



The bridge to possible

Webex Contact Center – Data and Analytics Lab

Wafik Hert – Customer Success Manager

Anuj Bhatia – Technical Leader, CX

LTRCCT-2011

cisco Live!

#CiscoLive

Contact Center Operational Concerns

Customer Satisfaction and Experience

Long Wait times

Operational Costs and Efficiency

Employee burnout

Keeping pace with evolving customer expectations

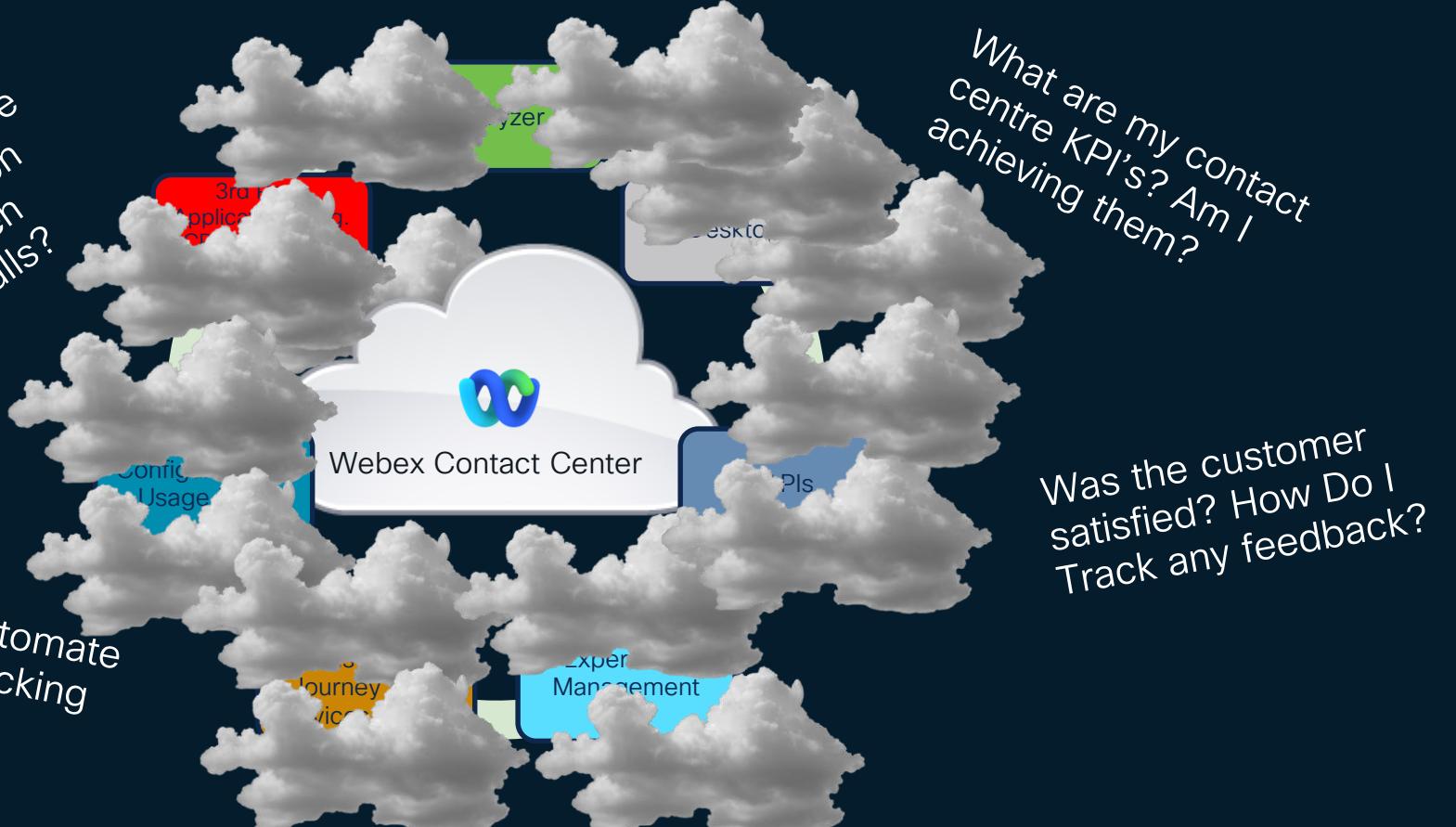
Customer Insights and Data Utilization

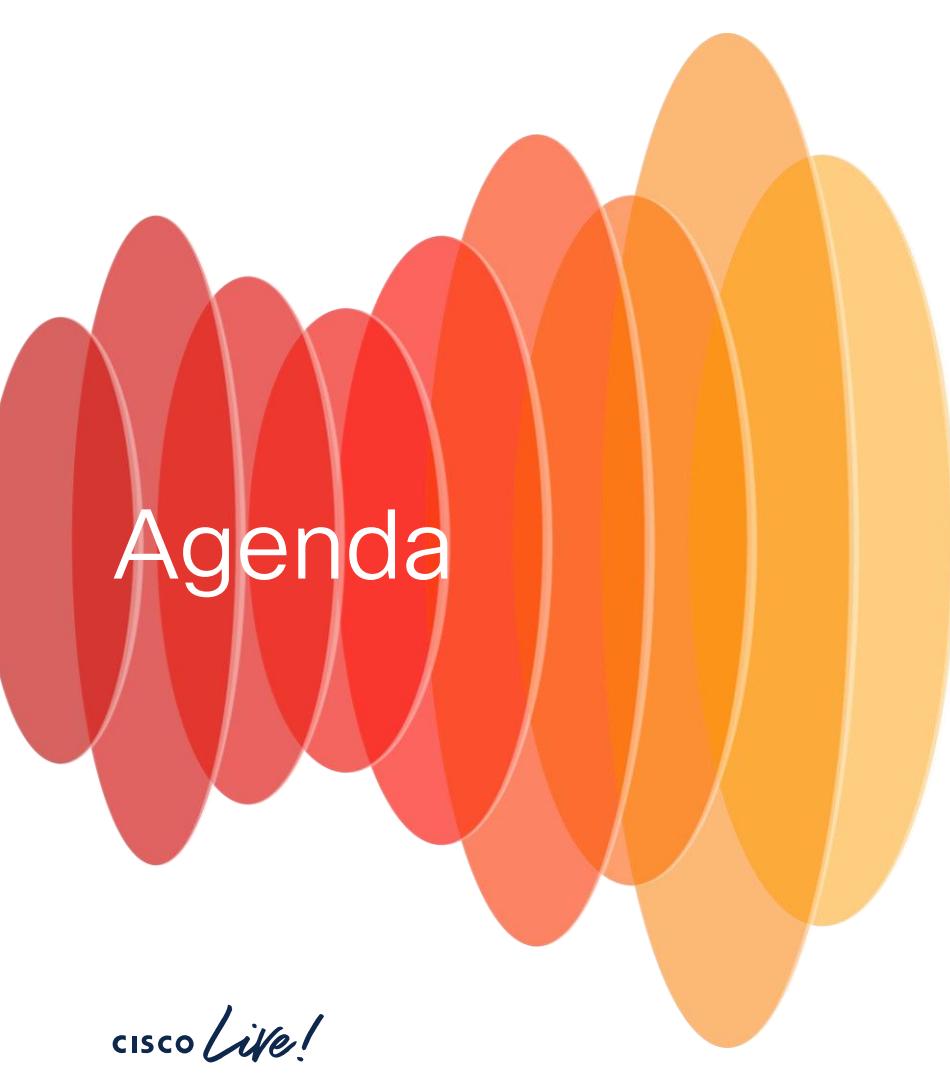


This is you (before this Lab)

Do my agents have all the information they need when answering calls?

