited)	UTF-8 (inherited)	4
**ASSEMBLE	asm	text	IBM-037 (inherited)	UTF-8 (inherited)	5
**OBJ	obj	binary	IBM-037 (inherited)	UTF-8 (inherited)	6
**LOAD	exe	binary	IBM-037 (inherited)	UTF-8 (inherited)	7
**CLIST	cmd	text	IBM-037 (inherited)	UTF-8 (inherited)	8
**JCL	jcl	text	IBM-037 (inherited)	UTF-8 (inherited)	9
**SIGYCLST	cmd	text	IBM-037 (inherited)	UTF-8 (inherited)	10
**CNTL	jcl	text	IBM-037 (inherited)	UTF-8 (inherited)	11
**LISTING	lst	text	IBM-1047	UTF-8 (inherited)	12
**OUTLIST	out	text	IBM-037 (inherited)	UTF-8 (inherited)	13
**INCLUDE	inc	text	IBM-037 (inherited)	UTF-8 (inherited)	14
**MACRO	mac	text	IBM-037 (inherited)	UTF-8 (inherited)	15
**REXX	rex	text	IBM-037 (inherited)	UTF-8 (inherited)	16



Important Z Open Editor settings

that affect RSE API differently than z/OSMF



z/OSMF

- Primarily created for maintenance activities
- Scales by creating reusable address spaces
- Editor running many parallel downloads for retrieving copybooks create many address spaces.
- Not finding a file results in errors in system logs. Avoided by internally running "list" command before trying to open file. One extra server round-trip.
- One central conversion setting in Zowe profile.

Remote System Explorer API

- Primarily created for development use cases
- Scales by creating load balancing server groups
- Efficient resource usage when downloading many copybooks in parallel.
- Separate logs configuration.
- User can share conversion mappings or use personal settings.

Documented REST API

- Write your own applications in Java, Typescript or any other language accessing z/OS via RSE API.
- Docs: <https://www.ibm.com/docs/en/developer-for-zos/16.0?topic=interfaces-rse-api-sdk-overview>
- Java SDK Blog: <https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/dave-mcknight1/2022/10/19/new-remote-capabilities-for-java-developers-in-ibm>
- Annotated TS Type Lib: <https://www.npmjs.com/package/@ibm/rse-api-for-zowe-cli>

 **Swagger**
Supported by SMARTBEAR

swagger.json

Explore

RSE API Docs

v1.1.6 OAS3

swagger.json

IBM Remote System Explorer API (RSE API) is a collection of REST APIs that allow a client to work with various components on the z/OS host system, including MVS data sets, z/OS UNIX files and commands, JES jobs, TSO commands, and more.

Servers

/rseapi/api/v1

Authorize

Authentication

JES Jobs

MVS Datasets

UNIX Files

UNIX Commands

Scan Service

TSO Commands

Administrator Commands

Server Information

Common Properties

Schemas



Manage your system's meta-data with RSE API Name/Value store

- RSE API provides a simple document database that allows storing JSON objects.
- Provides put and get methods.
- Use for CLI automation or with SDKs to store system specific meta-data needed in your organization.

CLI command for creating a namespace

```
1 % zowe rse common-properties create ns1
2 Namespace created successfully.
```

