

WSO2con2025

From Friction to Flow : Streamlining Enterprise Software Development



Tishan Dahanayakage
Director & Head of Engineering, Choreo
WSO2



Bit of Psychology

The background is a deep space scene with a color gradient from orange-red on the left to dark blue and purple on the right. It is filled with numerous stars of varying sizes and brightness, some with prominent diffraction spikes. In the lower-left quadrant, there is a large, teal-colored planet with a thin, dark ring. In the lower-right quadrant, there is a smaller, purple-colored planet with a thin, dark ring.

Cognitive Fatigue

- Like any other tissue in human body, brain also feel fatigue
- Limited working memory
- Cognitive switching penalty



Flow State of Mind

- State with energized focus, deep involvement and enjoyment
- Neuroscience factors
 - Transient Hypofrontality
 - Dopaminergic Regulation
 - Gamma and Alpha wave synchronization
 - Suppression of amygdala



Current State of Enterprise Developers

The background is a vibrant space scene. It features a gradient from deep red on the left to dark blue on the right. Numerous stars of varying sizes and colors (white, yellow, orange) are scattered across the field. In the lower-left corner, there is a large, dark blue planet with a prominent ring system. In the lower-right corner, there is a smaller, reddish-brown planet, also with a ring system.



Developers spend **3-4**
hours daily on non-
core task

- 70% respondents of state of
IDevP report



COMMIT

Lead time for deployment is **several weeks**

- 68% respondents of state of IDevP report

Friction 01 - Configuring CI/CD



Configuring CI/CD (God Mode)

- CI - Build and upload
 - ⦿ To Github
 - ⦿ To Dockerhub
 - ⦿ To ACR
 - ⦿ To any artifact repo
- CD
 - ⦿ Shutdown server and apply patches
 - ⦿ Update deployment.yaml with new version
 - ⦿ Use :latest tag and re-trigger deployment
- Friction
 - ⦿ Switching to multiple systems
 - ⦿ Manual repetitive tasks



Configuring CI/CD (Superhuman Mode)

- Configure and use CI tools
 - ⦿ Github Actions, Circle CI, Argo workflows, Jenkins
- Configure and use CD tools
 - ⦿ Azure pipelines, ArgoCD, Flux, Spinnaker
- Friction
 - ⦿ Writing CI/CD configs
 - ⦿ Switching to other systems
 - ⦿ Repetitive configs



Configuring CI/CD (Human Mode)

- CI/CD native platform
 - ⦿ Automated CI/CD
 - ⦿ Simplified customizability
 - ⦿ Minimal or no configs
- Flow
 - ⦿ Developer just commit; platform handle CI/CD
 - ⦿ Move to testing directly



Builds

BUILD ID	REFERENCE	STATUS	TIME
12374193037 Initial Build	 c800fa39c	 Success	 3 months ago

Auto Build on Commit ☒

[Build Latest](#)

Set up

Build ID

12374193037 Latest

 3 months ago

Commit Details

Improve sms message

 c800fa39c

by  tishan89  tishan89

Auto Deploy on Build ☒

Endpoint Configurations 

[Configure & Deploy](#)



Friction 02- Dependency Discovery



Dependency Discovery (God Mode)

- Ask Miguel
- Check existing source code
- Use of unstructured collection of information
- Friction
 - ⦿ Obvious disruption to work
 - ⦿ Time consuming
 - ⦿ Search fatigue



Dependency Discovery (Superhuman Mode)

- Use structured manual listings
 - ⦿ Confluent
 - ⦿ Spreadsheet
- Use of heterogeneous portals
 - ⦿ API developer portals
 - ⦿ Cloud provider portals
- Friction
 - ⦿ Context switching
 - ⦿ Less searchability
 - ⦿ Limited filtering capabilities



Dependency Discovery (Human Mode)

- Use of automated dependency listings
 - ⦿ Single discovery portal
 - ⦿ Automated dependency addition
 - ⦿ Searching and filtering
 - ⦿ Context awareness
 - ⦿ Dev tool integration
- Flow
 - ⦿ Minimal distractions
 - ⦿ Less effort to search



