



OPEN
MAINFRAME
PROJECT



Zowe

Tim Brooks (tim.brooks@ibm.com)
IBM System Z Associate Offering Manager

Dan Jast(Daniel.Jast@ibm.com)
Z Systems Technical Specialist
Client Experience Centers - IBM Systems

“Disclaimer”

- Zowe is an open source project - IBM was a founding member
- Zowe is not an IBM product
- All the information presented in this session is public information but subject to change
- Future plans depends on the open community decisions. Time frames for delivery are not guaranteed since it depends on the staffing and time commitment of the open community
- You are encouraged to get involved in the community for the latest information and to influence the project direction and priorities

Agenda



- Introducing Zowe
 - Zowe Vision Statement and Market Feedback
- What's in Zowe?
 - *Zowe REST Services*
 - *Zowe Web Desktop*
 - *Zowe CLI*
 - *Zowe API Mediation Layer*
- Example Zowe Commercial exploitation (so far)
- Three Example Zowe Integration Options
- Outlook for Zowe in 2019
 - Announcing Zowe Conformance Program
- Getting Started with Zowe

What is Zowe?



- An extensible software framework on z/OS for connecting primarily – but not limited to – system administrative and development tools to the mainframe.
- Aims to make the mainframe an integrated and agile platform within the changing IT architectural landscape.
- First open source project on z/OS. All code is licensed under the Eclipse Public License version 2.0

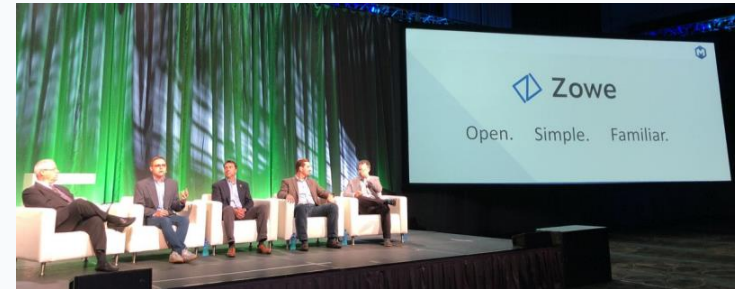


Zowe

Quick Facts about Zowe



- **Zowe 1.0.0 Announced at THINK 2019**
 - 100% Open Source (EPL 2.0)
 - Defined extensions points
 - Framework ready for commercial exploitation
- Pronounced as “Zoe” – [zoh-ee] in English
 - Not an acronym – just a simple, fun and easy name
 - Using the spelling “Zowe” allowed us to trademark
- An open source project under the Open Mainframe Project (OMP), a collaborative project within the Linux Foundation
 - IBM, Rocket Software and CA Technologies (now Broadcom) are founding members





Zowe Vision Statement

- Reduce the learning curve
 - ✓ Improve productivity
 - ✓ Modern, platform-neutral interfaces
 - ✓ Cloud-like experience
- Attract new people
 - ✓ Demystify the Z platform
 - ✓ Enhance integration and consumability
 - ✓ Promote Open community of practice
- Simplify architecture
 - ✓ Reduce operational overhead
 - ✓ Improve co-existence
 - ✓ Enable rich ecosystem of free and commercial solutions



Quote from a new System Programmer at Large US-based Insurance Company

“.....I’m 25 years old and have only been in mainframe for about a year. After seeing Zowe, I wish this was the standard when it comes to interacting with mainframe instead of TN3270. It’s learning curve is *vastly* less intimidating and it makes JCL look so much better and less archaic. **I’m honestly thrilled to use this. Total game changer.**”

Zowe accelerates digital transformation through API-enabled services



- z/OSMF
- TSO
- ISPF
- USS
- Rexx
- JCL
- Product Interfaces

- Dataset API
- Job API
- USS API
- System API



Swagger-defined REST APIs

API Discovery - APIs available from the API Discovery feature

Alias	Dataset APIs	Operations
DELETE	/dataset/apis/dataset/{id}	Delete a data set or member
POST	/dataset/apis/dataset	Create (and populate) a data set
GET	/dataset/apis/dataset/{id}/attributes	Retrieve attributes of a data set
GET	/dataset/apis/dataset/{id}/content	Read content from a data set or member

API Mediation Layer

API Mediation Layer API

The API Mediation Layer for z/OS Internal API services. The API Mediation Layer provides a single point of access to maintain REST APIs and other enterprise cloud features such as high-availability, scalability, dynamic API discovery, and documentation.

```

apicatalog
API Catalog
API Homepage
API Catalog service to display service details and API documentation for discovered API services.

API Catalog
API Version: 1.0.0
[? www.ibm.com/zos/1.0.0/notes.html ]

REST API for the API Catalog service which is a component of the API Mediation Layer. Use this API to retrieve information regarding catalog dashboard files, file contents and its status, API documentation and status for the registered services.

Swagger/OpenAPI JSON Document

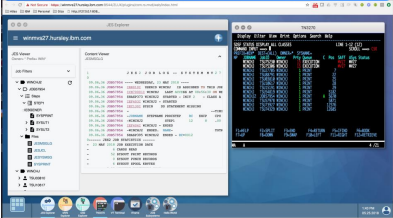
API Catalog: Current state information
GET /containers Lists catalog dashboard files
GET /containers/{id} Retrieves a specific dashboard file information.

API Documentation: Service documentation
    
```



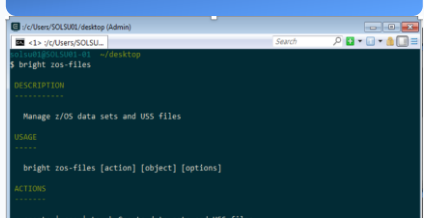
- MVS Explorer
- USS Explorer
- JES Explorer
- TN3270
- Workflow editor

Browser-based Virtual Desktop



Development and Operations Teams
(Internal/IT Facing)

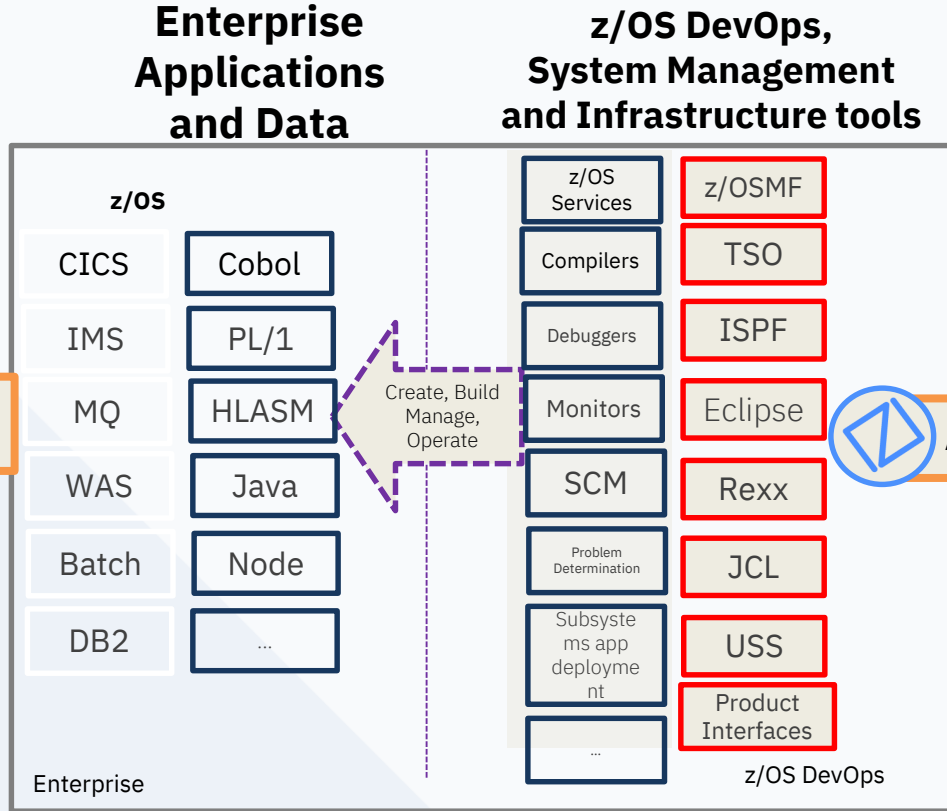
Node.js-based Cmd Line Interface



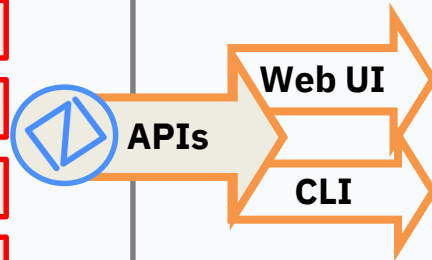
- zos-files
- zos-console
- zos-jobs
- zos-tso
- zos-workflow



