



Zowe
To Open Source and Beyond

Today's Presenters



Ivy Li
Senior Manager - Z Automation, IBM



Ashley Li
Zowe Content Designer, IBM

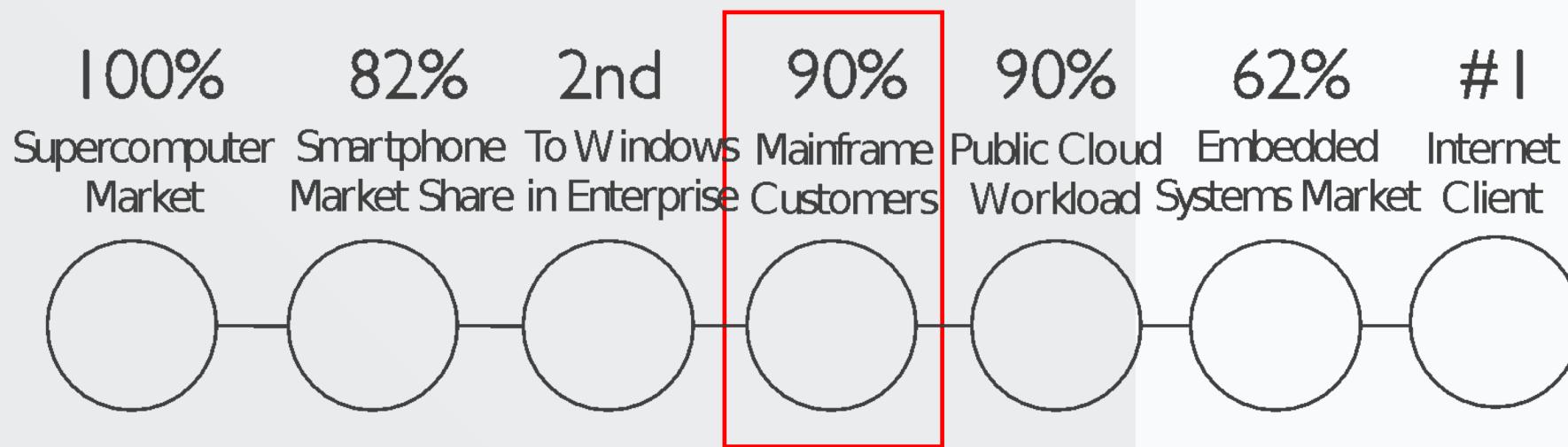


Agenda

- Open Mainframe Project Introduction
- Introduction to Zowe
- Zowe 1.0
- How to get Involved
- Q&A



Mainframe is key part of Open Source success



Every market Linux has entered it eventually dominates



Open Source on Mainframe challenges

Disconnected, independent efforts; no shared “hub” of innovation

Community events are industry specific, not vendor agnostic

Open source on the mainframe lacks a neutral home for growth

No place for students and academic institutions to engage

Enterprise level engagement with upstream projects limited



Look to The Linux Foundation



Thankfully, that's where The Linux Foundation® comes in. For nearly two decades, The Linux Foundation has provided unparalleled support for open source communities through financial and intellectual resources, governance structure, IT infrastructure, services, events, and training.

Dedicated to building sustainable ecosystems around open source projects, The Linux Foundation is working with the global technology community to solve the world's hardest problems through open source and **creating the largest shared technology investment in history**.

The Linux Foundation is the umbrella organization for **more than 60 open source projects** accelerating open technology development and commercial adoption. Some of the game-changing initiatives hosted by The Linux Foundation include:





Open Mainframe Project community traction

3

years since launch

29

Supporting organizations

6

Hosted Mainframe centric
Open Source Projects

100+

Students impacted through
internships and academic
programs

Members



HoGent



MARIST



Rocket

SHARE
EDUCATE • NETWORK • INFLUENCE



ubuntu
Supported by Canonical



UNIVERSITY OF WOLVERHAMPTON



WECOM Infinity

WILLIAM PATERSON UNIVERSITY

Open Mainframe innovation thrives here



- OMP provides a vendor-neutral home for mainframe-centric open source projects
 - Code hosting/infrastructure
 - Governance
 - Legal/Trademark defense
 - Ecosystem development
- LF staff support project communities establishing guidelines and best practices to enable diverse community growth and adoption
- Our philosophy creates natural collaboration opportunities between mainframe-centric open source projects

Project lifecycle, guidance, and proposal process openly defined at
<https://github.com/openmainframeproject/tsc/tree/master/process>

Sustained mainframe support in the broad open source community through the Supported Projects program



| Infrastructure | Developer support | Market awareness | Governance/IP Home |
|---|---|---|---|
|  |  |  |  |

Participating Open Source Projects include





Build community and adoption of Open Source on the mainframe

- **Eliminating barriers** to Open Source adoption on the mainframe
- **Demonstrating value** of the mainframe on technical and business levels
- **Strengthening collaboration** points and resources for the community to thrive



Eliminating barriers to Open Source adoption on the mainframe

- Engaged as central expert in demonstrating the mainframe as a viable open source platform, with compelling advantage.
- Promotion of modern application and workload examples on the mainframe
- Growing career opportunities and academic community engagement
- Hosting and participating in local programming and promotion of Open Source on the mainframe
- Sponsorship of global initiatives and contests to grow enthusiasm for the platform



Demonstrating value of the mainframe on technical and business levels

- Showcase of technical and business case studies through blogs, white papers, and other media.
- Champion software and hardware solutions with clients



Strengthening collaboration

points and resources for the community to thrive

- Engagement through Technical Steering Committee projects and independent projects by members
- Visibility to tools, resources, and community forums to tackle technical challenges
- Career opportunities from internships through retirement



Internship and academic engagement programs are putting mainframe in the hands of the next generation



Summer Internship Program

27

Interns sponsored

14

Supporting
academic
institutions

100+

Students
impacted



VCU Capstone Partnership



Telling the “mainframer” story

- Monthly interview series that highlights both new and old in mainframe
- Goal is to showcase why people have mainframe in their careers and their views of the technology and career field
- Read and listen at
<https://www.openmainframeproject.org/category/blog/i-am-a-mainframer>



Bringing together the open source and mainframe conversation



Sponsored Meetup Program

<https://www.openmainframeproject.org/meetup-program>



Slack Channel

<https://slack.openmainframeproject.org>

The screenshot shows a list of forum topics from the Open Mainframe Project Discourse Forum. The topics include discussions about SMPI, compatibility, academic committees, and various technical meetings. Each topic has a small preview image, the number of replies, and the number of views.

| Topic | Replies | Views |
|---|---------|-------|
| Calling SMPI from Linux | 6 | 266 |
| What Open Source tools are missing from OS/390 and need to be ported? | 13 | 506 |
| Welcome to the Open Mainframe Project Discourse Forum | 8 | 942 |
| Open Mainframe Academic Committee | 0 | 51 |
| Cloudstack for z/VM Meeting July 21, 2017 | 1 | 123 |
| Anyone doing Big Data on z? | 9 | 370 |
| Cloudstack for z/VM Meeting July 14, 2017 | 1 | 108 |
| What can Mainframers do to best capture all of its capabilities? | 0 | 63 |
| Cloudstack for z/VM Meeting June 28, 2017 | 2 | 136 |
| Blockchain and the Mainframe | 1 | 89 |
| Cloudstack for z/VM Meeting June 16, 2017 | 5 | 162 |
| Open Mainframe Project Summer 2017 Interns | 2 | 116 |

Community Forums

<https://community.openmainframeproject.org/>

How to participate in Open Mainframe Project



Open participation for all interested in Mainframe and Open Source

Community discussion and collaboration - both virtually and at regional events.