Storing a document in namespace

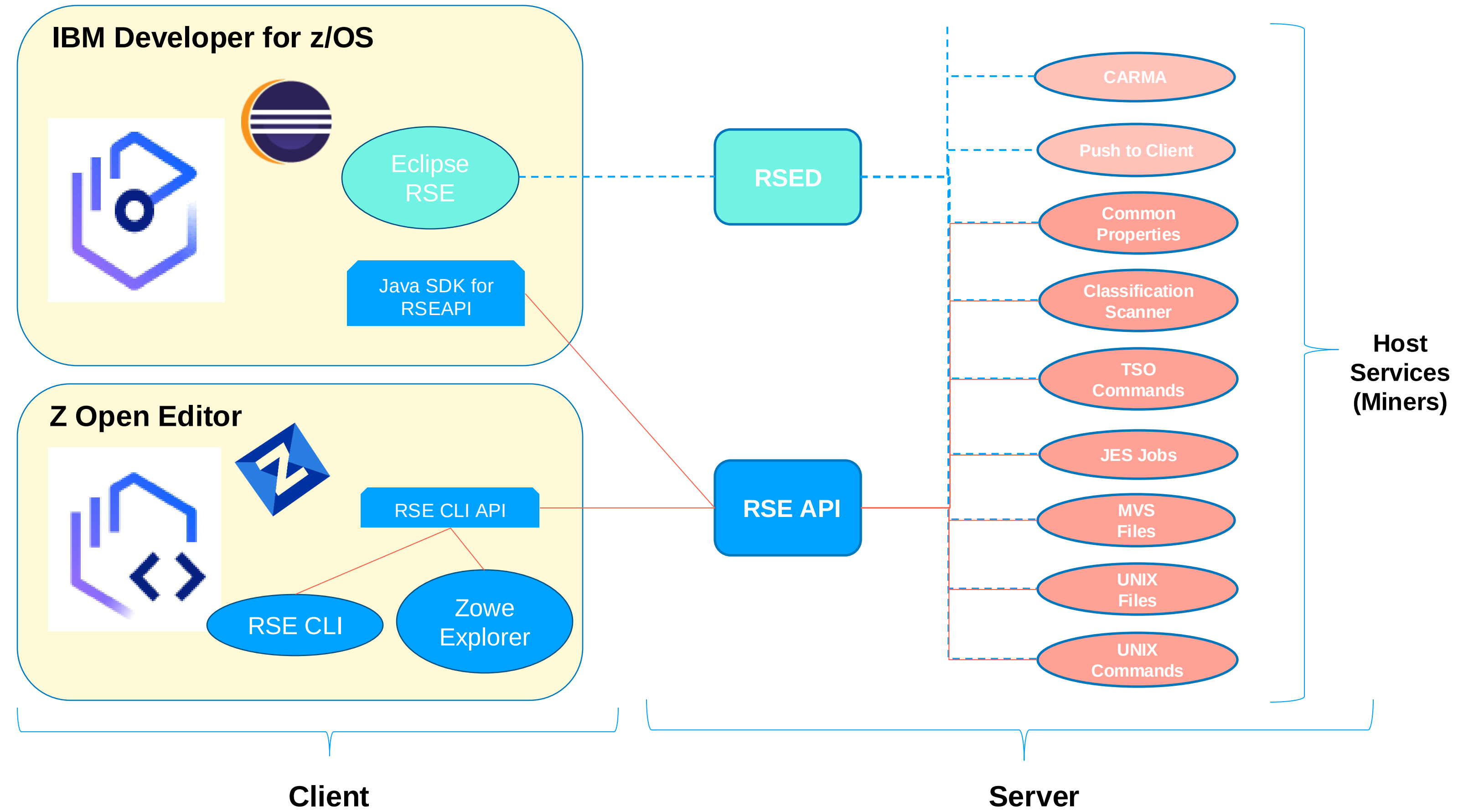
```
1 $ zowe rse common-properties set ns1 /test-server '{"hostname" : "www.test.com", "port" : 1234}' --type json
2 Property set successfully.
```

Retrieving content

```
1 $ zowe rse common-properties view ns1
2 {
3   "test-server": {
4     "hostname": "www.test.com",
5     "port": 1234
6   },
7   "prod-server": {
8     "hostname": "www.prod.com",
9     "port": 443
10  }
11 }
```