**REST API
Enablement
of Line of
Business
Applications is
not Zowe's
Focus**



APIs, Web UI, CLI for Development and Operations



Development and Operations Teams
(Internal/IT Facing)



Zowe V1.0.0

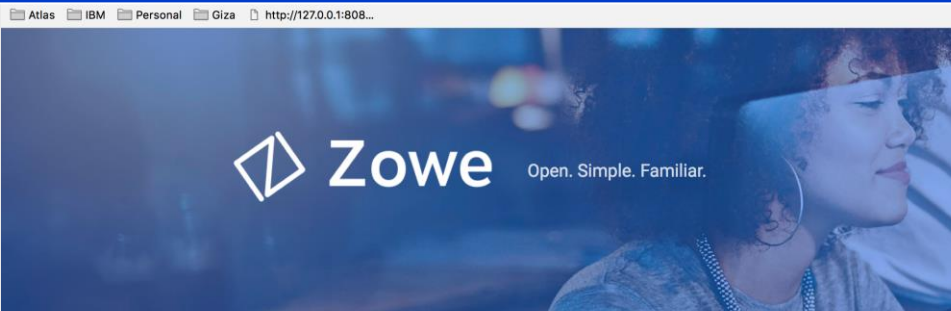
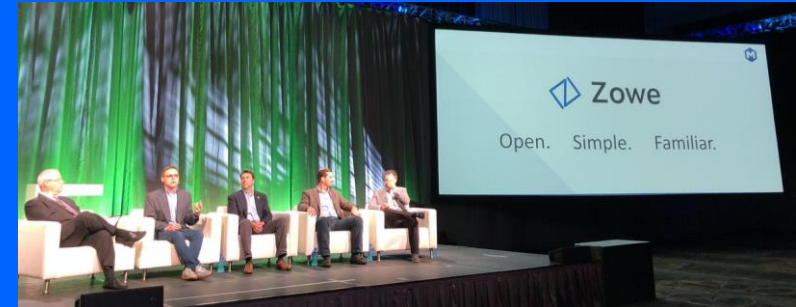
GA Feb 7th 2019

100% Open Source under EPL 2.0 license

Zowe V1.4 released August, 2019



Since Announce



Why Zowe?

With the ever-increasing speed of technology advancements, younger workforce and rising expectations in user experience, we need more intuitive, more productive and better integrated capabilities for z/OS.



With Zowe, we can enable delivery of such an environment

1,100+

Community Slack
Members Engaged

20+

Apps, plug-ins and
extensions

50+

Active committers.

4,432

Downloads
Beta + GA



“Best DevOps Open Source Project” - 2018 Finalist

- DevOps.com



“**Zowe** is a **truly powerful**, open collaboration initiative helping businesses experience the full benefit of digital transformation through a dynamic hybrid IT architecture inclusive of the mainframe.”

-IT Toolbox



“With **Zowe**, new possibilities are opening for **next generation applications**.”

-ZDNet



“Overall, we see **Zowe** as the first element of a scenario that will enable the mainframe to walk, then run, **then sprint into the hybrid cloud environments of the future**.”

-Clabby Analytics



Client Responses



“I’m **super excited** about this effort...

Awesome!”

- Large Retail and e-Commerce Company

“**Zowe** and the Open Mainframe will drive how the companies will **bring the Mainframe to the most modern solutions of software development.**”

- Large National Bank

“When I realized what the three companies are planning to provide, I about **fell out of my chair.**”

- Large National Bank

“Seeing CA and IBM working together on this is the **most exciting thing we have seen** in a long time.”

- Large International Bank

Universities seeking involvement: Western University, San Jose State University, Marist College, Virginia Commonwealth University (VCU), a major Turkish University, a major China University (Dalian University of Technology(DLUT))



"We have observed, throughout our 40 years in business, that approximately every decade or so a transformative technology comes along that greatly enhances the usability of our platform," said Ed Jaffe, Chief Technology Officer.

"We believe Zowe is this decade's transformative technology and it would be a mistake to not embrace it."

<https://phoenixsoftware.com/news/20190806.htm>

What's in Zowe?



Browser-based Web Desktop

The screenshot shows the JES Explorer web interface. On the left, there's a sidebar with 'swagger' and 'Api Document' sections. The main area is divided into 'JES Viewer' and 'Content Viewer'. The 'JES Viewer' shows a tree view of jobs, including 'WINCHJ2', 'JOBS7954', and 'STEP1'. The 'Content Viewer' displays a job log for 'JES2 JOB LOG -- SYSTEM MV37'. A terminal window on the right shows a command prompt with various system messages and job statistics.

API Mediation Layer (Gateway, Discovery Service, Catalog)

API Mediation Layer API

The API Mediation Layer for z/OS internal API services. The API Mediation Layer provides a single point of access to mainframe REST APIs and offers enterprise cloud-like features such as high-availability, scalability, dynamic API discovery, and documentation.

apicatalog

API Catalog

API Homepage

API Catalog service to display service details and API documentation for discovered API services.

API Catalog

API Version: 1.0.0

[Base URL: c83x.ca.com:10010/ap1/v1/apicatalog]

REST API for the API Catalog service which is a component of the API Mediation Layer. Use this API to retrieve information regarding catalog dashboard tiles, tile contents and its status, API documentation and status for the registered services.

Swagger/OpenAPI JSON Document

```
-----
USAGE

zowe <group>

Where <group> is one of the following:

GROUPS

config          Manage configuration and overrides
plugins         Install and manage plug-ins
profiles        Create and manage configuration profiles
provisioning | pv Perform z/OSMF provisioning tasks
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     Interact with TSO
zos-uss | uss     Issue z/OS USS commands and receive responses
zos-workflows | wf Create and manage z/OSMF workflows
zosmf           Interact with z/OSMF

OPTIONS
-----
```

Swagger-defined z/OS REST APIs

swagger

Api Document

[Base URL: winvms27.hursley.ibm.com]