6 Shared R&D open source projects - or bring your project to be hosted here

Corporate sponsorship for showing stewardship in the community



Learn more about the Open Mainframe Project

- Find out more and subscribe to our newsletter at www.openmainframeproject.org
- Organizational membership opportunities at
<https://www.openmainframeproject.org/about/join> or email at
membership@openmainframeproject.com



Zowe Overview



Introducing Zowe

- An extensible framework for connecting applications and tools to mainframe data and applications.
- 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 Announce at THINK SF 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 are founding members
- **Generally Available on Feb 8th, 2019**



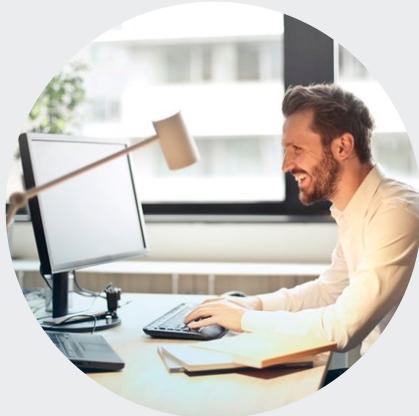
Why Zowe ?



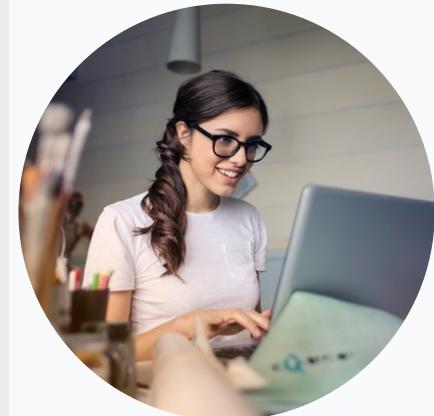
Information Technology is undergoing a revolution of changing architectures



Co-Existence With
Other Cloud Models



Protecting Current
and Future
Investments



Simple and
Familiar



Zowe Vision Statement

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



What's in Zowe?

Browser-based Web Desktop

The screenshot shows the Zowe web interface. On the left, there's a sidebar with links for Swagger, API Documentation, Terms of service, Apache 2.0, and JES job APIs. The main area has two windows: 'JES Explorer' showing job details for 'WINCHU2' and 'JOB05954', and a 'Content Viewer' showing the 'JESMSGLG' file. On the right is a terminal window titled 'TN3270' displaying a list of jobs and their execution details.

swagger

Api Document

[Base URL: <https://winmvs27.hursley.ibm.com:8454/LUX/plugin/com.rs/mvd/index.html>

API Documentation

Terms of service

Apache 2.0

JES job APIs

Jobs Controller

| | | |
|------------------------|---|--|
| GET | /api/v1/jobs | Get a list of all jobs |
| GET | /api/v1/jobs/{jobname} | Get a job by name |
| DELETE | /api/v1/jobs/{jobname}/{jobId} | Cancels a job and purge its associated files |
| GET | /api/v1/jobs/{jobName}/files | Get a list of output file names for a job |
| GET | /api/v1/jobs/{jobName}/{jobId}/files/{fileId}/content | Get content from a specific job output file |
| GET | /api/v1/jobs/{jobName}/{jobId}/steps | Get job steps for a given job |
| POST | /api/v1/jobs/dataset | Submit a job given a data set |

Swagger-defined z/OS REST APIs

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: <https://casx.ca.com:10010/api/v1/apicatalog>]

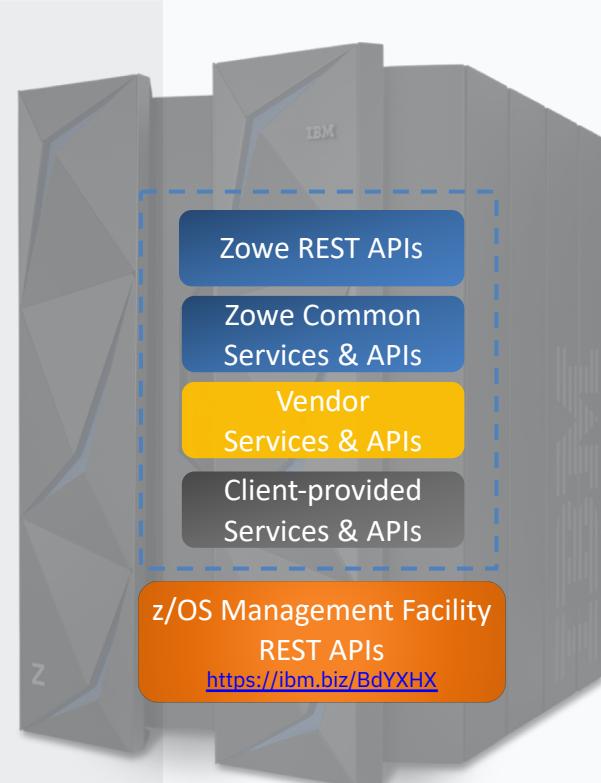
REST API for the API Catalog service which is a component of the API Mediation Layer. Use this API to retrieve information regarding catalog dashboard tiles, tile contents and its status, API documentation and status for the registered services.

The screenshot shows a terminal window with the command 'bright zos-files'. Below it is a detailed description of the 'zos-files' command, including its usage and actions. The 'DESCRIPTION' section states: 'Manage z/OS data sets and USS files'. The 'USAGE' section shows the command syntax: 'bright zos-files [action] [object] [options]'. The 'ACTIONS' section lists several actions: 'ADD', 'CREATE', 'DELETE', 'DISPLAY', 'MOVE', 'PURGE', 'REFRESH', 'RENAME', 'REPAIR', 'REVERSE', 'SET', 'STATISTICS', 'STOP', and 'START'.

Node.js- based CLI



- Industry standard REST interfaces to z/OS resources that are language and platform neutral, stateless and scalable
- Foundational building blocks for system services
 - ***Dataset APIs***
 - Create, read, update, delete, and list data sets
 - ***JES APIs***
 - View the information and files of jobs, and submit and cancel job
 - ***USS APIs***
 - Create, read, update, and delete USS files
 - ***System APIs***
 - View information about PARMLIB, SYSPLEX, and USER





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

The screenshot displays the Zowe Web Desktop environment with four active windows:

- JES Explorer**: Shows a job log for JES2 JOB LOG -- BY SYSTEM HVJ.7. It lists various jobs and their statuses, such as 09-06-14 JOB857954 and 09-06-14 JOB857954.
- TN3270**: A terminal window showing a command line interface with various commands and output.
- USS Explorer**: A Unix File Explorer window showing the directory structure of /u/winchj, including files like .bash_history, .profile, .sh_history, and .ssh.
- MVS Explorer**: A Content Viewer window displaying the contents of WINCHJ.JCL(COPY). It shows JCL code and a note about the SYSOUT parameter.