How, Can I automate my metrics tracking process?





Agenda

- Webex CC & Lab Introduction
- Analyzer
- Desktop
- APIs
- Experience Management
- Journey Data Services
- Configuration & Usage Data
- Closing Thoughts

And this is us!



Wafik Hert



welherte@cisco.com



[Whert](#)



Anuj Bhatia



anubhati@cisco.com



[Anuj-Bhatia](#)



Cisco Webex App

Questions?

Use Cisco Webex App to chat with the speaker after the session

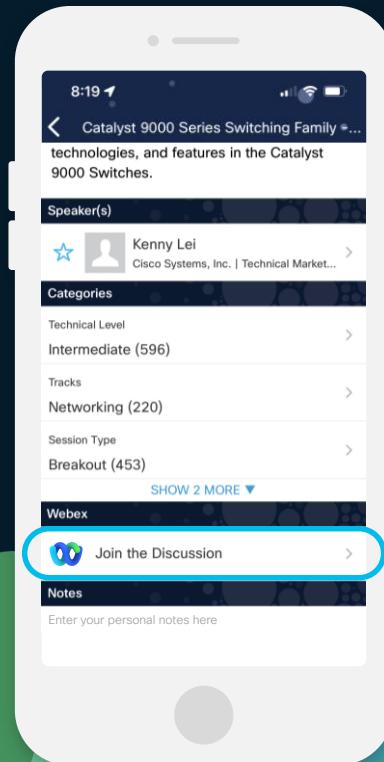
How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

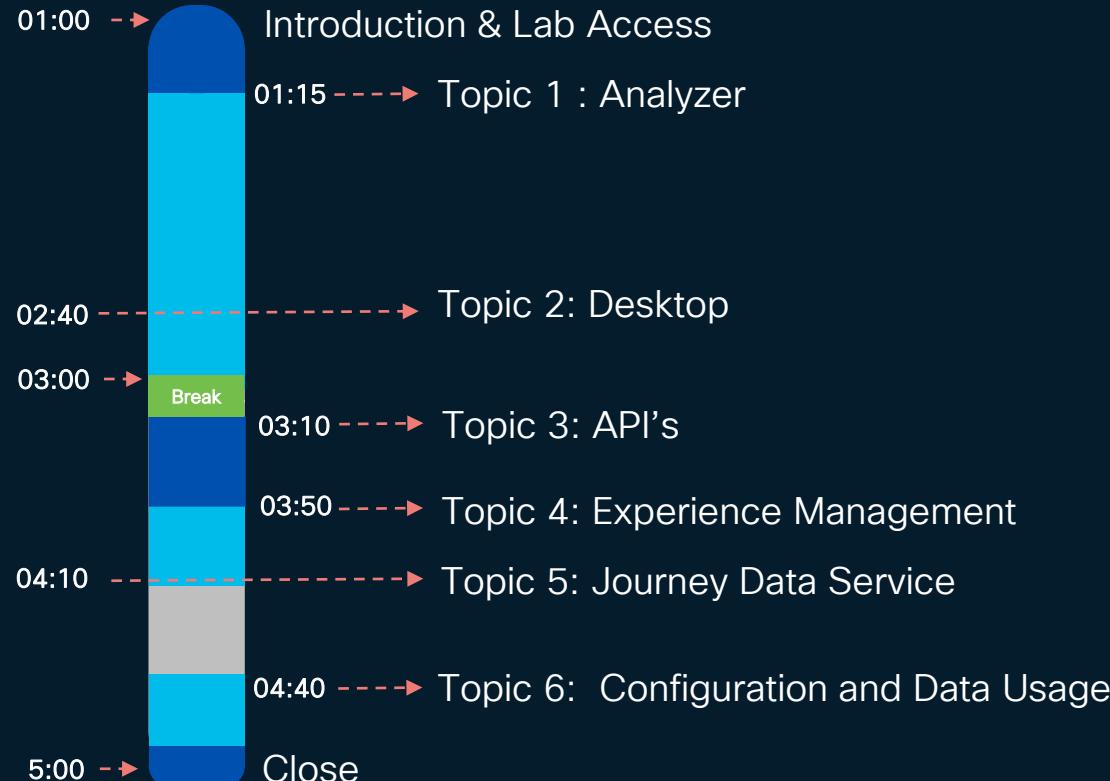
Webex spaces will be moderated by the speaker until June 7, 2024.