Marketplace

Services Databases Configuration Groups

Search Services

Name

Name

Content

Label

All

Type

☒ Internal

☒ Third Party

Network Visibility

☒ Project

☒ Organization

☒ Public

Categories

☐ Business Management/ERP

☐ GenAI

☐ HR

U

utility-service-go

wsa2con2025

REST

[Version v1](#) [Status Published](#)

Organization, Public

25 minutes ago

G

greeting-service-go

dev-testing

REST

[Version v1](#) [Status Published](#)

Public

2 days ago

D

department-api-proxy-from

con2025-feature-testing

REST

[Version v1](#) [Status Prototype](#)

Public

3 days ago

G

Registration Management Service

WSO2Con2025

REST

[Version v1](#) [Status Prototype](#)

Public

3 days ago

G

greeting-service

testProxy

REST

[Version v1](#) [Status Prototype](#)

Public

3 days ago

A

appointment user mgt service

wsa2con2025

REST

[Version v1](#) [Status Published](#)

Organization

5 days ago

T

Twilio API

third-party

REST

[Version v1](#) [Status Prototype](#)

Public

6 days ago

C

Open AI GPT-4o

REST

[Version v1](#) [Third Party](#) [Status Published](#)

GenAI

Organization

8 days ago

1 - 8 of 158

Items per page

8

Marketplace

Services Databases Configuration Groups

Search Services

Name

Name

Content

Label

All

Type

☒ Internal

☒ Third Party

Network Visibility

☒ Project

☒ Organization

☒ Public

Categories

☐ Business Management/ERP

☐ GenAI

☐ HR

U

utility-service-go

was2con2025

REST

[Version v1](#) [Status Published](#)

[Organization, Public](#)

[25 minutes ago](#)

G

greeting-service-go

dev-testing

REST

[Version v1](#) [Status Published](#)

[Public](#)

[2 days ago](#)

D

department-api-proxy-from

con2025-feature-testing

REST

[Version v1](#) [Status Prototype](#)

[Public](#)

[3 days ago](#)

G

Registration Management Service

WSO2Con2025

REST

[Version v1](#) [Status Prototype](#)

[Public](#)

[3 days ago](#)

G

greeting-service

testProxy

REST

[Version v1](#) [Status Prototype](#)

[Public](#)

[3 days ago](#)

A

appointment user mgt service

was2con2025

REST

[Version v1](#) [Status Published](#)

[Organization](#)

[5 days ago](#)

T

Twilio API

third-party

REST

[Version v1](#) [Status Prototype](#)

[Public](#)

[6 days ago](#)

C

Open AI GPT-4o

REST

[Version v1](#) [Third Party](#) [Status Published](#)

[GenAI](#)

[Organization](#)

[8 days ago](#)