UI Challenge

Previous solutions to the problem fall short

- Fat clients installed on end-user machines
 - Adds prereqs to each machine
 - May be windows-only
 - May replicate data already on Z

- Good UIs use your intuition to gain insight and complete high level tasks
- Coexistence with CLI where low level tasks & automation exceed
- z/OS doesn't provide tech to present a UI on the level of those of consumer devices
- X11 present on USS, but not much software support

- Websites
- Scope limited to targeted category of software Extensibility limited by traditional capabilities of HTML/CSS/JS
- Limited communication with other software/sites for logical workflow
- Some based on technology becoming unsupported (activex, java plugin)

UI Solution

Zowe App Framework

App framework makes it possible to have multiple apps, written by different parties, using different web technologies, to coexist within the same page

- Create an extensible webpage for running multiple web applications simultaneously
- UI seen as a so-called “Single Page App” (actions done without navigating browser between pages)

Dev freedom to code using different web tech
Security & Compatibility – isolation of framework inner objects from Apps, isolation between Apps (JS, HTML, CSS)
Performance – page load speed unaffected – Apps loaded only first opening
Minimal memory consumption – common libraries are included in base, Apps can use for “deduplication”
Flexibility – Design by interface allows for changes in layout & technology support

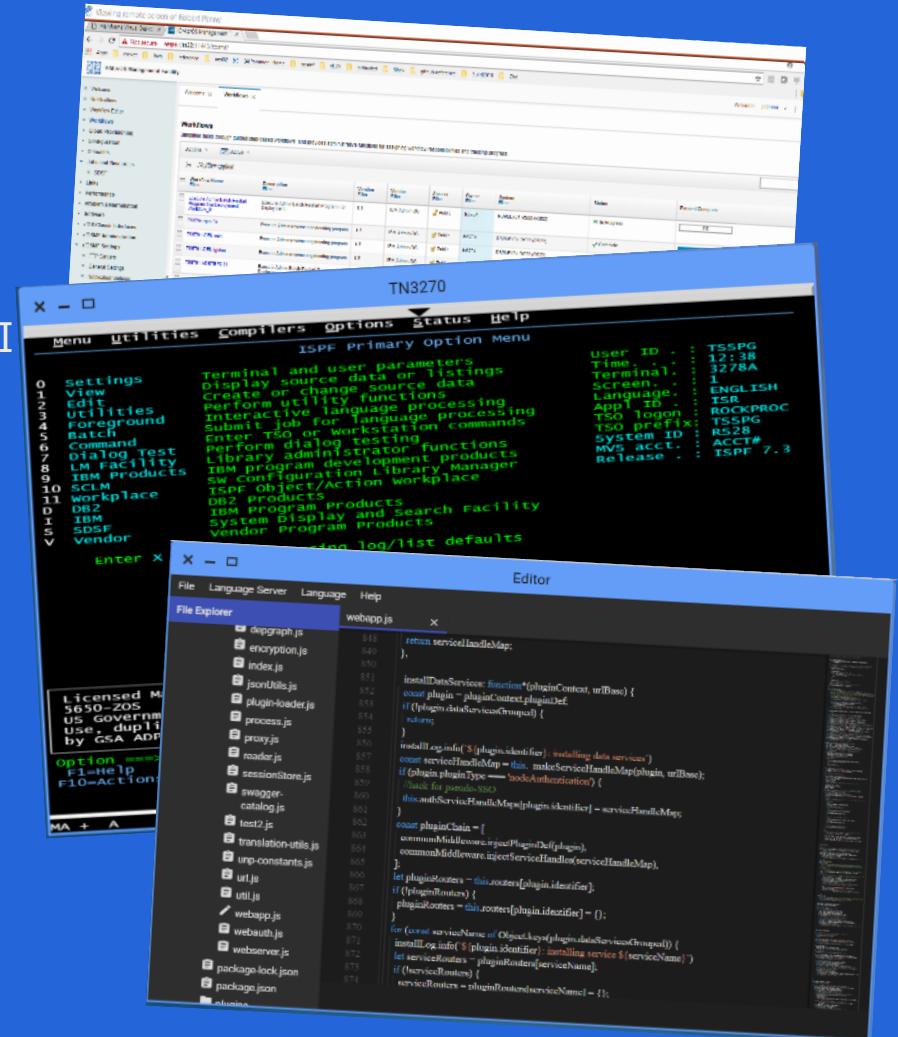
UI Solution

Goal: one intuitive & modern place for all z/OS UIs

- Existing websites can be presented in UI via iframe wrapper, mediation layer for solving CORS
 - Terminal (3270 and SSH) present for compatibility
 - Modern web libraries rich enough for representing look&feel seen in fat clients

Zero install: Pure HTML, CSS, and JS in a browser

- Chrome, Firefox, Edge, Safari – use it on a chromebook or a desktop



Want's a web dev to do ?

Making an App for Zowe isn't so different, and you have choices

1. Have a preexisting site?

- Connect its server to the mediation layer
- Make an iframe App that references the mediation layer link

```
# Get database
#db = client[MONGO_DB]
#print("initialized db at %s:%i" % (MONGO_DB, MONGO_HOST, MONGO_PORT))

In [1]: # Created database
# Create table
import pymongo
conn = pymongo.connect(ssid="AZK5")
cursor = conn.cursor()
cursor.execute("select * from client_info")
rows = cursor.fetchall()
print("#rows: {}".format(len(rows)))

#rows: 6001

In [2]: # Imports and definitions

import numpy
import scipy
import scipy.optimize
from bokeh import plotting, charts, models
from pyspark import SparkContext

if "sc" in globals():
    sc.stop()
    print("stopped current context")
sc = SparkContext("local", "SparkDemo2")
print("created new context")

def printSample(rdd, str, lines):
    print(str % (rdd.count()))
    sample = rdd.take(lines)
    for s in sample:
        print(s)

def sortByValue(rdd, asc = True):
    return rdd.map(lambda a, b: (b, a)).sortByKey(asc).map(lambda a, b: (b, a))
```

What's a web dev to do ?

Making a new program?

Code in React or Angular (Vue, etc possible in future)

Utilize App framework base webpack config, and typescript config (typescript not required, but very recommended)

BIGGEST difference – webpack config clashes with dev tools such as angular cli

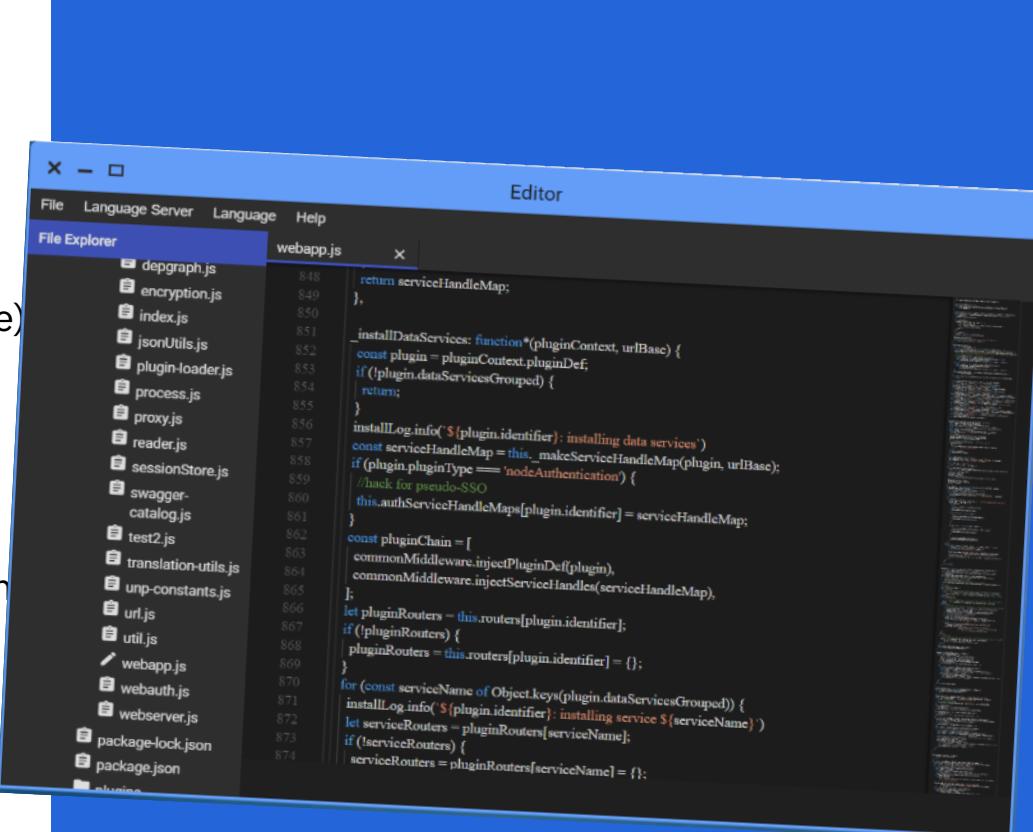
Minor zowe accommodations for entry, css isolation (ex css modules), and windowed-awareness (events)