CISCO *Live!*

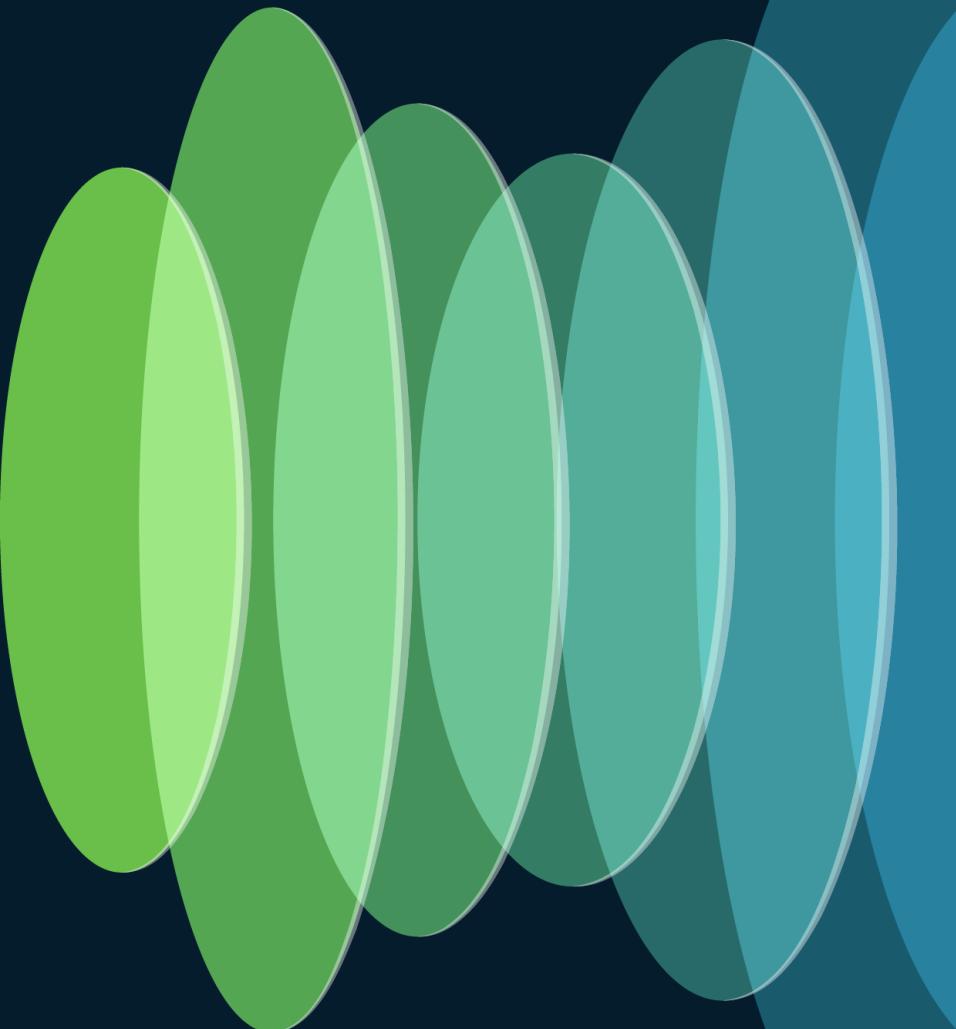
[https://ciscolive.ciscoevents.com/
ciscoalivebot/#LTRCCT-2011](https://ciscolive.ciscoevents.com/ciscoalivebot/#LTRCCT-2011)