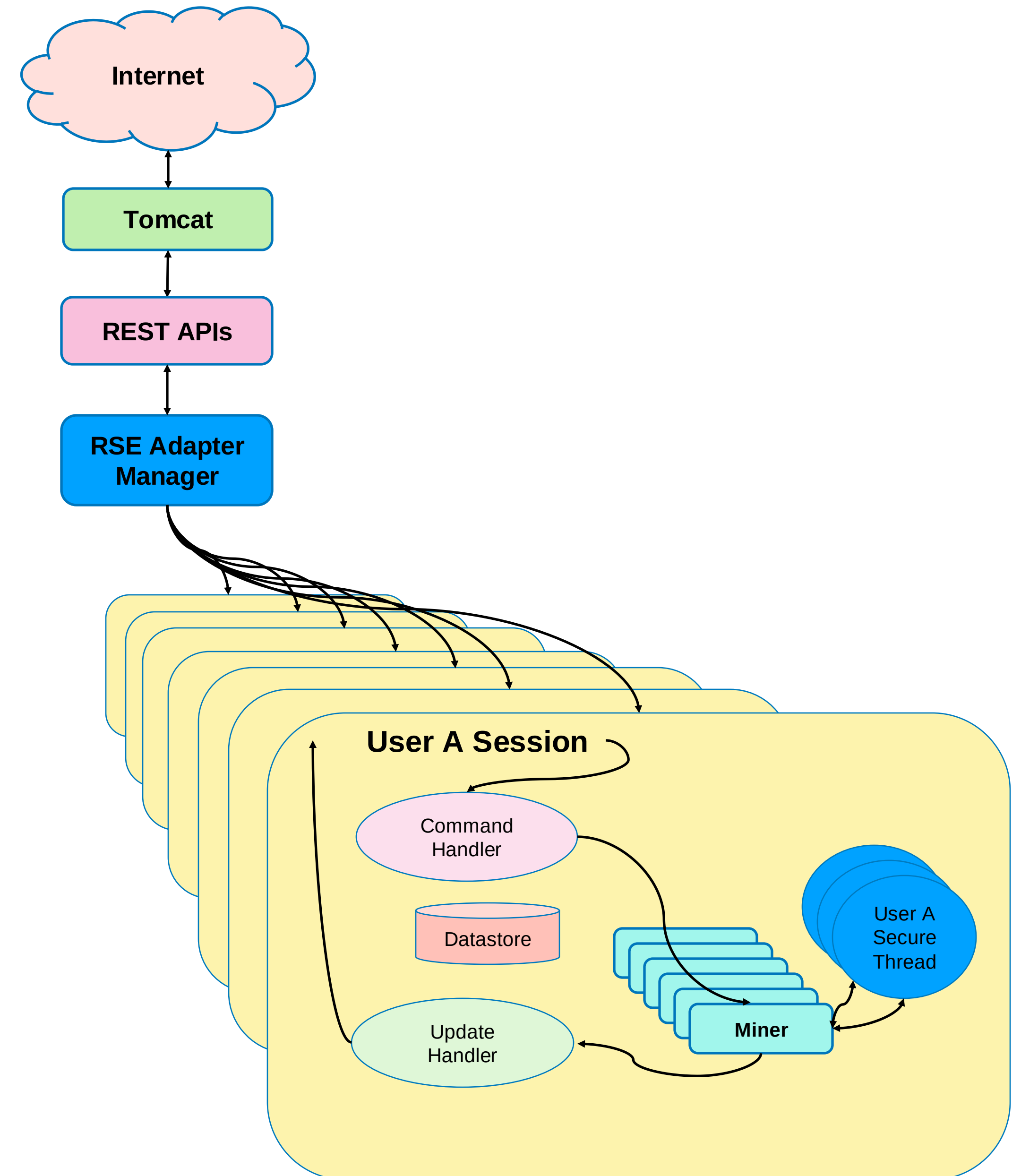

RSE Architecture

- RSED (used by Idz) provides a TCP communication layer and a framework that facilitates routing of commands to miners (host services) and propagating information back to clients
- RSEAPI uses the same underlying host service architecture with some differences:
 - Requests come over https via REST API calls
 - An adapter layer manages user sessions facilitating interactions with miners