Api Documentation

Terms of service

Apache 2.0

JES job APIs

Jobs Controller

GET /api/v1/jobs

GET /api/v1/jobs/{jobName}

DELETE /api/v1/jobs/{jobName}/{jobId}

GET /api/v1/jobs/{jobName}/{jobId}/files

GET /api/v1/jobs/{jobName}/{jobId}/files/{fileId}/content

GET /api/v1/jobs/{jobName}/{jobId}/steps

POST /api/v1/jobs/dataset

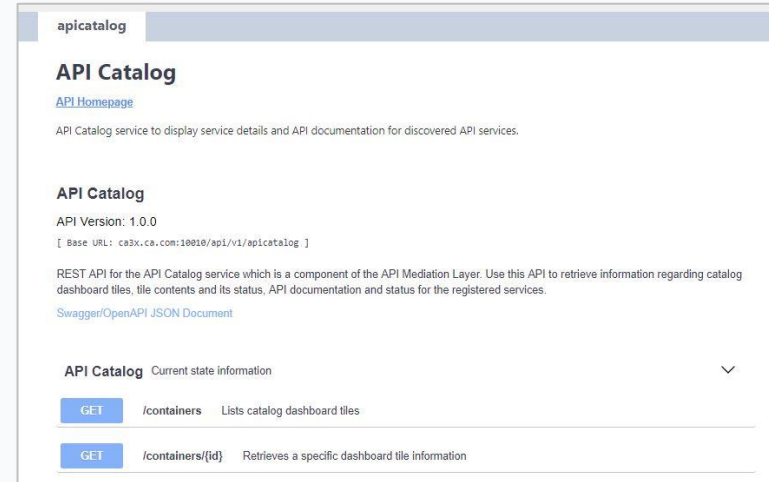
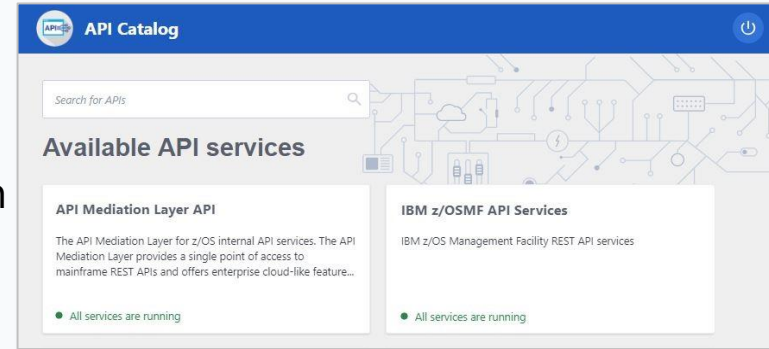
Node.js-based CLI



Zowe API Mediation Layer – Gateway to mainframe APIs

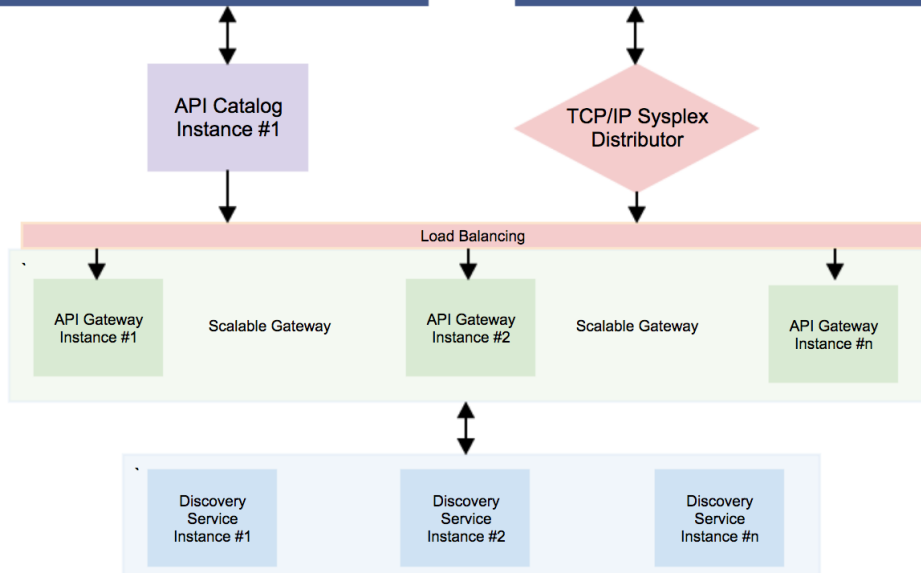


- Enables a single point of access to mainframe APIs with high-availability, scalability, dynamic API discovery, consistent security, “one-time” sign-on experience and unified standard API documentation (OpenAPI / Swagger)
- **API Catalog**
UI Catalog of available APIs with their Swagger doc and service status
- **Gateway**
Single secure point of entry to an ecosystem of API services. Hides complexity. Highly available. Based on Netflix Zuul.
- **Discovery Service**
Discover APIs across many applications. Repository of active API services. Based on Netflix Eureka.

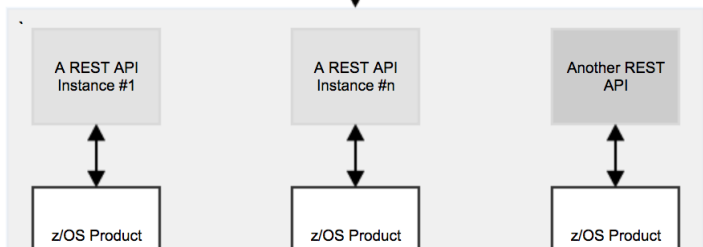


People Find APIs

Apps Call APIs



Dynamic Registration



z/OS Connect

z/OSMF

Jax-RS App

API Layer Components*

- **API Catalog**

UI Catalog of available APIs with their Swagger doc and service status

- **API Gateway**

Single point of entry to an ecosystem of microservices. Hides complexity. Highly available. Based on Netflix Zuul.

- **Discovery Service**

Discover APIs across many applications. Repository of active services. Based on Netflix Eureka.

- **z/OSMF API**

Authenticate Zowe users with mainframe credentials

* Separate microservices, might be running as separate address spaces



Browser address bar: <https://winmvs3b.hursley.i...>

Browser tabs: Atlas, IBM, Personal, Giza, http://127.0.0.1:80...

API Catalog header with search icon and power button

Search for APIs

Available API services

API Mediation Layer API

The API Mediation Layer for z/OS internal API services. The API Mediation Layer provides a single point of access to mainframe REST APIs and offers enterprise cloud-like feature...

● All services are running

z/OS Datasets services

IBM z/OS Datasets REST services

● All services are running

My Server APIs

z/OS Jobs services

IBM z/OS Jobs REST services

● All services are running

z/OSMF services

IBM z/OS Management Facility REST services

● All services are running

API Catalog lists API servers on its “Southbound edge”

API servers can be **statically** defined through .yaml files or else REST API calls to the gateway

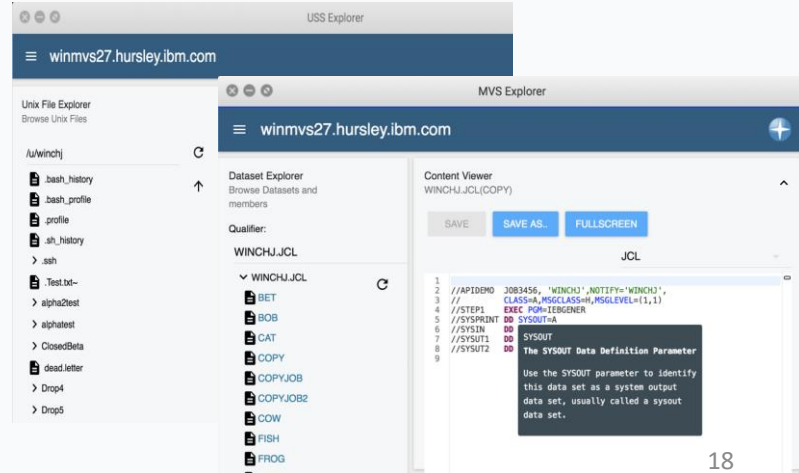
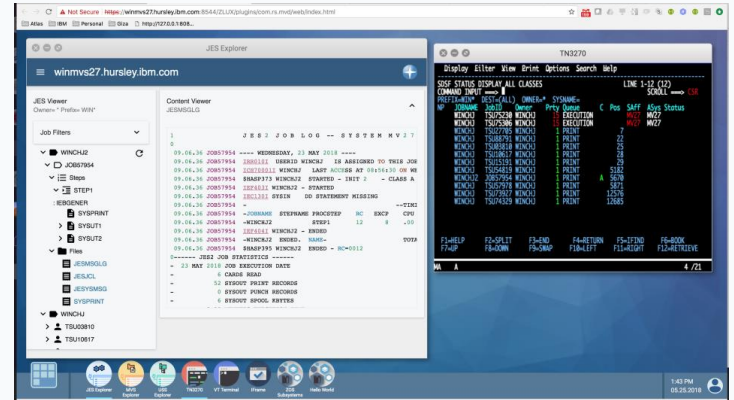
Dynamic discovery service based on Netflix Eureka framework