Topic Timeline



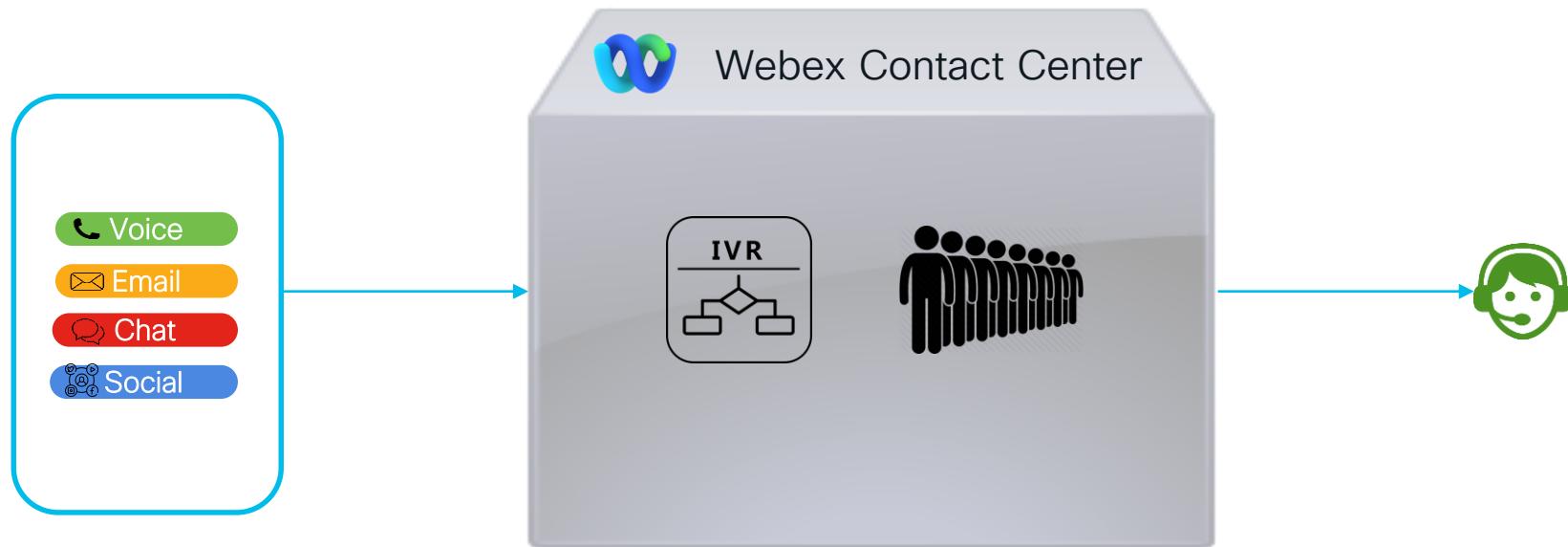
Webex CC & Lab Introduction



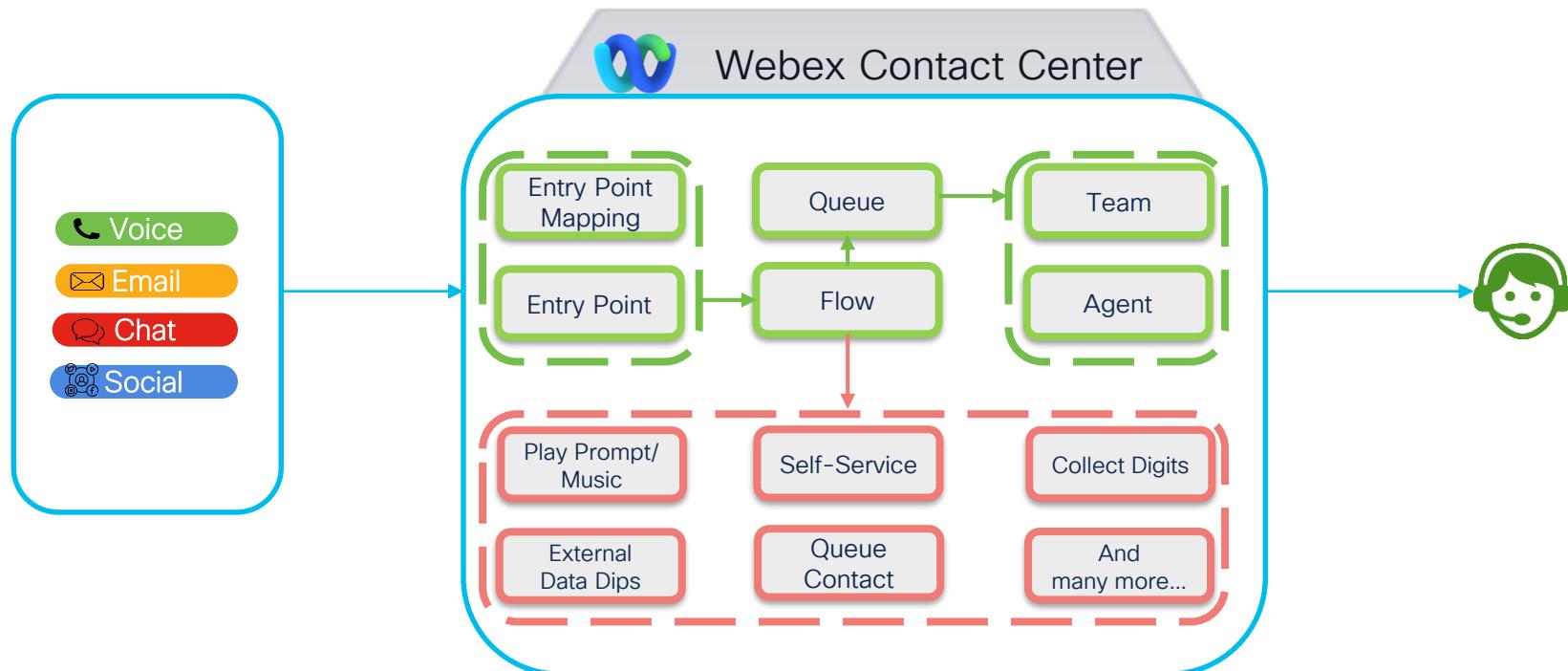
Contact Center from 20,000 ft.



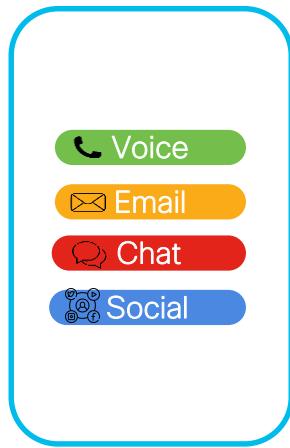
Contact Center from 10,000 ft. : Customer