1 - 8 of 158

Items per page

8

Marketplace

Services Databases Configuration Groups

Search Services

Name
Name
Content
Label
All

Type

- ☒ Internal
- ☒ Third Party

Network Visibility

- ☒ Project
- ☒ Organization
- ☒ Public

Categories

- ☐ Business Management/ERP
- ☐ GenAI
- ☐ HR

U

utility-service-go

was2con2025

REST

Version v1 Status Published

Organization, Public

25 minutes ago

G

greeting-service-go

dev-testing

REST

Version v1 Status Published

Public

2 days ago

D

department-api-proxy-from

con2025-feature-testing

REST

Version v1 Status Prototype

Public

3 days ago

G

Registration Management Service

WSO2Con2025

REST

Version v1 Status Prototype

Public

3 days ago

G

greeting-service

testProxy

REST

Version v1 Status Prototype

Public

3 days ago

A

appointment user mgt service

was2con2025

REST

Version v1 Status Published

Organization

5 days ago

T

Twilio API

third-party

REST

Version v1 Status Prototype

Public

6 days ago

C

Open AI GPT-4o

REST

Version v1 Third Party Status Published

GenAI

Organization

8 days ago

1 - 8 of 158

Items per page 8

Marketplace

Services Databases Configuration Groups

Q

Search Services

Type

☒ Internal

☒ Third Party

Network Visibility

☒ Project

☒ Organization

☒ Public

Categories

☐ Business Management/ERP

☐ GenAI

☐ HR

U

utility-service-go

wsa2con2025

REST

Version v1

Status Published

Organization, Public

25 minutes ago

G

greeting-service-go

dev-testing

REST

Version v1

Status Published

Public

2 days ago

D

department-api-proxy-from

con2025-feature-testing

REST

Version v1

Status Prototype

Public

3 days ago

G

Registration Management Service

WSO2Con2025

REST

Version v1

Status Prototype

Public

3 days ago

G

greeting-service

testProxy

REST

Version v1

Status Prototype

Public

3 days ago

A

appointment user mgt service

wsa2con2025

REST

Version v1

Status Published

Organization

5 days ago

T

Twilio API

third-party

REST

Version v1

Status Prototype

Public

6 days ago

C

Open AI GPT-4o

REST

Version v1

Third Party

Status Published

GenAI

Organization

8 days ago

Name

↑

Name

Content

Label

All

1 - 8 of 158

Items per page

8

20

No summary provided

REST

Project: wso2con2025

Status: Published

[Overview](#) [API Definition](#) [How to Use](#) [Related Documents](#) [Download](#)

GET

/user Get a user



Greet a person by name

Parameters

NAME	DESCRIPTION
name string (query)	Name of the person to get name

Responses

CODE	DESCRIPTION	LINKS
200	Successful response	No links

Friction 03- Dependency/Configuration Usage



Why Dependencies/Configurations

Dependencies

- You need parameters to connect to dependencies
 - ⦿ Service URI
 - ⦿ Keys
- Typically environment bound
- **Represented using key-value pairs**
- DevOps/SRE knows the production values

Configurations

- You need parameters to configure the application
 - ⦿ Log level
 - ⦿ Thread count
- Typically environment bound
- **Represented using key-value pairs**
- DevOps/SRE knows the production values



Dependency Usage (God Mode)

- Hard code
 - ⦿ In source code
 - ⦿ In deployment configs
 - ⦿ Docker
 - ⦿ K8s deployment.yaml
- Friction
 - ⦿ Manual work involved
 - ⦿ Repetitive to handle multiple environments
 - ⦿ Human error can occur



Dependency Usage (Superhuman Mode)

- Use configuration management tools
 - ⦿ Template based
 - ⦿ Helm, Puppet, Ansible
 - ⦿ Overlay based
 - ⦿ Kustomize, Jsonnet
- Friction
 - ⦿ Dependent on DevOps/SRE
 - ⦿ Limited reusing
 - ⦿ Repeated process for multiple environments



Dependency Usage (Human Mode)

- Integrated configuration management
 - ⦿ Auto populated
 - ⦿ Declare using application configs
 - ⦿ No manual intervention from Ops
- Runtime service discovery
 - ⦿ Use a discovery API
 - ⦿ Fronts configuration stores
 - ⦿ Resolves at runtime
- Flow
 - ⦿ No or minimal intervention from Ops
 - ⦿ Allows seamless deployment and promotion
 - ⦿ Integration with minimal steps



`https://{org-id}-dataplane-handle/{project-handle}/{component-handle}/{version}`



Overview

Dependencies

Connections

Build

Deploy

Test

Manage

Insights

Observability

DevOps

Runtime

Containers

Configs & Secrets

Health Checks

Scaling

Storage

← Back to Connection Listing

Create a Connection

Connecting to

utility-service-go

REST

Open API

Name

utility service connection

Description (Optional)

Enter Description

Access Mode

Organization

Access the service with the organization-level endpoint. Authentication may be required.

Public

Access the service with the public endpoint. Authentication may be required.

Authentication Scheme

API Key

Authenticate using a simple, unique API key for direct access.

OAuth2

Generate and use an access token with client credentials for secure access

Cancel

Create

Overview

Dependencies

Connections

Build

Deploy

Test

Manage

Insights

Observability

DevOps

Runtime

Containers

Configs & Secrets

Health Checks

Scaling

Storage

← Back to Connection Listing

Create a Connection

Connecting to

utility-service-go

REST

Open API

Name

utility service connection

Description (Optional)

Enter Description

Access Mode

Organization

Access the service with the organization-level endpoint. Authentication may be required.