Zowe Web Desktop – An app container in a browser



- Known as **zLUX**, the Zowe Web UI is a virtual desktop system that offers a rich and open platform for a web-based mainframe user experience.
- **Mainframe Virtual Desktop**
 - A web-based window manager that provides full screen interactive experience
- **Zowe Node Server**
 - Runs zLUX; uses Express.js as web service framework for communication between applications and z/OS services and components, pre-reqs Node.js for z/OS
- **ZSS Server**
 - Provides secured REST API services
- **Application plug-in**
 - Dataservices, Configuration dataservice, URI broker, app-to-app communication, Error reporting UI, Logging utility
- **Explorers**
 - JES, MVS, USS explorers
 - Basic editing support for REXX and JCL





Zowe 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 and DB2, IMS, z/OS Connect, MQ,.....and Visual Studio

```
USAGE
-----

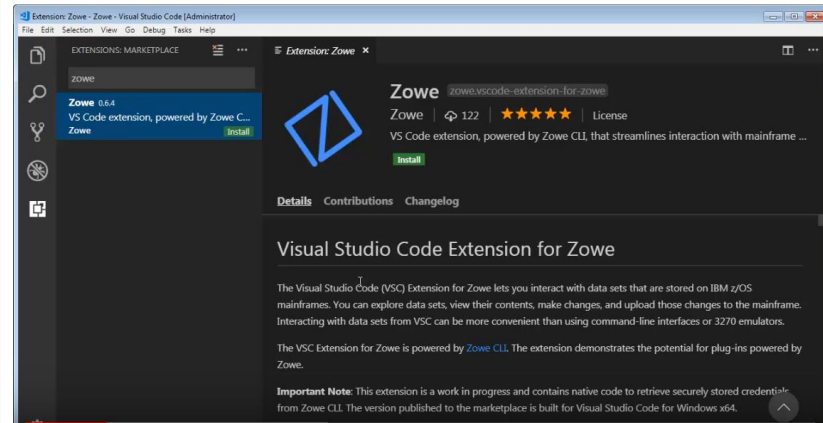
zowe <group>

Where <group> is one of the following:

GROUPS
-----

config          Manage configuration and overrides
plugins         Install and manage plug-ins
profiles        Create and manage configuration profiles
provisioning | pv Perform z/OSMF provisioning tasks
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      Interact with TSO
zos-uss | uss      Issue z/OS USS commands and receive responses
zos-workflows | wf Create and manage z/OSMF workflows
zosmf           Interact with z/OSMF

OPTIONS
-----
```



Zowe framework and ecosystem at a glance

Core Zowe includes API Mediation Layer, Command Line Interface (CLI), and Web User Interface

Vendors can build 'Zowe Compliant' plugins that will leverage this framework

Bold - Commercial Extensions

CLI Plug-in and Extensions

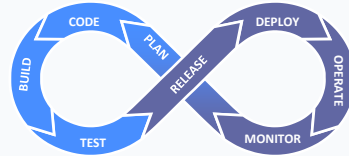
- CLI Plug-in for CICS
- CLI Plug-in for Db2
- Visual Studio Code Extension
- **Phoenix Software International (E)JES (Tech Preview) Wazi**
- **CA OPS MVS**
- **CA Workload Automation**
- **CA Sysview**
- **CA Endevor**
- **CA File Master**
- **CA Brightside**



Plugins



COMMAND LINE



WEB DESKTOP



Plugins

CLI: BUILD | DEPLOY

WEB DESKTOP

API MEDIATION LAYER

REST-ful APIs

Web Desktop Apps and Plug-ins

- JES Explorer
- Data Set Explorer
- UNIX System Services File Explorer
- Iframe, Angular and React Samples
- TN3270
- VT Terminal
- API Catalog
- Zowe Editor
- z/OSMF Workflow Editor
- Jupyter (separate download)
- **IBM OMEGAMON**
- **IBM Service Management Unite**
- **SMU**
- **IBM NetView (SOD)**
- **(Demo) Vicom Infinity Voice Assistant**
- **Db2 DevOps Experience**
- **Rocket Bluezone Web**

Zowe Core vs. Zowe Exploiters

• Zowe Exploiters

Exploiters are CLI plug-ins, web-UI extensions/apps, and new API's

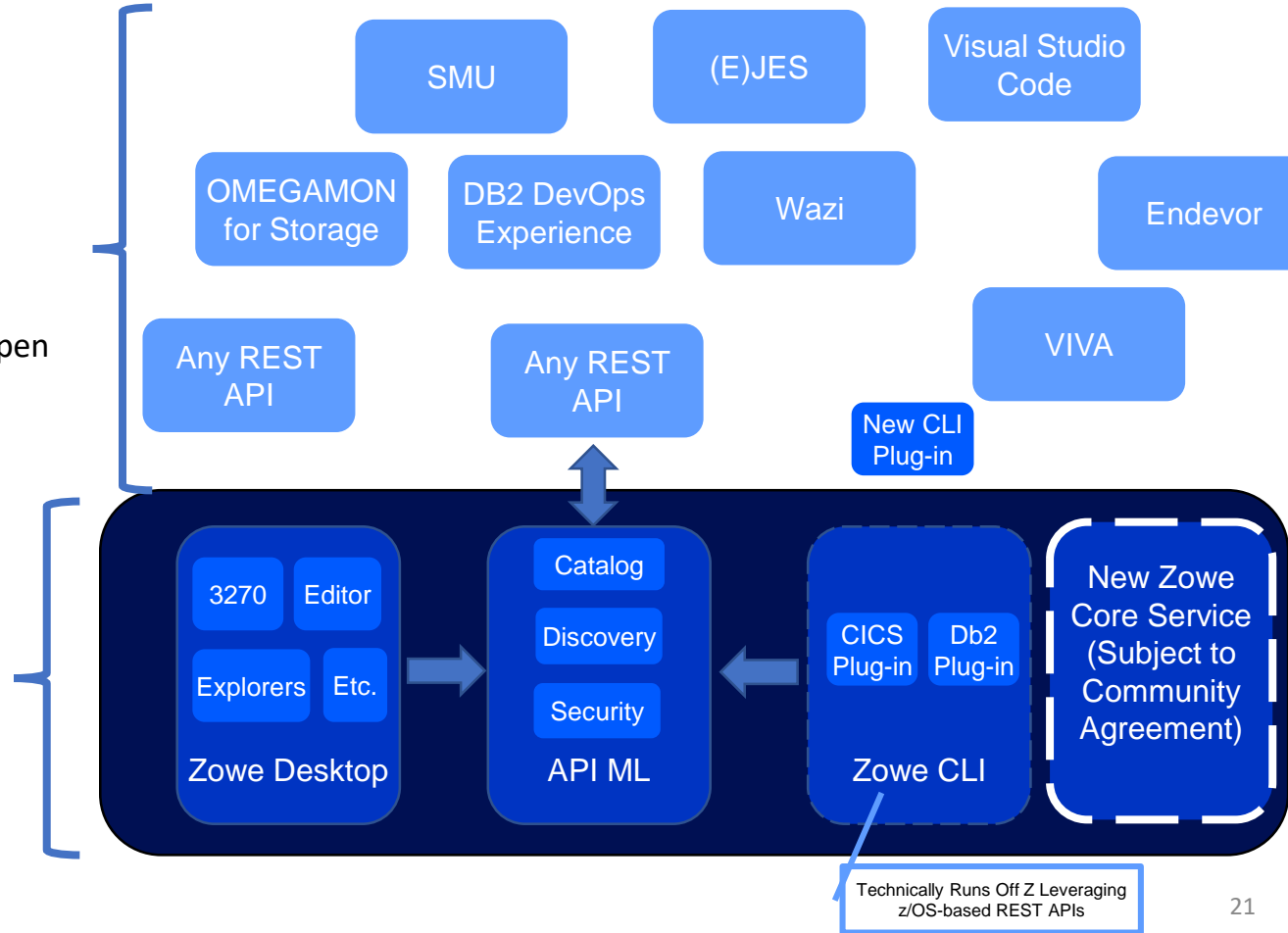
The exploiters use Zowe as infrastructure pre-req "runtime".