Contact Center from 5,000 ft. : Contact Center



Contact Center from 2,000 ft. : The Portals



Webex Contact Center

Control Hub

Flow Designer

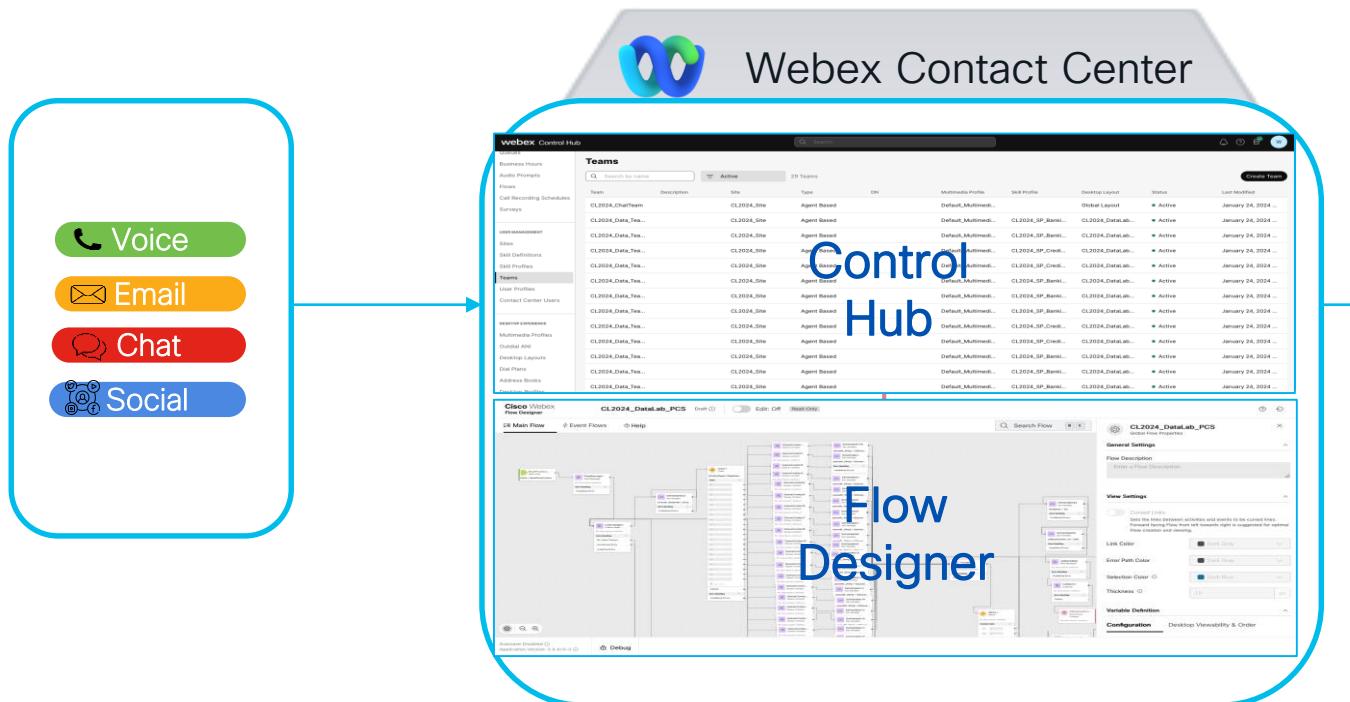
Analyzer

Desktop

Portals

- **Control Hub:** Configuration portal, used by administrators or supervisors to create & edit configuration objects in the Webex Contact Center tenant.
- **(Management Portal):** Older configuration portal (same configuration as Control Hub), goal is to move all configuration to Control Hub and slowly decommission this portal.
- **Flow Designer:** Interface for administrators to design the logic paths for call handling & contact routing.
- **Desktop:** Agent (or Supervisor) interface which handles incoming and outgoing contacts.
- **Analyzer:** Main reporting tool for Webex Contact Center, where most of the customer and agent related data reside.

Contact Center from 3.1415 ft. : The Live Call



CISCO *Live!*

#CiscoLive

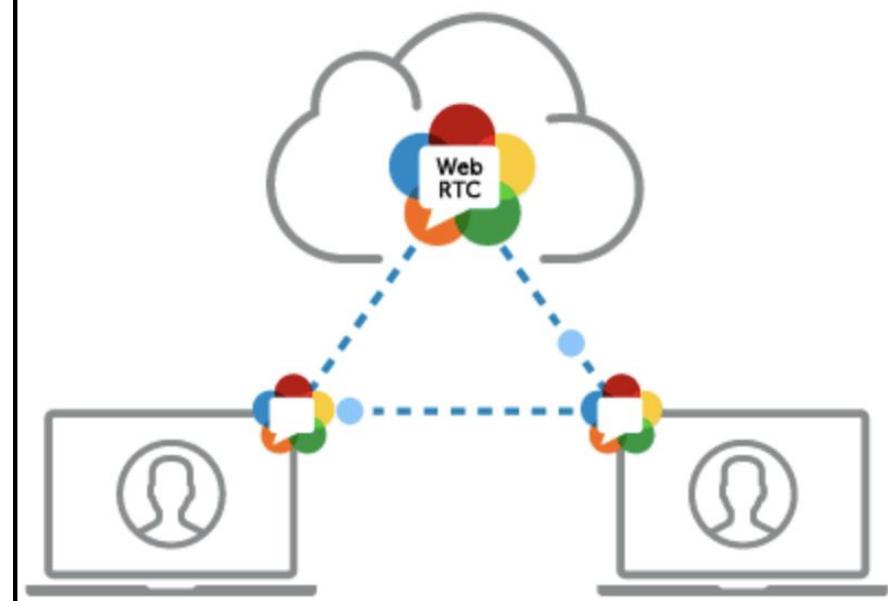
LTRCCT-2011

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

14

WebRTC

- WebRTC (Web Real-Time Communication) provides direct communication between two browsers (peer-to-peer connection).
- It allows to transmit **audio & video** as well as **a data channel** for real-time data exchange between the peers.
- For Webex Contact Center, WebRTC today allows agents to log in Desktop without using an external device (e.g. mobile, Webex) and handle calls directly from the browser.
- Video, WebRTC on Customer leg on the roadmap.