Growing collection of optional-but-recommended framework features

Notification API, Settings storage API

URI abstraction API, App2App communication API, Globalization API

Logging API, Backend APIs for building REST/WS



The screenshot shows a code editor window with the file 'webapp.js' open. The code is a JavaScript file containing logic for managing service handle maps and plugin routers. It includes imports for common middleware and service handles, and defines functions for installing data services and service routers based on plugin identifiers. The code uses template literals and object destructuring.

```
return serviceHandleMap;
},
installDataServices: function*(pluginContext, urlBase) {
  const plugin = pluginContext.pluginDef;
  if (!plugin.dataServicesGrouped) {
    return;
  }
  installLog.info(` ${plugin.identifier}: installing data services`);
  const serviceHandleMap = this._makeServiceHandleMap(plugin, urlBase);
  if (plugin.pluginType === 'nodeAuthentication') {
    //Inec for pseudo-SSO
    this.authServiceHandleMaps[plugin.identifier] = serviceHandleMap;
  }
  const pluginChain = [
    commonMiddleware.injectPluginDef(plugin),
    commonMiddleware.injectServiceHandles(serviceHandleMap),
  ];
  let pluginRouters = this.routers[plugin.identifier];
  if (!pluginRouters) {
    pluginRouters = this.routers[plugin.identifier] = {};
  }
  for (const serviceName of Object.keys(plugin.dataServicesGrouped)) {
    installLog.info(` ${plugin.identifier}: installing service ${serviceName}`);
    let serviceRouters = pluginRouters[serviceName];
    if (!serviceRouters) {
      serviceRouters = pluginRouters[serviceName] = {};
    }
  }
}
```



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

Build | Test | Deploy



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

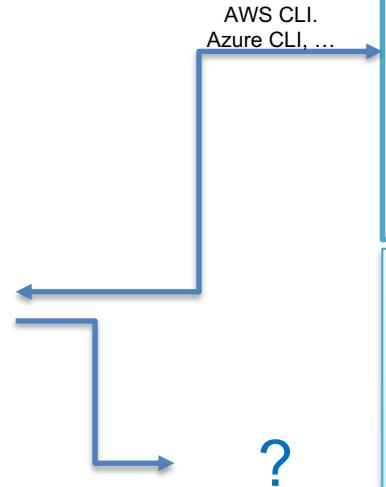
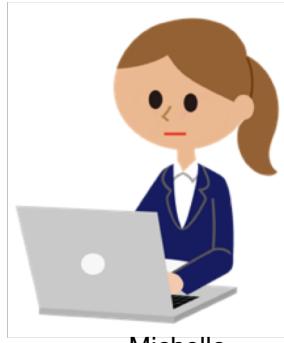
Access to CICS and DB2

```
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)
```

Challenge



Cloud



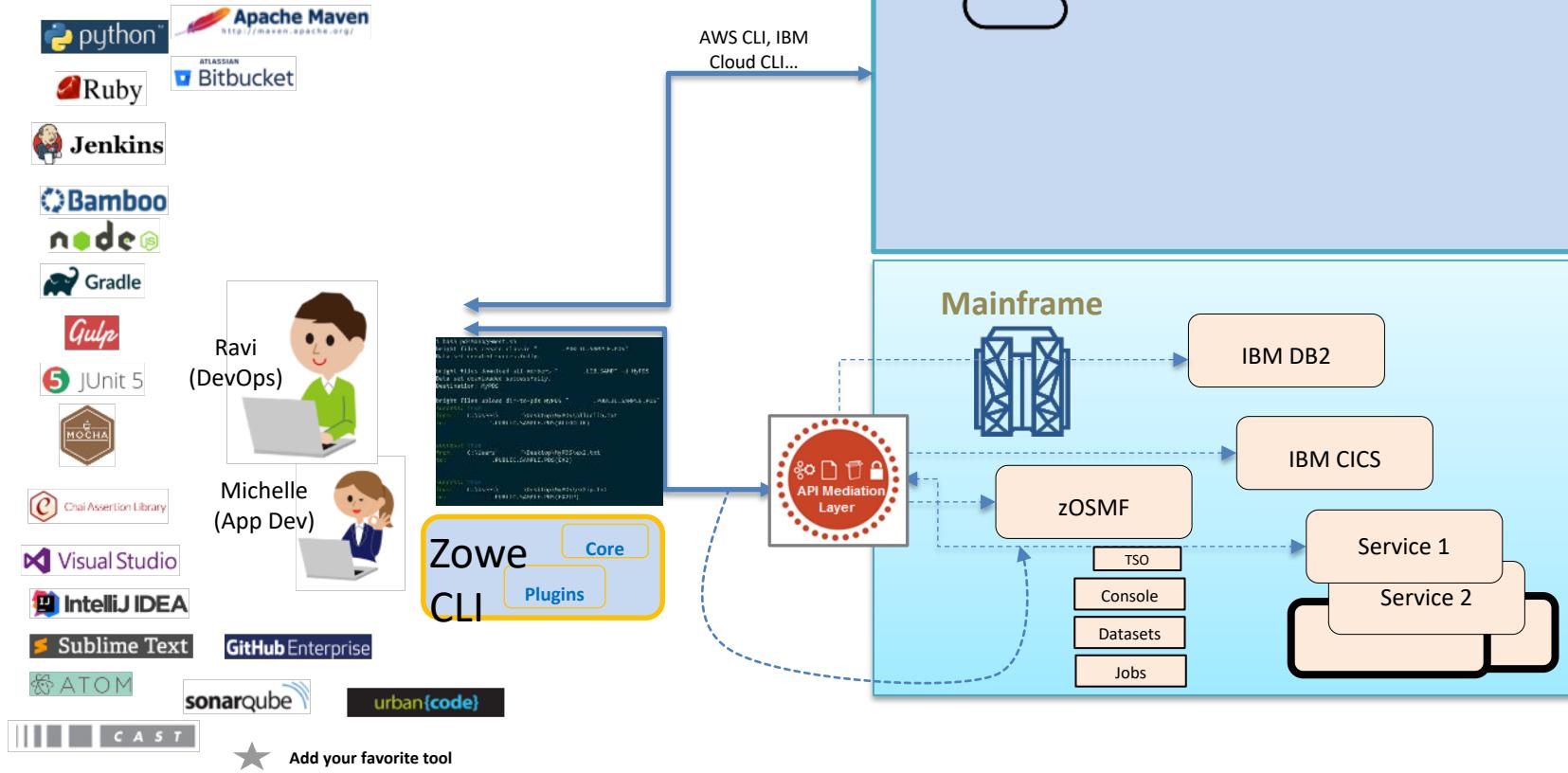
Mainframe



★ Add your favorite tool



Zowe CLI



Community



- How do I call mainframe from Jenkins?