Public

Access the service with the public endpoint. Authentication may be required.

Authentication Scheme

API Key

Authenticate using a simple, unique API key for direct access.

OAuth2

Generate and use an access token with client credentials for secure access

Cancel

Create

Overview

Dependencies

Connections

Build

Deploy

Test

Manage

Insights

Observability

DevOps

Runtime

Containers

Configs & Secrets

Health Checks

Scaling

Storage

← Back to Connection Listing

Create a Connection

Connecting to

utility-service-go

REST

Open API

Name

utility service connection

Description (Optional)

Enter Description

Access Mode

Organization

Access the service with the organization-level endpoint. Authentication may be required.

Public

Access the service with the public endpoint. Authentication may be required.

Authentication Scheme

API Key

Authenticate using a simple, unique API key for direct access.

OAuth2

Generate and use an access token with client credentials for secure access.

CancelCreate

Overview

Dependencies

Connections

Build

Deploy

Test

Manage

Insights

Observability

DevOps

Runtime

Containers

Configs & Secrets

Health Checks

Scaling

Settings

[← Back to Connection Listing](#)

utility connection

Connecting to [utility-service-go](#)

Connection Schema Default API Key Connection - Organization

Service URL <https://1456f846-f4f9-4e8f-bb99-a461df78c2d4-dev-int>

Development

Production

Config key

Config Value

ServiceURL

<https://1456f846-f4f9-4e8f-bb99-a461df78c2d4-dev-int>

ChoreoAPIKey

[chk_eyJjYjZ5uZWN0aW9uLWlkjoiMDFmMDAxY2EtNTc2OC0xZT...](#)

How to Use the Connection

What is your source configuration file ?

The source configuration file is located within the `.choreo` directory at the root of the project directory.

component.yaml v1.1

component.yaml v1.0

component-config.yaml

Introduction

> Create programmatic connections to other services from your component [\(See more\)](#)

Add Connection Configuration

To integrate the mentioned configurations into your application, copy and paste the following snippet into the [component.yaml file](#).

```
dependencies:
  connectionReferences:
    - name: utility connection
      resourceRef: service:/wso2con2025/utility-service-go/v1/803f0/ORGANIZATION
```

The name represents the name of your connection, and the `resourceRef` represents a readable service identifier unique to the service you are connecting to.

The table below details the Choreo-defined environment variables.

Configuration Key	Choreo-Defined Environment Variable Name
ServiceURL	CHOREO_UTILITY_CONNECTION_SERVICEURL
ChoreoAPIKey	CHOREO_UTILITY_CONNECTION_CHOREOAPIKEY

> Manage connection references [\(See more\)](#)

Using the Connection in Application

> Use the connection configurations within the application [\(See more\)](#)

Overview

Dependencies

Connections

Build

Deploy

Test

Manage

Insights

Observability

DevOps

Runtime

Containers

Configs & Secrets

Health Checks

Scaling

Settings

[← Back to Connection Listing](#)

utility connection

Connecting to [utility-service-go](#)

Connection Schema Default API Key Connection - Organization

Service URL <https://1456f846-f4f9-4e8f-bb99-a461df78c2d4-dev-int>

Development

Production

Config key Config Value

ServiceURL <https://1456f846-f4f9-4e8f-bb99-a461df78c2d4-dev-int>ChoreoAPIKey [chk_eyJjb25uZW50aW9uLWlkjoiMDFmMDAxY2EtNTc2OC0xZT...](#) [Regenerate](#)

How to Use the Connection

What is your source configuration file ?

The source configuration file is located within the `.choreo` directory at the root of the project directory.

component.yaml v1.1

component.yaml v1.0

component-config.yaml

Introduction

> Create programmatic connections to other services from your component [\(See more\)](#)

Add Connection Configuration

To integrate the mentioned configurations into your application, copy and paste the following snippet into the [component.yaml file](#).

```
dependencies:
  connectionReferences:
    - name: utility connection
      resourceRef: service:/wso2con2025/utility-service-go/v1/803f0/ORGANIZATION
```

The name represents the name of your connection, and the `resourceRef` represents a readable service identifier unique to the service you are connecting to.