Webex CC KPIs

- Key Performance Indicators (KPIs) are measures of performance for a specific objective.

Contact Metrics

- ❖ Service Level, Abandon Rate
- ❖ Queue Wait Time, Avg Handle Time, Avg Speed of Answer
- ❖ IVR, Connected, Consult, Transfer Count & Durations
- ❖ # of Callbacks, Avg # of Callback Retries, Status
- ❖ Sudden Disconnect, Short Calls (Termination Types)
- ❖ % Calls Opt-Out Recording

Agent Metrics

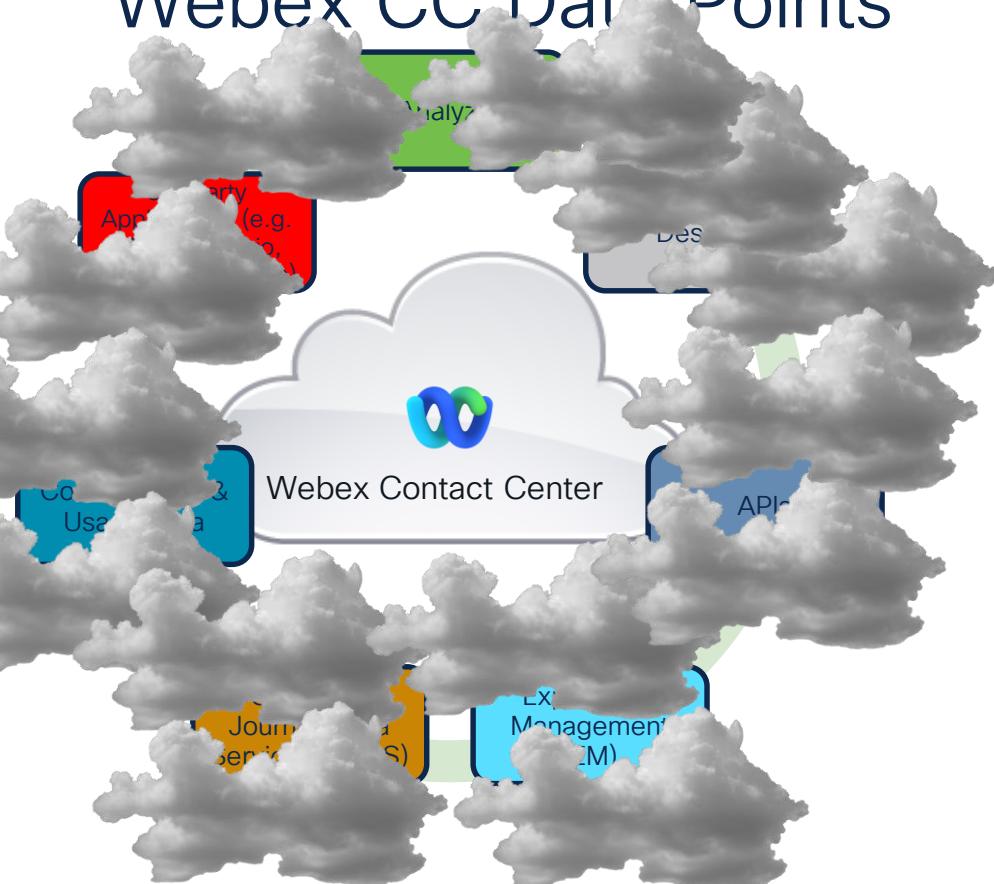
- ❖ Occupancy / Staff Hours
- ❖ Available, Idle, Wrap-Up Counts & Durations
- ❖ Connected, Consult, Transfer, Conference, Hold Counts & Durations
- ❖ # of RONA Calls
- ❖ Avg NPS of Contacts Handled

Customer Metrics

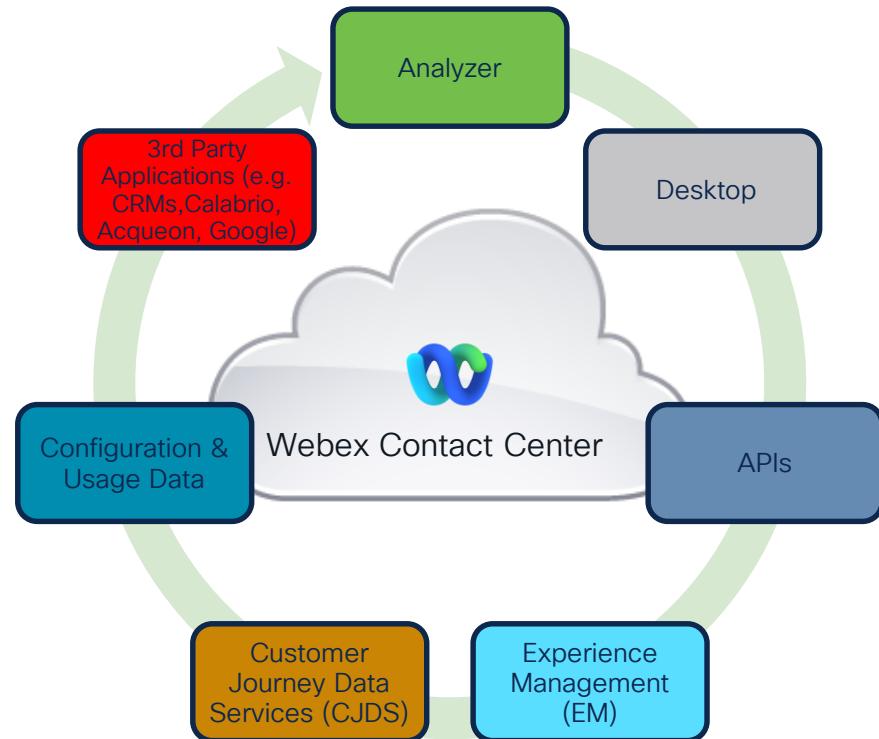
- ❖ CES, NPS, CSAT, Surveys
- ❖ Avg. Queue Waiting Time, Avg. Call Handle Time
- ❖ # of Calls in an Interval (e.g. Day)
- ❖ # of Callback Requests, # of Avg. Callback Retries