A screenshot of a Stack Exchange question page. The title of the question is "Which plugin should I use to get started with Jenkins to manage mainframe components in PDS format?". The question text asks for ways to automate actions from Jenkins, listing two methods: installing a plugin or writing a script. It also discusses the drawbacks/benefits of each approach. A note at the bottom says the scripting approach uses a lot of CLIs like aws cli, docker cli, etc.

In general, there are two ways to automate actions from Jenkins:

1. Install a plugin for Jenkins and use it in steps to automate tasks.
2. Write a script (shell, python, other) and call it from a step to automate tasks.

There are drawbacks / benefits to both approaches. While #1 - the plugin approach - gives you out of the box functionality and does not require you to script, it may be rigid and limited in its abilities. #2 - the scripting approach - requires you to write scripts, as well as maintain them, but may provide much more flexibility and freedom in automating things.

The scripting approach generally uses a lot of CLIs - aws cli, docker cli, etc.

- Build IDE extensions!

A screenshot of the Visual Studio Marketplace. The page shows the "Zowe" extension by Zowe. It has 196 installs and 560 downloads. The rating is 5 stars based on 1 review. The description states it is a VS Code extension powered by Zowe CLI, streamlining interaction with mainframe data sets. There are "Install" and "Trouble Installing?" buttons.

A screenshot of a GitHub repository for "zowe/vscode-extension-for-zowe". The repository has 17 issues, 0 pull requests, 0 projects, 0 wiki pages, and 0 insights. The main page features a "Submit Jobs #16" section with a merged pull request from ChrisBoehmCA. A comment from crshnburn on Nov 4, 2018, suggests adding the ability to submit the content of the editor or a dataset member as a job and get back the job id. The comment has 2 likes.

- 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***

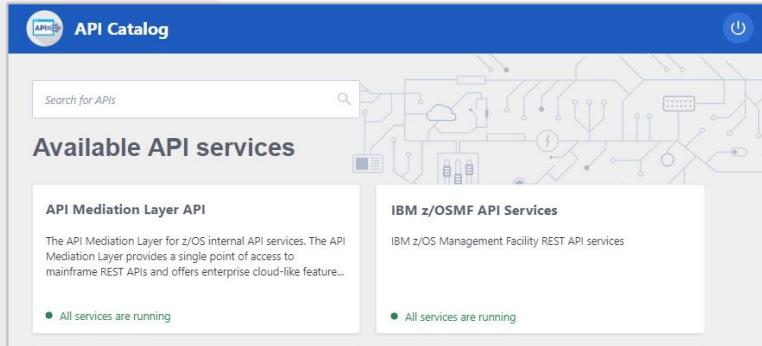
UI Catalog of available APIs with their Swagger doc and service status

- ***Gateway***

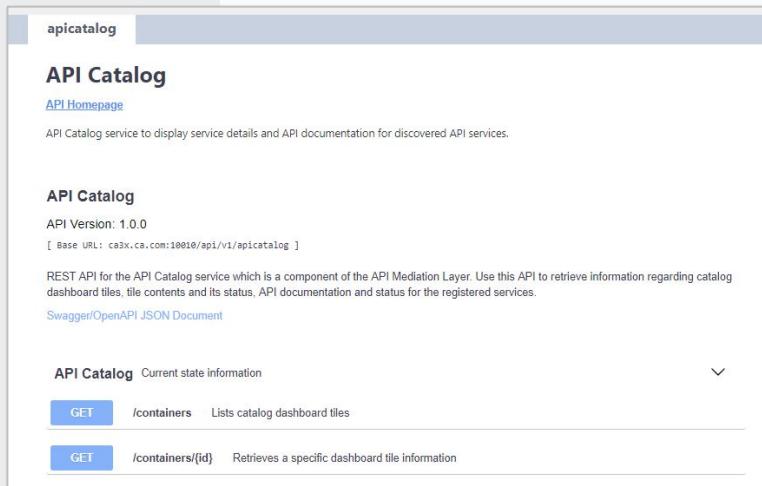
Single secure point of entry to an ecosystem of API services.
Hides complexity. Highly available. Based on Netflix Zuul.

- ***Discovery Service***

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



The screenshot shows the API Catalog interface. At the top, there's a search bar labeled "Search for APIs". Below it, a section titled "Available API services" lists two items: "API Mediation Layer API" and "IBM z/OSMF API Services". Each item has a brief description and a status indicator: "All services are running". The background features a stylized circuit board graphic.

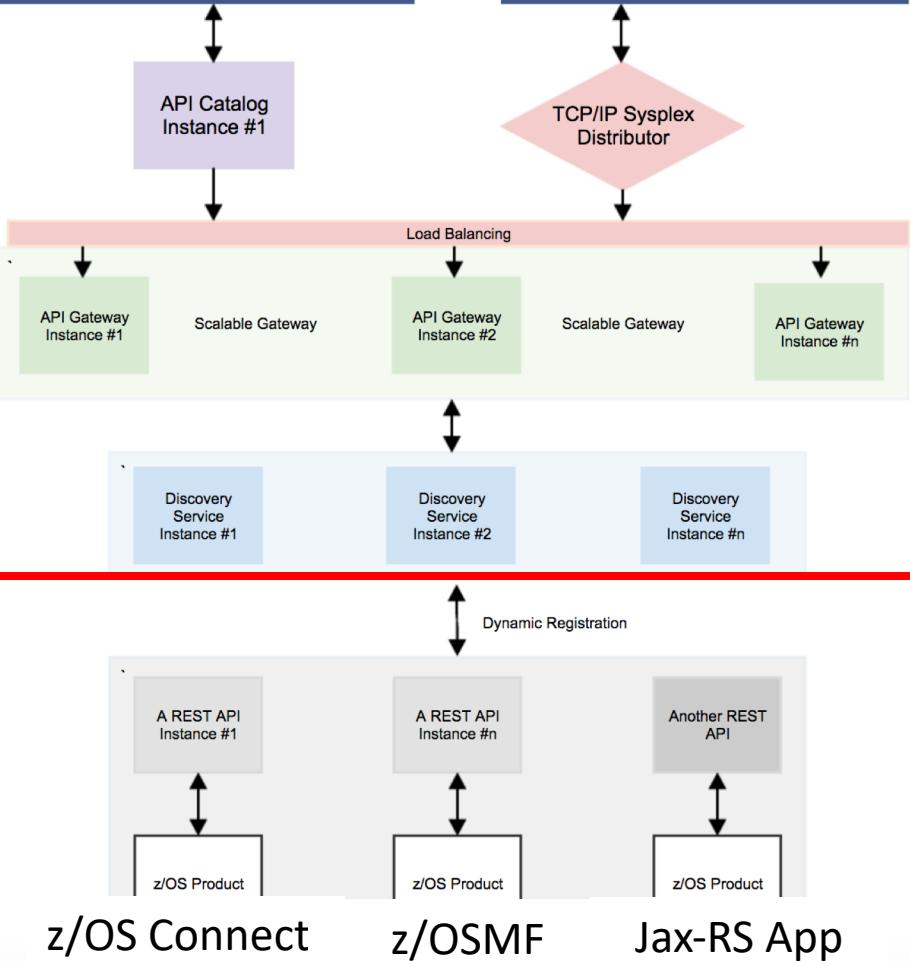


The screenshot shows the API Catalog service documentation. It includes a navigation bar with "apicatalog" and "API Catalog". Below that, it says "API Catalog" and "API Version: 1.0.0". It provides the base URL "[Base URL: ca3x.ca.com:10810/api/v1/apicatalog]". A detailed description states: "REST API 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." It links to "Swagger/OpenAPI JSON Document". At the bottom, there's a table of API endpoints:

| Method | Endpoint | Description |
|--------|------------------|---|
| GET | /containers | Lists catalog dashboard tiles |
| GET | /containers/{id} | Retrieves a specific dashboard tile information |

People Find APIs

Apps Call APIs



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.