The table below details the Choreo-defined environment variables.

Configuration Key	Choreo-Defined Environment Variable Name
ServiceURL	CHOREO_UTILITY_CONNECTION_SERVICEURL
ChoreoAPIKey	CHOREO_UTILITY_CONNECTION_CHOREOAPIKEY

> Manage connection references [\(See more\)](#)

Using the Connection in Application

> Use the connection configurations within the application [\(See more\)](#)

[← Back to Connection Listing](#)

utility connection

Connecting to [utility-service-go](#)

Connection Schema Default API Key Connection - Organization

Service URL <https://1456f846-f4f9-4e8f-bb99-a461df78c2d4-dev-int>Development

Production

Config key	Config Value
ServiceURL	https://1456f846-f4f9-4e8f-bb99-a461df78c2d4-dev-int
ChoreoAPIKey	chk_eyJjb25uZWNoZW9uLWlkjoiMDFmMDAxY2EtNTc2OC0xZT... Regenerate

How to Use the Connection

What is your source configuration file ?

The source configuration file is located within the `.choreo` directory at the root of the project directory.

[component.yaml v1.1](#)[component.yaml v1.0](#)[component-config.yaml](#)

Introduction

> Create programmatic connections to other services from your component [\(See more\)](#)

Add Connection Configuration

To integrate the mentioned configurations into your application, copy and paste the following snippet into the [component.yaml file](#).

```
dependencies:
  connectionReferences:
    - name: utility connection
      resourceRef: service:/wso2con2025/utility-service-go/v1/803f0/ORGANIZATION
```

The name represents the name of your connection, and the `resourceRef` represents a readable service identifier unique to the service you are connecting to.

The table below details the Choreo-defined environment variables.

Configuration Key	Choreo-Defined Environment Variable Name
ServiceURL	CHOREO_UTILITY_CONNECTION_SERVICEURL
ChoreoAPIKey	CHOREO_UTILITY_CONNECTION_CHOREOAPIKEY

> Manage connection references [\(See more\)](#)

Using the Connection in Application

> Use the connection configurations within the application [\(See more\)](#)

DEFAULT PROJECT

+

🔄

...

🔍

🔗

📁

🔌

🧪

☰

📦

🏠

🔄

📄

🔍

Welcome

inventory-svc

×

Service

inventory-svc

...

Path: vscode-extension-v2-smoke-tests/service/s/go

Service

payment-svc

...

Path: vscode-extension-v2-smoke-tests/service/s/bal

Service

auth-svc

...

Path: vscode-extension-v2-smoke-tests/service/s/bal

inventory-svc Service

Deployment Track: master | Project: Default Project | Organization: Kajandran Alagan

Open in Console Open Git Repository

Builds

🔄

Build ID

Commit

20000008837

bcc8ce2b

Development Environment

🔄

Status

Not Deployed

Staging Environment

🔄

Status

Not Deployed

Production Environment

🔄

Status

Not Deployed

Connection Guide

×

Introduction

> Create programmatic connections to other services from your component (Show more)

Add Connection Configuration

To integrate the mentioned configurations into your application, copy and paste the below snippet into the component yaml file.

dependencies:

connectionReferences:

- name: auth-svc

resourceRef: service:/inaae/auth-svc/v1/803f8/PROJECT

The name represents the name of your connection, and the resourceRef represents a readable service identifier unique to the service you are connecting to.

The table below details the Choreo-defined environment variables.