RSE Architecture and API Servers

- RSEAPI is a servlet application running on Tomcat
- Provides REST endpoints for a variety of services
- An adapter layer is used to integrate with RSE host framework (Dstore)
 - Under the covers, after logging in, each user has their own RSE session
 - Includes a data store for managing data, command handler, update handler and miners (just as is provided for RSED)
 - API calls result in commands that get handled by miners, which use secured threads running under the user id



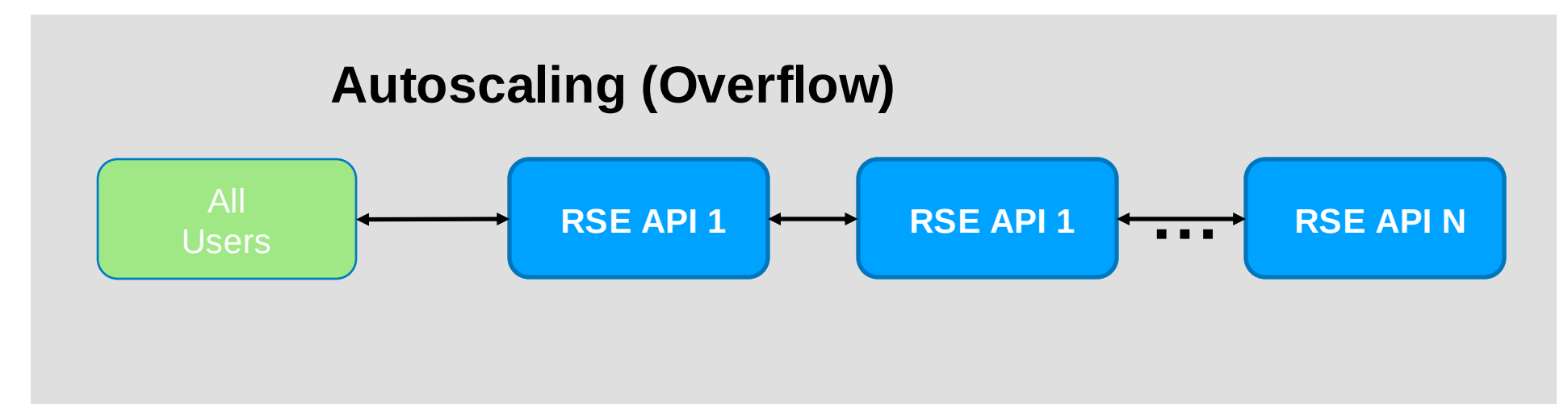
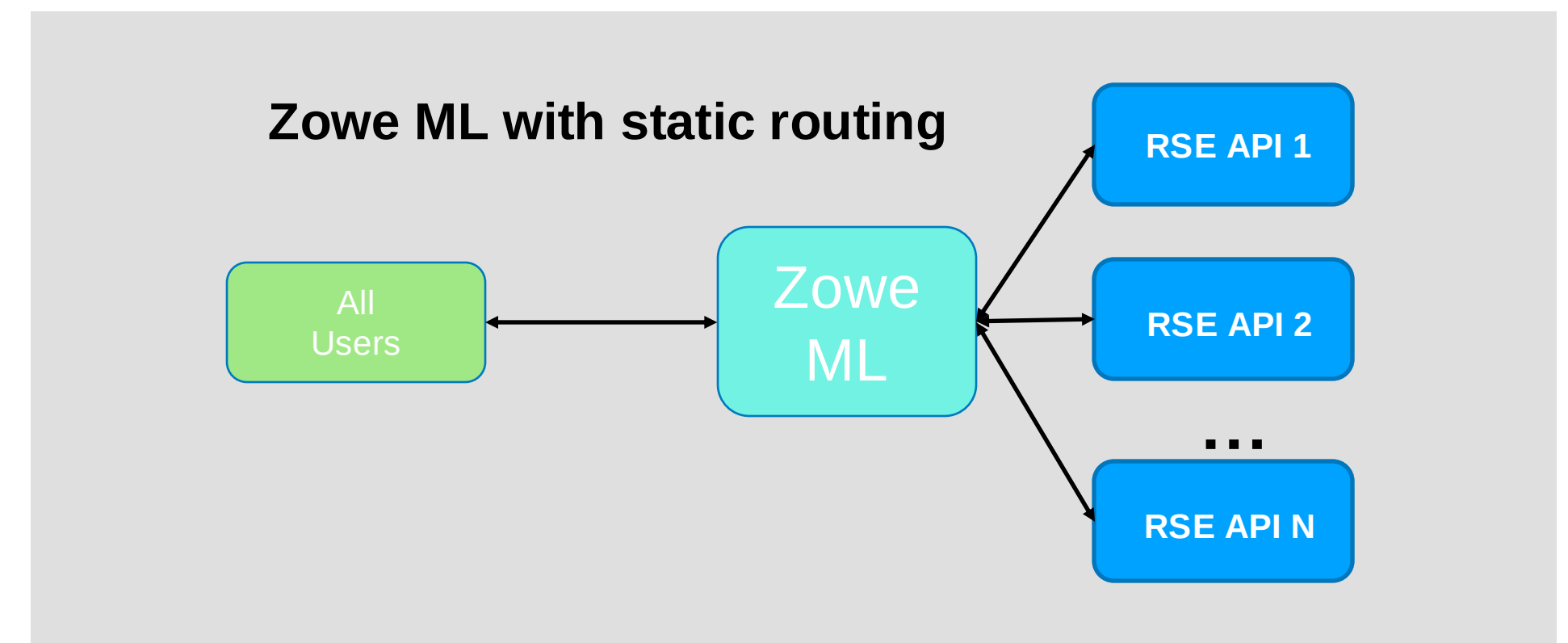
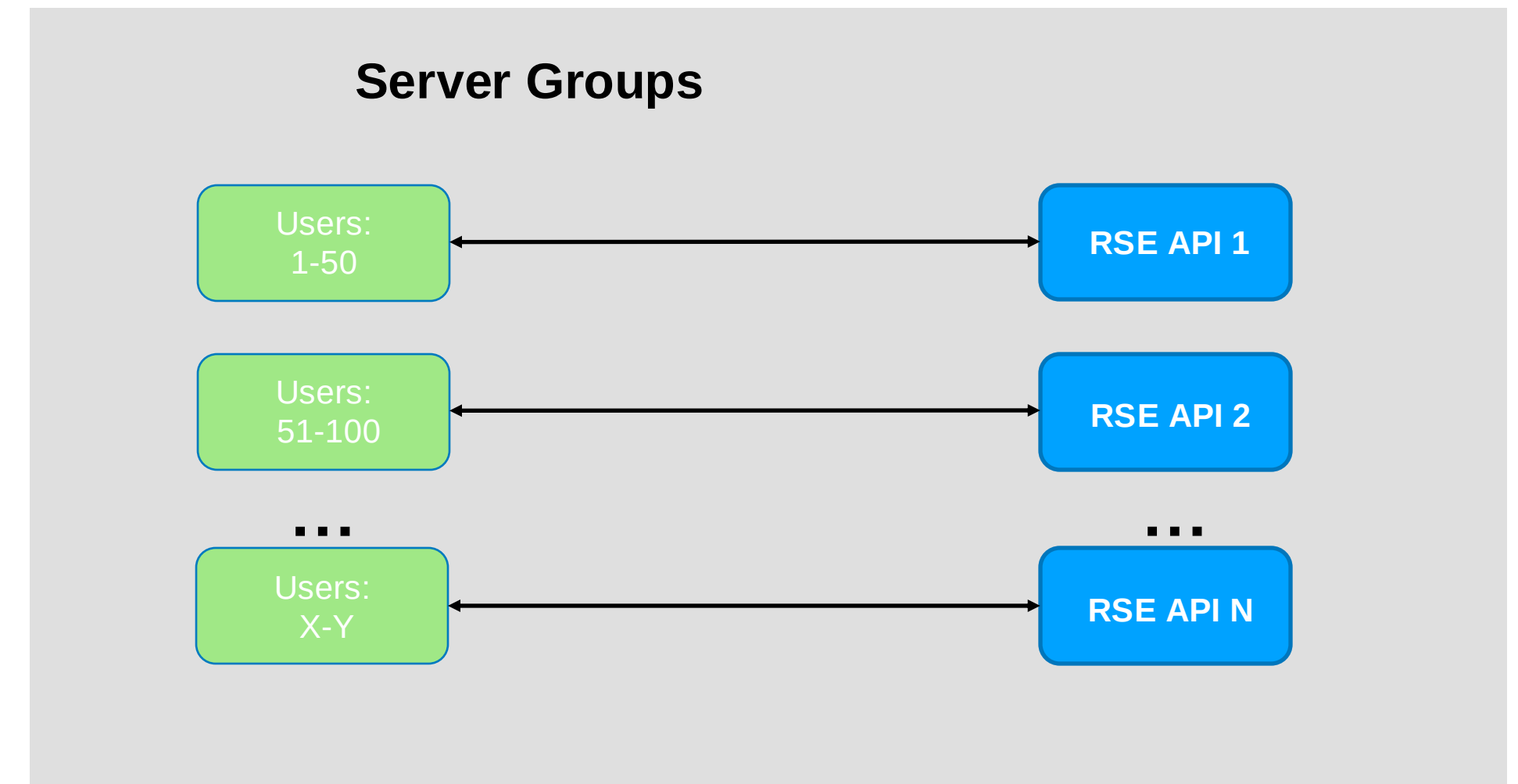
RSEAPI Scaling and Server Management