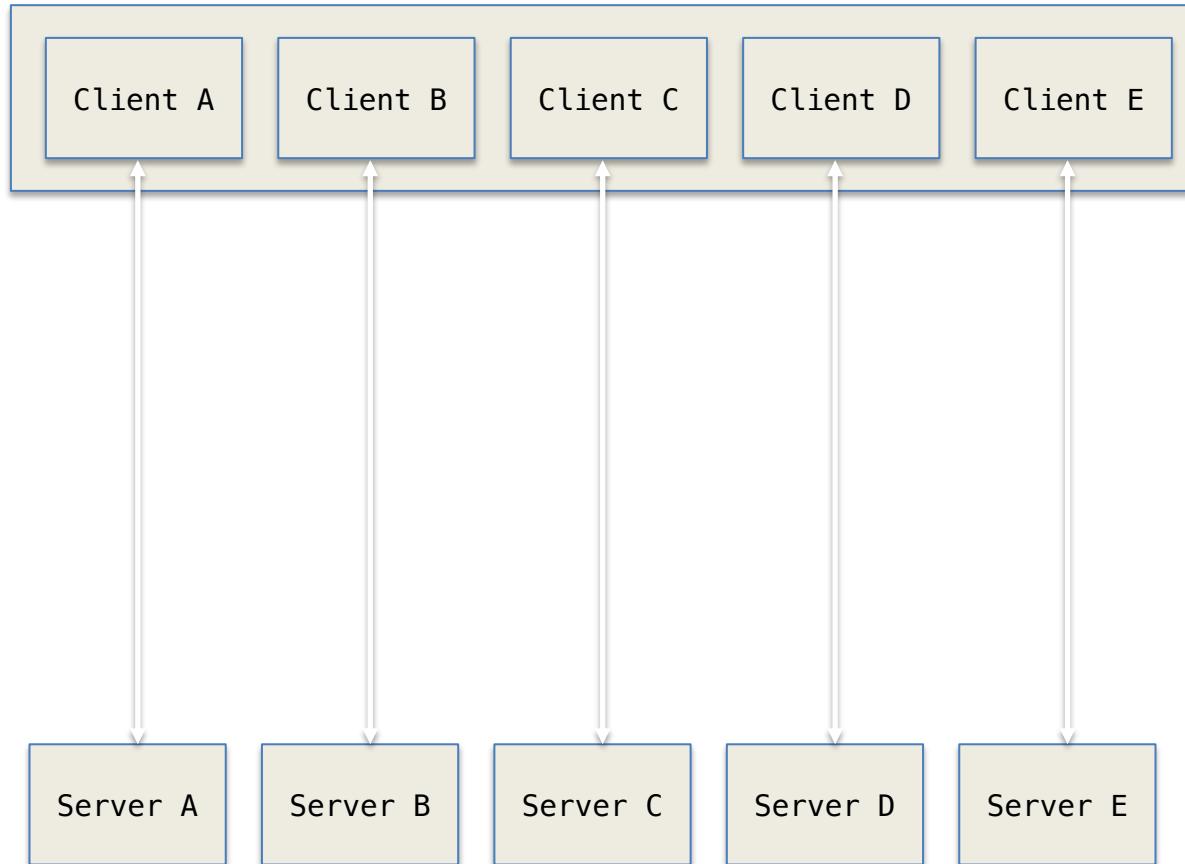
■ Discovery Service

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



Problems

Multiple Sign On

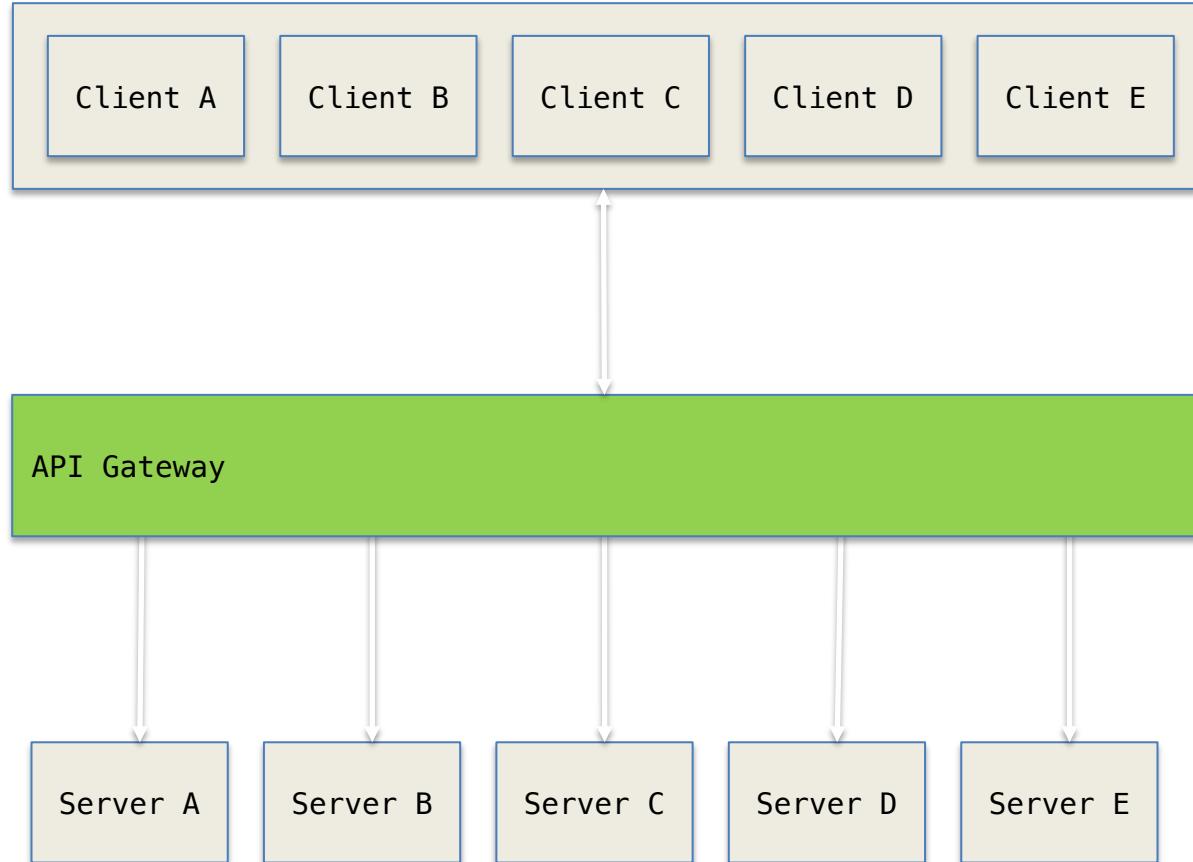
n Endpoint Configuration

Heterogeneous certificates

Cross Origin Request

...