Configuration Key	Choreo-Defined Environment Variable Name
ServiceURL	CHOREO_AUTH_SVC_SERVICEURL
ChoreoAPIKey	CHOREO_AUTH_SVC_CHOREOAPIKEY

> Manage connection references (Show more)

Read Configurations

> Read configurations within the application (Show more)

// sample go code snippet

import "os"

serviceurl := os.Getenv("CHOREO_AUTH_SVC_SERVICEURL")

choreoapikey := os.Getenv("CHOREO_AUTH_SVC_CHOREOAPIKEY")

Invoke the service

> Invoke the service using the resolved service URL (Show more)

// sample go code snippet

req, err := http.NewRequest("GET", serviceURL, nil)

if err != nil {

// Handle error

}

req.Header.Add("Choreo-API-Key", choreoApiKey)

client := &http.Client{}

response, err := client.Do(req)

34

Step 2/4 Environment Configurations



+ Add a Configuration

Link a Configuration Group

common logging config Configuration Group

Unlink



ENVIRONMENT VARIABLE ?

CONFIG KEY NAME

LOG_LEVEL



log_level

Back

Next

configurations:

file:

- name: application.properties
mountPath: /src/resources
type: yaml
values:
 - name: thread.pool.size
valueFrom:
 - configForm:
 - displayName: Thread Pool Size
 - required: true
 - name: request.timeout
valueFrom:
 - configForm:
 - displayName: Request Timeout
 - required: true

Step 1/4 Configurations

Required

Thread Pool Size string

10

Request Timeout string

5

Cancel Next

Friction 04 - Application Security

Application Security (God Mode)

- Private user stores
 - ⦿ Each app maintains user tables
 - ⦿ Token based on credential validation
 - ⦿ User onboarding is per application
- Friction
 - ⦿ Time to implement log in processes
 - ⦿ Dealing with user on-boarding



Application Security (Superhuman Mode)

- Use an IdP
 - ⦿ With API management layer for services and API
 - ⦿ With suitable libraries for web apps
- Friction
 - ⦿ UI/context switching to enable security
 - ⦿ Additional work to add API representation to APIM Layer
 - ⦿ API definition inconsistencies
 - ⦿ Code changes to integrate security(web apps)



Dependency Usage (Human Mode)

- Use an identity native platform
 - ⦿ Treats identity as first class
 - ⦿ Security is integrated to the CD flow
 - ⦿ Minimal or no code changes to engage security
- Flow
 - ⦿ No or minimal context switching
 - ⦿ No additional work
 - ⦿ Consistent API definitions



Security Scheme

Security Header

Description



API Key

choreo-api-key

Secure your API with API Key protocol.



OAuth2

Authorization



Secure your API with OAuth 2 protocol.



Pass end-user attributes to upstream ?

Resources



Search Operations

GET

/users/{email}



Security ?



1 - 1 of 1



Items per page

5



Managed Authentication with Choreo

Choreo manages authentication for your application.

[↗ Configuration Guide](#)

Setup Managed Authentication

Post Login Path ?

/

Post Logout Path ?

/

Error Path ?

/



```
window.configs = {  
  apiUrl: '/choreo-apis/wso2con2025/utility-service-go/v1',  
};
```



```
<button onClick={() => {window.location.href="/auth/login"}}>Login</button>
```



```
const apiUrl = window?.configs?.apiUrl ? window.configs.apiUrl : "/";  
const response = await axios.get(apiUrl/{RESOURCE_PATH});
```



The background is a cosmic scene with a color gradient from bright orange-red on the left to deep blue and purple on the right. It is filled with numerous stars of varying sizes and brightness, some with prominent four-pointed diffraction patterns. In the lower-left, there is a large, dark blue planet with a thin, light blue ring. In the lower-right, there is a smaller, reddish-brown planet with a thin, dark red ring.

Why we should do it?

Gartner

According to report “Why and how developer experience drives outcomes”, teams with high quality developer experience are:

Quality

33%

More likely to attain their business outcomes

Efficiency

31%

More likely to improve delivery flow

Skill Retention

20%

More likely to intent to stay with their employer



Overview

Components

Dependencies

Insights

Usage

Operational

Delivery

Business

Cost

Observability

DevOps

Performance for **Customer Success**

1M

3M

6M

1Y

View by Day



Deployment Frequency

Elite

3/week (12 total)

Lead Time for Change

High

5 days 23 hours

Deployment Frequency



Lead Time for Change



Question Time!