The extension may or may not be open sourced themselves.

• Zowe Core

Zowe Core includes the Zowe Desktop, Command Line Interface and the API mediation layer.

Core must be EPL2.0 license



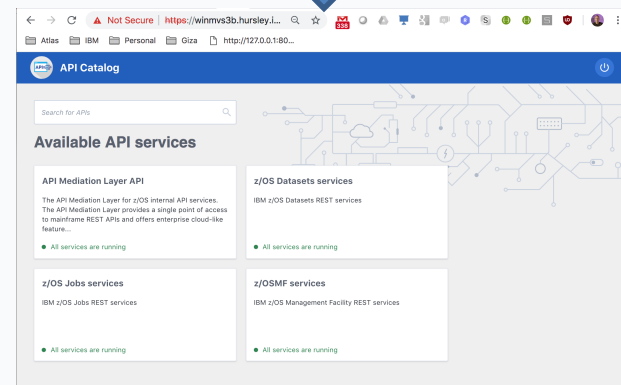
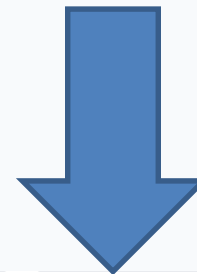
Integration Options



1. Your REST APIs into Zowe API catalog

- Whether you have swagger definition or not (Static or Dynamic API Discovery)
- <https://zowe.github.io/docs-site/latest/extend/extend-apiml/api-mediation-onboard-overview.html#overview-of-apis>
 - Example integration:
<https://developer.ibm.com/mainframe/2019/04/01/expose-z-os-connect-ee-apis-in-zowe-api-mediation-layer/>
- Catalog entry first step towards Single Sign On (SSO)
 - Work in progress
 - Teams converging of JSON Web Tokens (JWT)

APIs Can Be Easily Integrated with the Zowe REST API Catalog





Search for APIs



Available API services

API Mediation Layer API

The API Mediation Layer for z/OS internal API services. The API Mediation Layer provides a single point of access to mainframe REST APIs and offers enterprise cloud-like feature...

● All services are running

z/OS Datasets services

IBM z/OS Datasets REST services

● All services are running

Your services

Your APIs

z/OS Jobs services

IBM z/OS Jobs REST services

● All services are running

z/OSMF services

IBM z/OS Management Facility REST services

● All services are running

API Catalog lists API servers on its “Southbound edge”

API servers can be **statically** defined through .yaml files or else REST API calls to the gateway

Dynamic discovery service based on Netflix Eureka framework



DEMO

Integration Options



2. Create a CLI “plug in” for your services

- <https://zowe.github.io/docs-site/latest/extend/extend-cli/cli-devTutorials.html>
- <https://github.com/billpereira/zowe-cli-training>

CLI using z/OS-based REST APIs

```
GSE UK 2018 -- -bash -- 81x20
joes-mbp-3:GSE UK 2018 Joe$ zowe zos-console issue command "d a,l" | grep MSTR
OMEGRH OMEGRH RHAGENT NSW SO DJ27MSTR DJ27MSTR IEFPROC NSW S
DK27MSTR DK27MSTR IEFPROC NSW S DFHSM HSM DFHSM NSW S
QG27MSTR QG27MSTR PROCSTEP NSW S QJ27MSTR QJ27MSTR PROCSTEP NSW S
QM27MSTR QM27MSTR PROCSTEP NSW S IM3HDLI IM3HDLI IEFPROC NSW S
joes-mbp-3:GSE UK 2018 Joe$
```

Your Services here

```
GSE UK 2018 -- -bash -- 77x21
joes-mbp-3:GSE UK 2018 Joe$ zowe jobs list jobs
TSU45820 ABEND S222 WINCHJ OUTPUT
TSU45817 ABEND S222 WINCHJ OUTPUT
TSU45801 ABEND S222 WINCHJ OUTPUT
TSU45794 ABEND S222 WINCHJ OUTPUT
TSU45792 ABEND S222 WINCHJ OUTPUT
JOB45491 CC 0000 ZOWEDEMO OUTPUT
JOB40718 CC 0000 WINCHJL OUTPUT
TSU78332 ABEND S222 WINCHJ OUTPUT
TSU77652 ABEND S222 WINCHJ OUTPUT
TSU47736 ABEND S222 WINCHJ OUTPUT
TSU10038 ABEND S222 WINCHJ OUTPUT
TSU52101 WINCHJ ACTIVE
joes-mbp-3:GSE UK 2018 Joe$ zowe jobs list jobs -p Z*
JOB45491 CC 0000 ZOWEDEMO OUTPUT
joes-mbp-3:GSE UK 2018 Joe$
```

```
share2019 -- -bash -- 90x16
GROUPS
-----
cics          Interact with IBM CICS programs and transactions.
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
```

Desktop/laptop/cloud environment (Node.js app)



Integration Options

- 3. Have your existing (or new) web applications in the Zowe web UI
 - <https://zowe.github.io/docs-site/latest/extend/extend-desktop/mvd-creatingappplugins.html#setting-the-environment-variables-for-plugin-development>

The screenshot shows a web browser window titled "Zowe Desktop" with a URL of "https://winmvs3b.hursley.ibm.com:26508/ZLUX/plugins/org.zowe.zlux.bootstrap/web/". The main content area is a dark blue dashboard with a sidebar on the left containing icons for "TN3270", "USS Explorer", "MVS Explorer", "JES Explorer", and "Editor". A red callout box with the text "Your Apps in UI here" is overlaid on the dashboard, with two red arrows pointing to the central area.

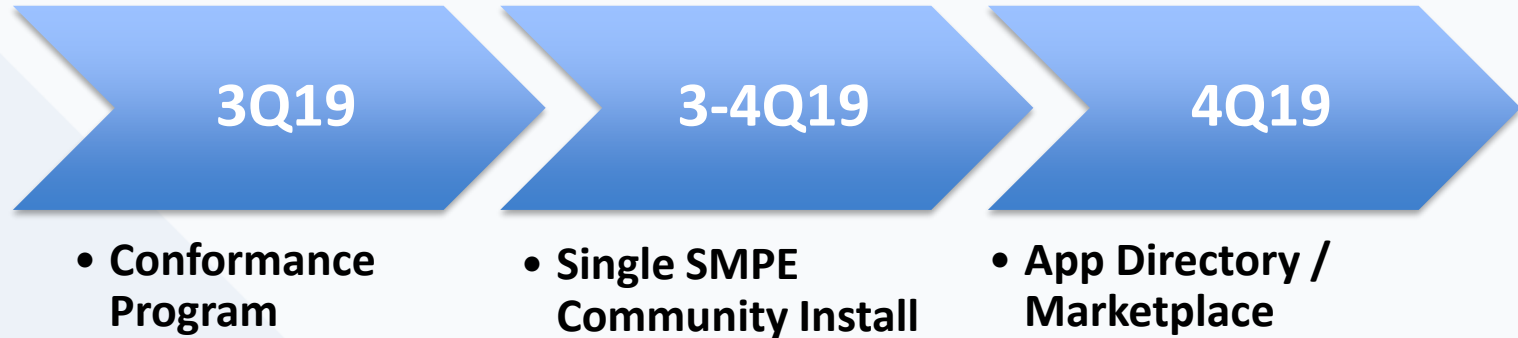
- Web UI apps also leveraging REST APIs
- UI participates in SSO
- App to app launch “in context” a key strength of the UI – breaking down product silos – more task oriented

Sample angular app and sample react app are tutorials for development

<https://github.com/zowe/sample-angular-app/branches> and <https://github.com/zowe/sample-react-app/branches>



2019 will define how you participate in the framework





What is Conformance?