Server Capacity

- Each user session in RSEAPI includes a set of dedicated threads as well as in-memory data
- Large numbers of users on the same server can mean lots of threads and memory usage
- Applications on z/OS have limits on total memory used
 - As a result, we recommend capping the maximum number of concurrent users on a single RSEAPI instance ~50

Scaling

- Multiple instances of RSEAPI used to accommodate large numbers of users
- Recommended ways to deploy for scale
 1. **Manually:** deploy multiple servers, assigning groups of users to dedicated ports
 2. **Zowe ML (with static routing):** onboard multiple RSEAPI instances under the Zowe ML load balancer and configure Zowe ML to use static routing
 3. **Autoscaling:** configure RSEAPI to use the 'Overflow' mechanism. Additional server instances automatically deployed as needed



Try it...

IBM Z Open Editor

<https://ibm.github.io/zopeneditor-about>

and provide feedback

Try our free Eclipse Che
plugin and devfile

<https://github.com/IBM/zopeneditor-about/tree/main/che>

Peter Haumer
phaumer@us.ibm.com
<https://www.linkedin.com/in/haumer>

Learn more about

Wazi Code

<https://www.ibm.com/docs/en/cloud-paks/z-modernization-stack/2024.2?topic=develop-zos-applications-wazi>

Architecture deep-dive and demo:

<https://youtu.be/8Z0GPN7Ld2w>

IBM zSystems and Cloud Modernization Stack

<https://www.ibm.com/products/z-and-cloud-modernization-stack>

IBM Wazi as a Service

<https://www.ibm.com/cloud/wazi-as-a-service>

Demo Video Playlist:

https://mediacenter.ibm.com/playlist/details/1_5q5qeehb/

z/OS Development in the IBM Cloud Demo:

<https://youtu.be/OdsfPiOfKto>