Segmentation possible on Site, Queue, Team, Agent, Channel and custom levels

Webex CC Data Points



Webex CC Data Points



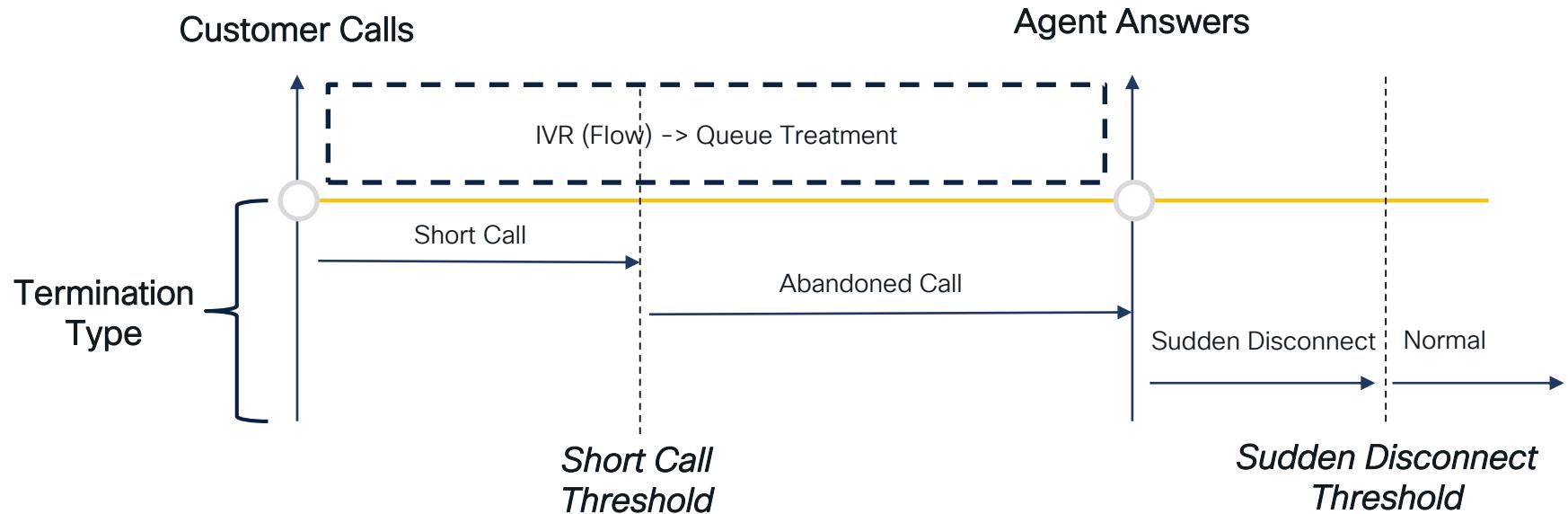
- **Data Point:** A place where information is stored or presented to a specific data user persona.
- Multiple data points can be accessible under the same portal (e.g. Control Hub).
- Today, we will discuss all the data points **besides** 3rd party applications.

Webex CC KPIs Dictionary

- **Service Level:** Indicates if a call was answered within the queue configured threshold level. The Service Level timer starts counting from the moment a call enters a queue for the first time until it is answered by an agent (or abandoned).
- **Handle Time:** The total duration of the call from the moment it was answered by an agent until completion.
Formula: Connected Duration + Hold Duration + WrapUp Duration
- **Speed of Answer:** Indicates how long it took to answer a call from the moment it was queued. Formula: Queue Duration + Ringing Duration
- **RONA (Route-On-No-Answer):** A RONA call is a call that rang to an agent's device but it was not answered by the agent within the tenant configured timer. After that, call would be placed back in queue for next available agent.
- **Staff Hours:** The total duration an agent was logged in the Contact Center. Formula: Available + Idle + Ringing + Connected + Consult + Hold + WrapUp + Not Responding Duration
- **Occupancy:** The percentage of time the agent was in an active call during a shift. Formula: Total Connected + Total Hold + Total WrapUp Duration / Staff Hours
- **CES/NPS/CSAT:** Score indicators that measure a customer's experience and satisfaction with a product or service. Feedback is usually gathered through the form of customer surveys.

Webex CC Termination Types

- Termination Type specifies how a call was terminated based on what point of the call journey the contact was ended as well as the threshold timers configured by the administrators. Administrators can configure three threshold timers: (1) Sudden Disconnect Threshold (2) Short Call Threshold (3) Service Level Threshold. In the diagram below, you can see what is the expected termination type based on the point the call was ended.



4 + 1 Data User Personas

Agent

- Do I have all the necessary information to handle the call?
- Who is my customer? What is their contact history with us?
- How are my teammates doing? Do we need additional resources?

Supervisor

- How is my team doing? Do I need to provide additional resources?
- Are we handling my team's calls quickly & efficiently?
- Are the customers satisfied with my team?

Administrator

- Are all calls handled properly or are there any issues to troubleshoot?
- How can I provide my teams with the necessary information and metrics to handle the calls and assess their performance?
- How do we capture customer information & feedback for future use?