- Opt-in guidelines for how to participate and build on top of the Zowe framework
- Achieve the goal of flexibility, integration and interoperability
- Play nice with others





Why Should I Care?

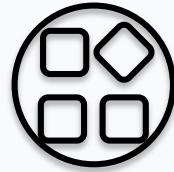
- As a **DEVELOPER** of Zowe Conformant Applications:
 - Ensures my application will behave as designed
 - Guidelines/direction on how to maximize value
 - No one else's application will break mine

- As a **CONSUMER** of Zowe Conformant Applications:
 - Leverages a common stack on infrastructure for multiple vendor apps
 - Confidence and transparency in how apps will operate
 - Conformant apps have been vetted by the community



Software Vendors can build Conformant offerings with Zowe

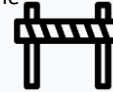
Make existing, or build new, products that are Zowe Conformant



Zowe distribution and services

Conformance Criteria

Conformance criteria will progress over time



Conformance Program



1. Prepare

Learn about the conformance program and the test evaluation guide



2. Run the Test

Follow instruction for how to test your application



3. Submit

Complete the terms and conditions and submit your request



4. Review

A reviewer will approve your request and you will be conformant



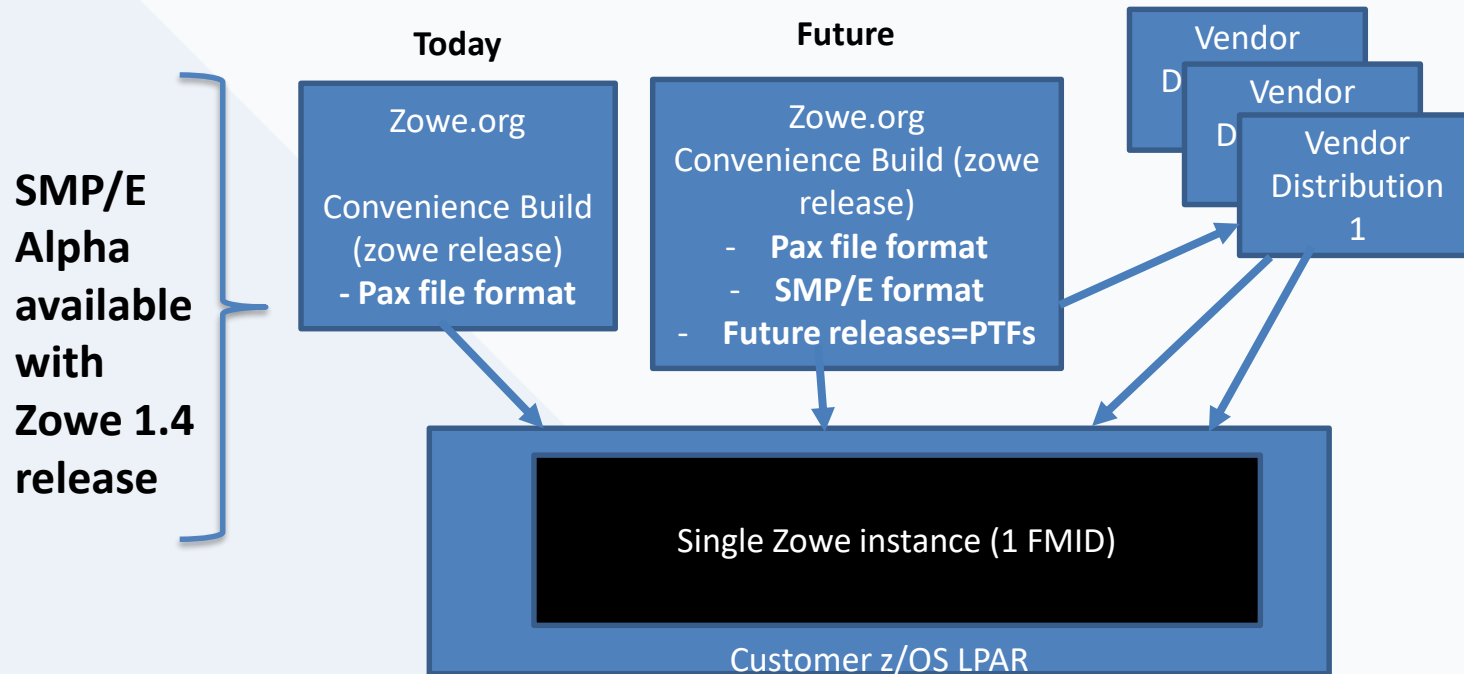
Pain points on Today's Zowe Install

- **No enterprise SMP/E installer**
 - Rollback, install history, customer production acceptance
- **Configuration data is specified at install time**
 - A single Zowe install cannot be started with different config parms
 - Configuration data lost on upgrade
- **Start up of Zowe brings up everything even if not all is needed**
 - Too many pre-reqs for Zowe
- **Install scripts fail if user doesn't have sufficient permissions**
- **Diagnosing problems difficult. Many runtime problems require re-install**



Packaging/install direction

- “Vendor Neutral FMID” for Single Zowe Instance
- We want the same customer experience regardless of distribution of the code



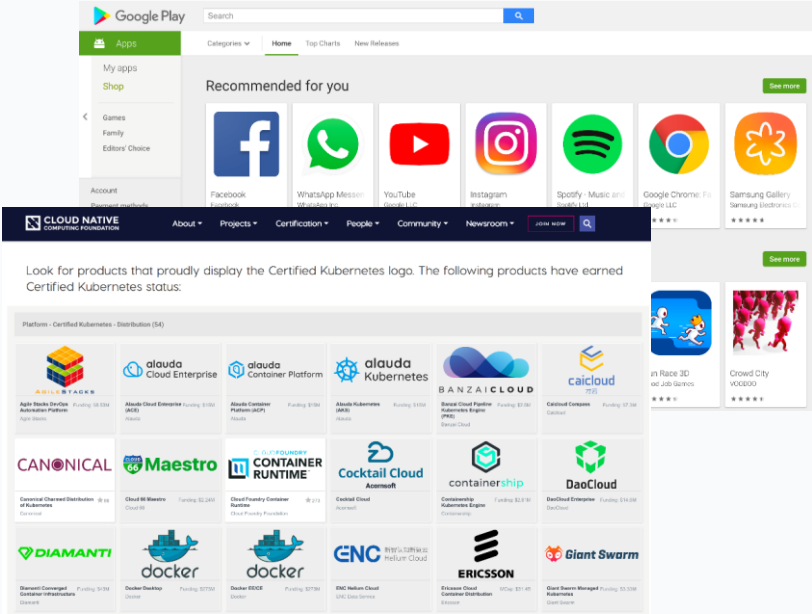


Conformant offerings/applications can be easily discoverable in the “App Directory”

Early stages of design

A directory of both free and commercial offerings that meet the conformance criteria (ensuring compatibility and interoperability)

More to come...