API Gateway

Netflix Zuul/Eureka base

Single Sign On (SSO)

JSON Web Token

Reverse proxy endpoint

Swagger open API Catalog

Single client certificate

Gateway<-> Server

Static registration

Dynamic Discovery



Not Secure | https://winmvs3b.hursley.i... 338

Atlas IBM Personal Giza http://127.0.0.1:80...

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
- 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

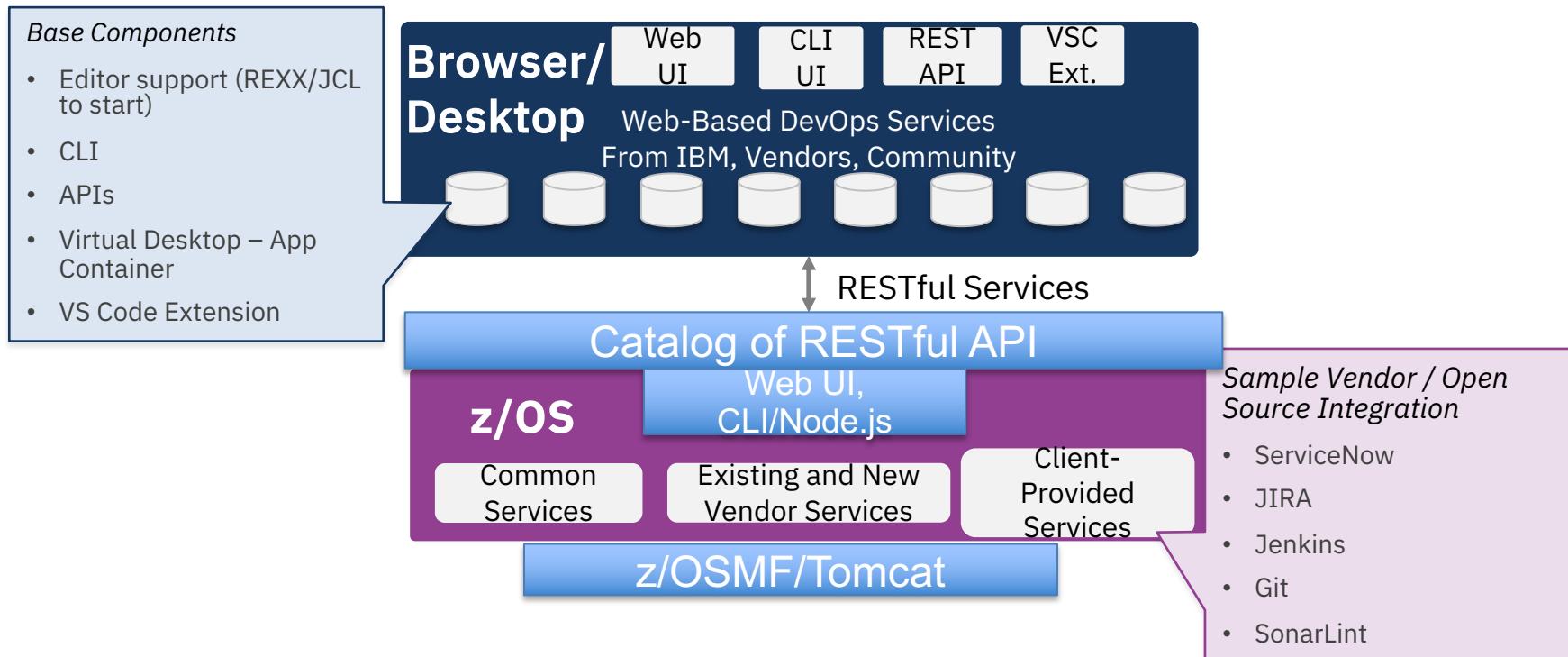
My Server APIs

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

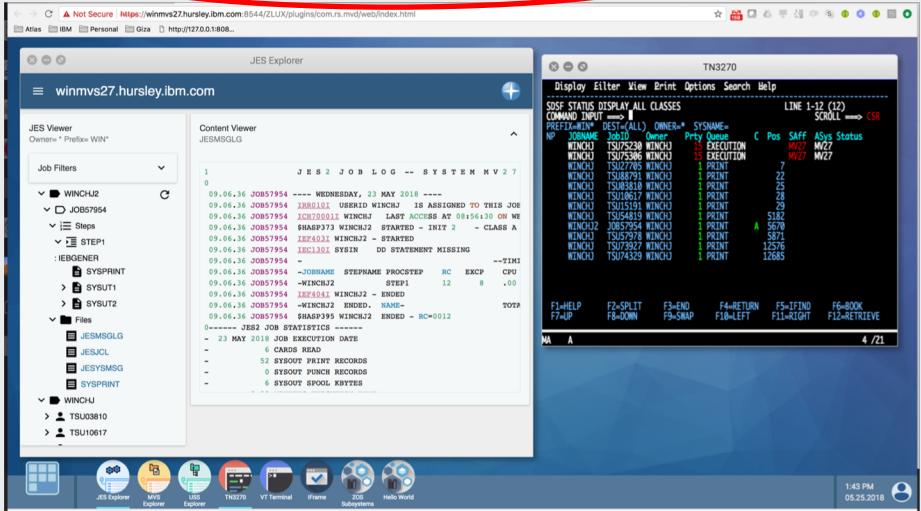
Zowe High Level Architecture



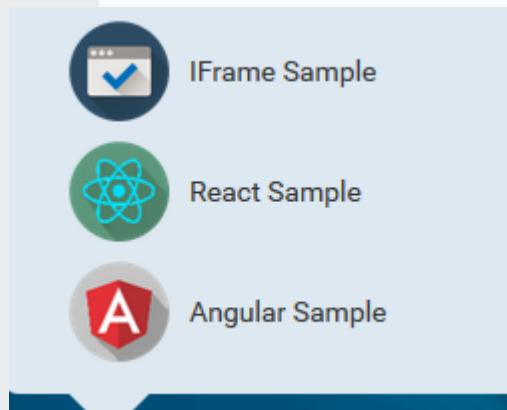
Where is Zowe Extensible?



Browser-based Web Desktop



- z/OS Native Web UI for applications
 - Launch in context (i.e., right mouse click 3270 to web app)
 - App to app communication
 - Exploit graphic widgets planned for inclusion





Where is Zowe Extensible?

- REST API enable your products
 - REST API for product controls/admin
 - Sharing of information

swagger

Api Documentation 1.0

[Base URL: <http://wimvms3b.hursley.ibm.com:7949/>]
<https://wimvms3b.hursley.ibm.com:7949/v2/api-docs>

Api Documentation
[Terms of service](#)
[Apache 2.0](#)

JES job APIs Jobs Controller

| | | |
|------------------------|---|--|
| GET | /api/v1/jobs | Get a list of jobs |
| GET | /api/v1/jobs/{jobName}/{jobId} | Get the details of a job for a given job name and identifier |
| DELETE | /api/v1/jobs/{jobName}/{jobId} | Cancel a job and Purge it's associated files |
| GET | /api/v1/jobs/{jobName}/{jobId}/files | Get a list of output file names for a job |
| GET | /api/v1/jobs/{jobName}/{jobId}/files/{fileId}/content | Get content from a specific job output file |
| GET | /api/v1/jobs/{jobName}/{jobId}/steps | Get job steps for a given job |
| POST | /api/v1/jobs/dataset | Submit a job given a data set |

Swagger-defined z/OS REST APIs

- Opt in to API Mediation
- Participate in Single Sign On, High Availability and Status tracking capabilities

API Mediation Layer
(API Catalog, Discovery Service, Gateway)

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: casx.ca.com:10010/api/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, 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 tiles |
| GET | /containers/{id} | Retrieves a specific dashboard tile information |

API Documentation Service documentation



Where is Zowe Extensible?