Why you and your organization should be a part of Zowe

Individual Contributors and Committer

- Exploit and learn new technology
- Build a global network
- Develop marketable skills through involvement in open source

Company Exploitation of Zowe Technology

- Quick adoption and new app development
- Improved integration and co-existence with other solutions
- Access to market place (future)

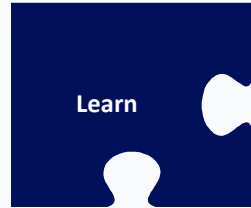
Company Investment in Developer Community

- Influence Zowe direction with contributions
- Expand reputation as industry leader
- Anticipate technology direction
- Reduced dependency on outside troubleshooting



Getting Involved

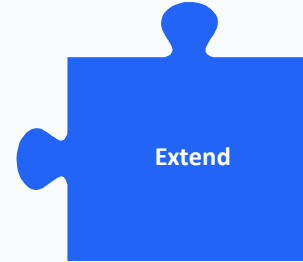
We are building more than just technology, we are building a **community**



- Visit the [Open Mainframe Project](#)
- Visit [Zowe.org](#)
- Connect with us on [Slack](#) or via [email list](#)
- Zowe on [zTrial](#) (under IT Operations)



- [Zowe Github](#)
- [Download Zowe](#)
- Review [documentation](#)
- [Troubleshooting](#)
- Reach out to us on [Slack](#) or via [email list](#)



- Review the [extenders guide](#)
- Zowe [Tutorials](#) and [Samples](#)



- Provide feedback, problems or recommendations to us on [Slack](#) or via [email list](#)
- Submit [Git](#) Issues
- Review the community [backlog](#) and contribute code
- Earn your committer status through [meritocracy](#)



Thanks

Intro Demo:

<https://www.youtube.com/watch?v=NX20ZMRoTtk&feature=share>

Visual Studio Code (using command line):

https://www.youtube.com/watch?v=la1_Ss27fn8

Program Terms and Conditions



Zowe Conformance Program Terms and Conditions

Definitions

In this document, these terms have the following meanings:

- **"Conformance Marks"** means (1) the word mark "Zowe Conformance", (2) the "Zowe Conformance" logo, and (3) all other logos and marks described in the Visual Branding Guide;
- **"Project Word Mark"** means the trademark "Zowe";
- **"Conformance Program"** means the conformance, testing and trademark usage program described in this document;
- **"Participant"** means a company or other entity that is permitted to use one or more of the Conformance Marks in association with their goods, services, pursuant to the Conformance Program;
- **"Qualifying Offering(s)"** means a Participant's goods or services that are conformant pursuant to the testing procedures described in the Test Evaluation Guide, where the conformance testing results or confirmations, as applicable have been submitted to and accepted by The Linux Foundation; and
- **"Component(s)"** means each of (i) the Zowe API Mediation Layer, (ii) the Zowe CLI, (iii) the Zowe App Framework and (iv) any future technology included in the Zowe project that is designated as a "Component" in the Test Evaluation Guide.

Related Documents

Together with these terms and conditions, the following documents are part of the Conformance Program, each as updated from time to time:

- **"Test Evaluation Guide"**: the conformance testing instructions and requirements for the Conformance Program, available at <https://www.openmainframeproject.org/projects/zowe/conformance/test-evaluation-guide>;
- **"Visual Branding Guide"**: the Conformance Program Visual Branding Guide, available at <https://www.openmainframeproject.org/projects/zowe/conformance/branding-guide>;
- **"Trademark Usage Guidelines"**: The Linux Foundation's Trademark Usage Guidelines, available at <https://www.linuxfoundation.org/trademark-usage/>; and
- **"Participation Form"**: the Conformance Program Participation Form, available at <https://www.openmainframeproject.org/projects/zowe/conformance/participation-form-sign>.

Introduction

The Conformance Marks and the Project Word Mark are trademarks of The Linux Foundation. In connection with the Zowe community, The Linux Foundation has established the Conformance Program to achieve two objectives: 1) to ensure that the Conformance Marks and the Project Word Mark remain reliable indicators of the qualities that they have been created to

preserve, and 2) to ensure that community members are able to accurately describe their Qualifying Offerings.

To achieve these objectives, The Linux Foundation requires that third parties who use the Conformance Marks in association with their goods and services may do so only as Participants pursuant to the Conformance Program described in this document. By participating in the Conformance Program and by using the Conformance Marks, a Participant acknowledges that third parties will be relying on the accuracy of its self-testing results and confirmations and on its compliance with the terms of the Conformance Program.

Participant Requirements:

In order to be a Participant in the Conformance Program and to use the Conformance Marks, a Participant must do the following:

- <https://www.openmainframeproject.org/projects/zowe/conformance/terms-and-conditions>
- The Open Mainframe Project - a vendor neutral entity - and top level project of the Linux Foundation Operates the program
 - This helps commercial vendors build and submit commercial offerings and keep confidentiality from potential competitors
 - Un-biased assessment of conformance
- Zowe Open Source Community identifies and develops the conformance criteria (Test Evaluation Guide)
- Program is responsible for managing the use of the Zowe Brand and Marks

Test Evaluation Guide



Zowe Conformance Test Evaluation Guide

The Zowe Conformance Test Evaluation Guide is a set of self-certify and self-service tests to help the developer community integrate and extend specific technology into the Zowe framework.

Below are the requirements for the available conformance programs. Items marked **(required)** are required for achieving conformance in a given program.

Zowe API Mediation layer - 2019

- 1) **Application Service**
 - a) An application provides at least one service or UI **(required)**
- 2) **Register with discovery services**
 - a) A service must be registered using one of the following methods: **(required)**
 - Dynamic registration - **preferred** (best practice)
 - Static definition - **minimum requirement**
 - b) The service must provide a default service ID that is prefixed by the provider name (for example: 'acme', 'xyzcorp', 'bar'). **(required)**
 - c) The service ID must be configurable externally after deployment. **(required)**
 - d) The service ID must be written in lower case, contain no symbols, and is limited to 64 characters. **(required)**
 - e) The API ID must follow the same rules as for Java packages. The example of the API ID: org.zowe.apiml.apicatalog. **(required)**
 - f) The published service URL must follow the gateway URL conventions. **(required)**
 - g) For versioned APIs, service URLs must contain a service version before the service ID in the following formats:
 - api/v1/{serviceId} reserved for REST APIs
 - ui/v1/{serviceId} reserved for UIs
 - ws/v1/{serviceId} reserved for WebSocketsFor non-versioned APIs or APIs versioned differently (e.g. z/OSMF), use the following formats:
 - api/{serviceId} reserved for REST APIs
 - ui/{serviceId} reserved for UIs
 - ws/{serviceId} reserved for WebSockets
- 3) **API Documentation**
 - a) Documentation is Swagger/Open API 2.0 compliant. **(required)**
 - b) Every public resource is documented with a description of each resource. **(required)**
 - c) Every method of each REST endpoint is documented. **(required)**
 - d) Every method of each REST endpoint is demonstrated by an example. **(required)**
 - e) Every parameter (headers, query parameters, payload, cookies, etc.) is documented with definitions of all possible values and their associated meanings. **(required)**
 - f) Every error code, including errors returned in the payload is documented. **(required)**
- 4) **API naming and addressing**
 - a) Encoded slash is not used. **(required)**

- b) The service interprets values independent of their URL encoding. **(required)**
 - c) lowerCamelCase is used for names of resources, parameters, and JSON properties.
- 5) **Service requests and responses**
 - a) API - Request and response payloads are in JSON or binary data format.
 - b) API - In JSON format, links are relative, and must not contain the schema, hostname, and port. **(required)**
 - c) WebSocket - Service URLs contained in WebSocket messages payload are addressed through the API ML Gateway. **(required)**
 - d) UI - UI uses relative links and does not contain the schema, hostname, and port. **(required)**
 - 6) **Authentication and Authorization**
 - a) Resources are protected by mainframe credentials. **(required)**
 - b) Services accept basic authentication (minimum requirement). **(required)**
 - c) Services accept Zowe JWT token in the cookie.
 - 7) **Versioning and Support**
 - a) Services must support the following versioning schemes:
 - REST API - Versioned APIs must use the following format: org.zowe.apiml.apicatalog.v1

- <https://www.openmainframeproject.org/projects/zowe/conformance/test-evaluation-guide>
- Evaluation guide is organized based on the Zowe Core Components (API ML, CLI, Web-UI)
 - A set of self-service checklist
- Will continue to be developed as the technology develops
- Items marked as **(Required)** are in-fact required
 - Items not marked **(Required)** are suggested best practices

Participant Form



The screenshot displays the 'Zowe Conformance Program Participant Form' in a web browser. The form is titled 'Zowe Conformance Program Participant Form' and includes a 'Submit' button at the top left. The main content area contains a detailed introduction to the program, followed by a 'Participant Information' section with a red-bordered input field for the participant's name. Below this is a 'Qualifying Offering' section with a red-bordered input field for the offering name. The 'Conformance Details' section includes three numbered items: 1. Zowe JCL Module Support (2019), 2. Zowe CLI (2019), and 3. Zowe Tool Framework (2019). Each item has a red-bordered input field for the participant's response. At the bottom, there is a 'Conformance Submission Date' field and a 'Signed on behalf of Participant by' section with a red-bordered input field for the participant's name and a date field.

- <https://www.openmainframeproject.org/projects/zowe/conformance/participation-form-esign>
- Participant form is a digital submission to the Open Mainframe Project
- One submission per qualifying offering
 - A single offering can submit for one or more component criteria in a single submission
- Must identify any deviations from conformance

Review



- Once the Participation Form is received, it will then be reviewed
 - Approved, Needs Clarification or Rejected
- Review can take between 2-5 business days (depending on demand/backlog)
- If approved, will receive email with rights to use marks and a copy of the branding guidelines:
<https://www.openmainframeproject.org/project/zowe/conformance/branding-guide>



**ZOWE CONFORMANT
BRAND GUIDELINES**
July 2019

Software Vendors can build Conformant offerings with Zowe



Conformance Program



1. Prepare

Learn about the conformance program and the test evaluation guide

- Understand the Program Terms and Conditions: <https://www.openmainframeproject.org/projects/zowe/conformance/terms-and-conditions>
- Review the Program Test Evaluation Guide to understand the criteria required to meet conformance: <https://www.openmainframeproject.org/projects/zowe/conformance/test-evaluation-guide>



2. Run the Test

Follow instruction for how to test your application

- Make sure that you have reviewed the criteria and that you document any and all deviations: <https://www.openmainframeproject.org/projects/zowe/conformance/test-evaluation-guide>
- Review the Participation Format format (trust but verify): <https://www.openmainframeproject.org/projects/zowe/conformance/participation-form>



3. Submit

Complete the terms and conditions and submit your request

- Submit your Participant Form digitally through DocuSign: <https://www.openmainframeproject.org/projects/zowe/conformance/participation-form-esign>

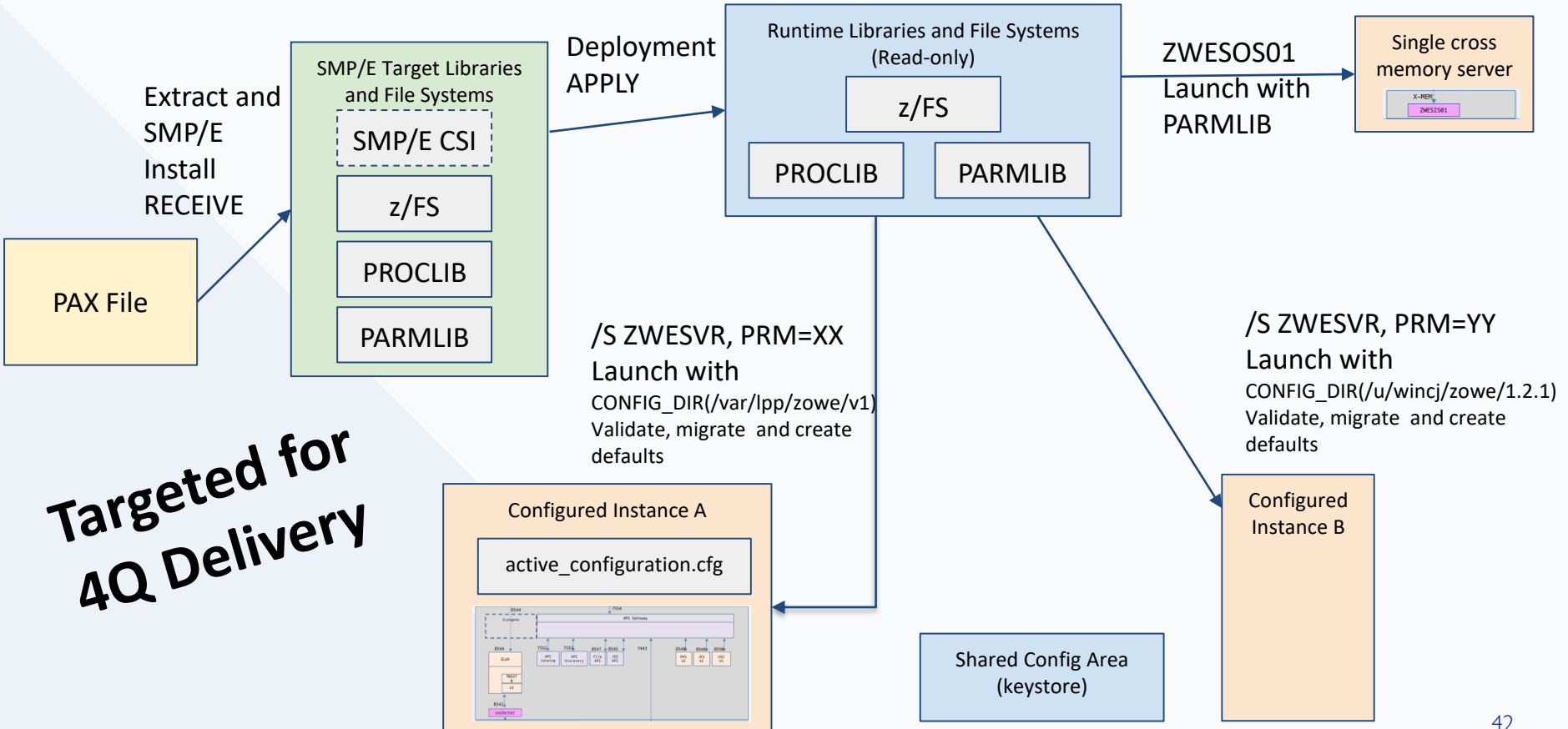


4. Review

A reviewer will approve your request and you will be conformant

- Upon review and confirmation that you have met conformance you will earn the rights to use the Zowe marks and logo. Review and adhere to the branding guidelines: <https://www.openmainframeproject.org/projects/zowe/conformance/branding-guide>

To Be - Installation and launch Topology



Trademarks

Zowe™, the Zowe™ logo, and the Open Mainframe Project™ are trademarks of The Linux® Foundation. Linux is a registered trademark of Linus Torvalds.

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

* Registered trademarks of IBM Corporation

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELoS Limited.

ITIL is a Registered Trade Mark of AXELoS Limited.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon 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 throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g. zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