Node.js- based CLI

```
GROUPS
-----
plugins          Install and manage plug
profiles         Create and manage config
provisioning | pv Perform z/OSMF provision
                   in the Service Catalog and
                   Service Registry.

zos-console | console Issue z/OS console commands
zos-files | files Manage z/OS data sets
zos-jobs | jobs   Manage z/OS jobs
zos-tso | tso    Issue TSO commands and
                   Interact with z/OSMF

OPTIONS
-----
--version | -v (boolean)
  Display the current version of CA Brightsides

GLOBAL OPTIONS
-----
--response-format-json | --rfj (boolean)
  Produce the command response as a JSON document
--help | -h (boolean)
```

zos-files DS
zos-files US
zos-jobs
TSO
Console

“plug-ins”

Out of box
commands

z/OSMF

REST APIs
TSO, Console
JES, MVS, USS

Custom
Extensions

Your
application,
product,
tool, ...

Commitment to Core Infrastructure Initiative (CII) & Badge Program



- “CII is a collaborative, pre-emptive program and approach for strengthening cyber security that is widely supported by industry leaders”
- “CII Badge Program is a self-certify, declaration of industry best practices and conformances in driving secure software development and governance”

Notable CII “backers”

Source: <https://www.coreinfrastructure.org>



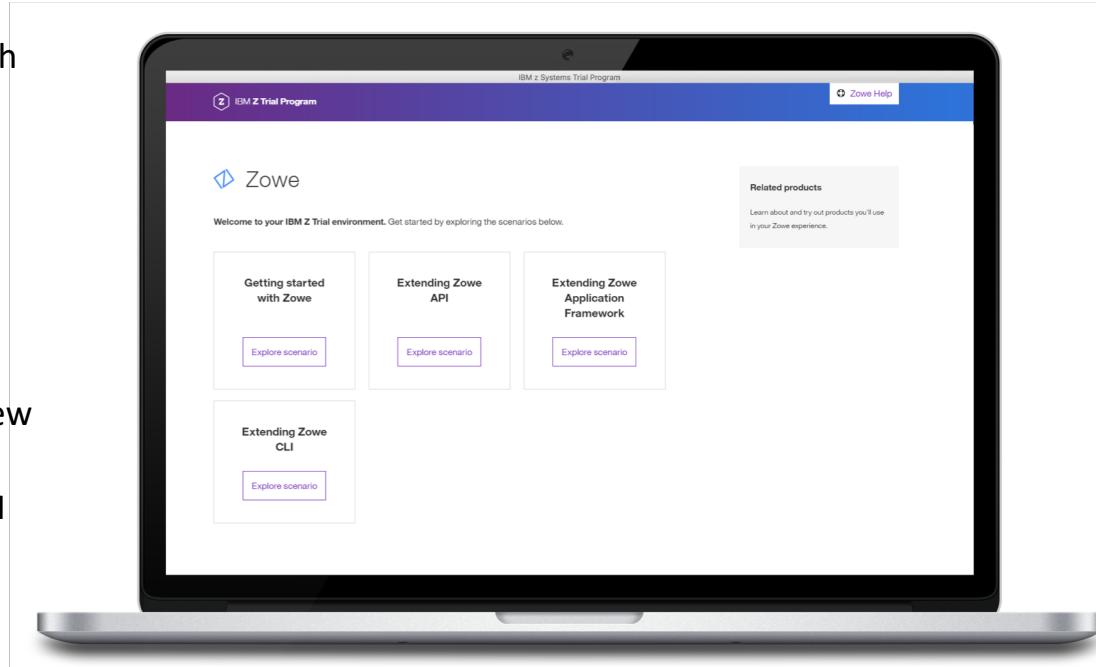
Notable Badge “earners”

Source: <https://bestpractices.coreinfrastructure.org>



Zowe Trial

- Try the Zowe capabilities at zero cost, and with no installation required.
- Pre-configured, remote desktop environment
- Your trial environment comes loaded with tutorials that show you how to:
 - Get started with Zowe
 - Create and extend Zowe with new APIs
 - Create and extend Zowe Desktop with new web application
 - Create and extend Zowe CLI with new CLI commands



IBM Z software trials:

<https://www.ibm.com/it-infrastructure/z/resources/trial>

The Zowe open community

Zowe Leadership Committee (ZLC)

Zowe Continuous Integration/Continuous Development (CI/CD)

Zowe On-boarding

Zowe Core Technology

Zowe API Mediation and Security

Mission: Develop the material and supporting activities for onboarding developers and customers

Mission: expand upon the base technologies being contributed to the project

Mission: expand upon integration and interface extension points and overall security

Open Mainframe Project Board

Open Mainframe Technical Steering Committee

Open Mainframe Marketing Committee

Check out

<https://zowe.org/about-us/>

A foundational principle of this new project is meritocracy. The more that somebody contributes, the more responsibility they will earn. A pattern of quality contribution to a project may lead to an invitation to join the project as a committer.

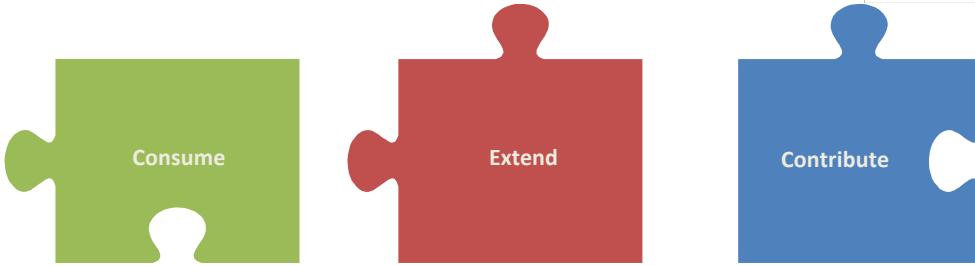
Leadership roles in the Project are also merit-based and earned by peer acclaim. Merit must be demonstrated in publicly-accessible forums. Committers and project leads are added to a project via an election.

Getting Involved



Learn

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



Consume

Extend

Contribute

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

Getting Started with ...



Open. Simple. Familiar.

- Project Community site
 - <https://zowe.org>
- Access to Beta Download
 - <https://zowe.org/download>
- Review Zowe squads, missions and activities
 - <https://zowe.org/contribute/>
- Code Guidelines
 - <https://zowe.org/code-guidelines/>
- Project Governance
 - <https://zowe.org/about-us/>
- GitHub
 - <https://github.com/zowe>
- Project Documentation (includes user and install guides)
 - <https://zowe.github.io/docs-site/>
- Developer Tutorials
 - <https://zowe.github.io/docs-site/guides/intro.html>
 - <https://developer.ibm.com/tutorials/zowe-step-by-step-tutorial/>



Community Slack Channels



Community Mailing Lists



Community Calendar



Community Meeting Minutes



Get involved in the Zowe community

Join Open Source Community @

<https://www.openmainframeproject.org/projects/zowe>

Participate in and contribute to the Zowe developer community at zowe.org

Learn how your organization can become a steward and supporter of this project with Open Mainframe Project membership at
openmainframeproject.org/about/join



Questions?

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

zowe.org