Analyst

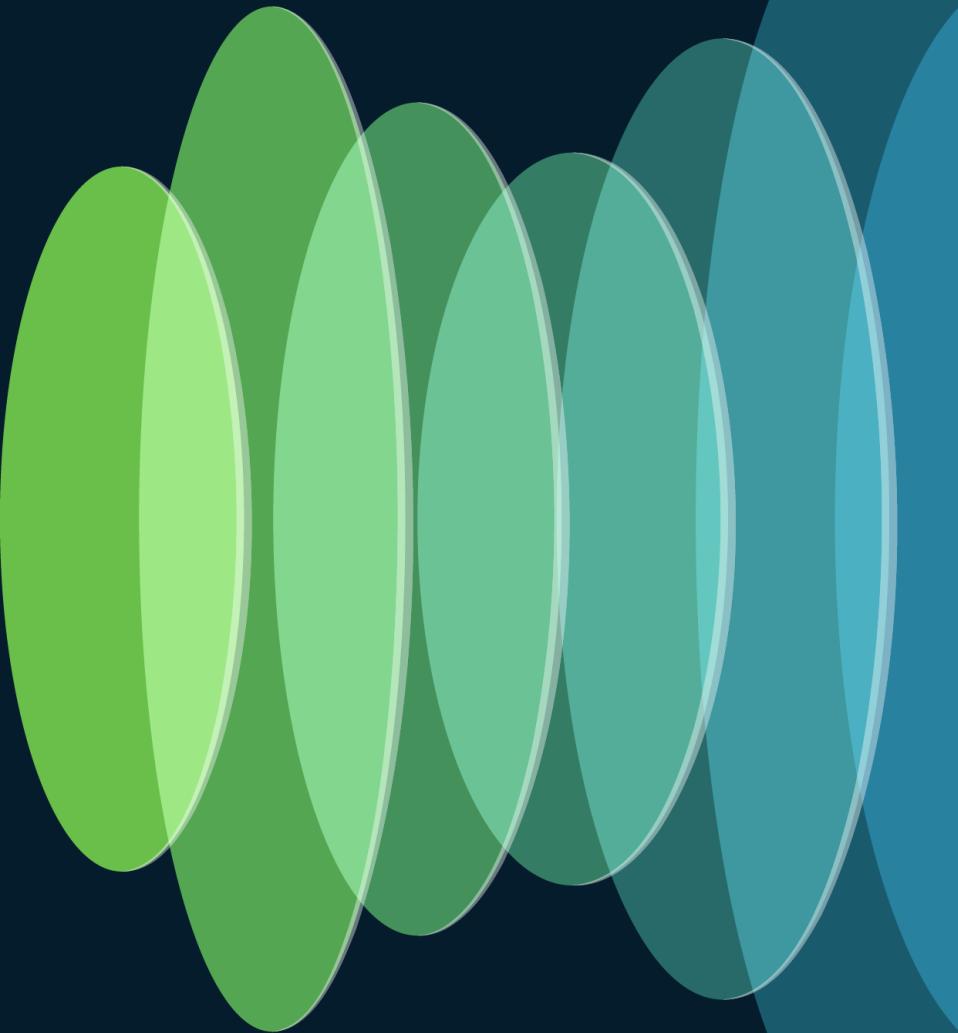
- Are we handling all calls quickly & efficiently?
- Are the customers satisfied with our service?
- With what actions can we improve both the agent productivity and the customer satisfaction?

Programmer

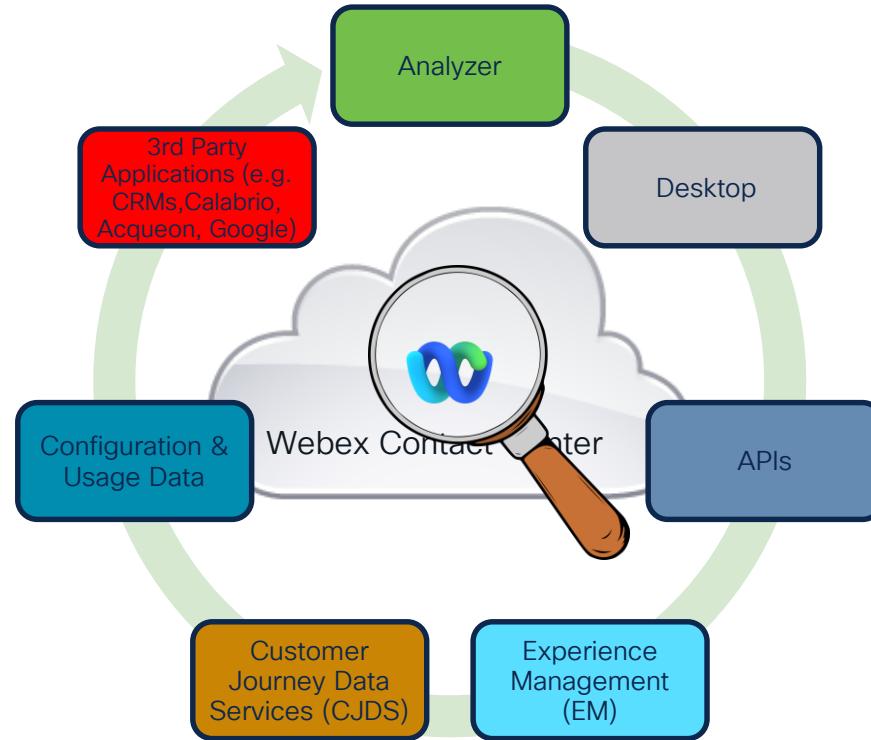
- Do I have the necessary tools from the product to do my job?
- What automations can I create to support the other roles in their daily tasks?

[Learn more about
Webex CC Personas](#)

Analyzer

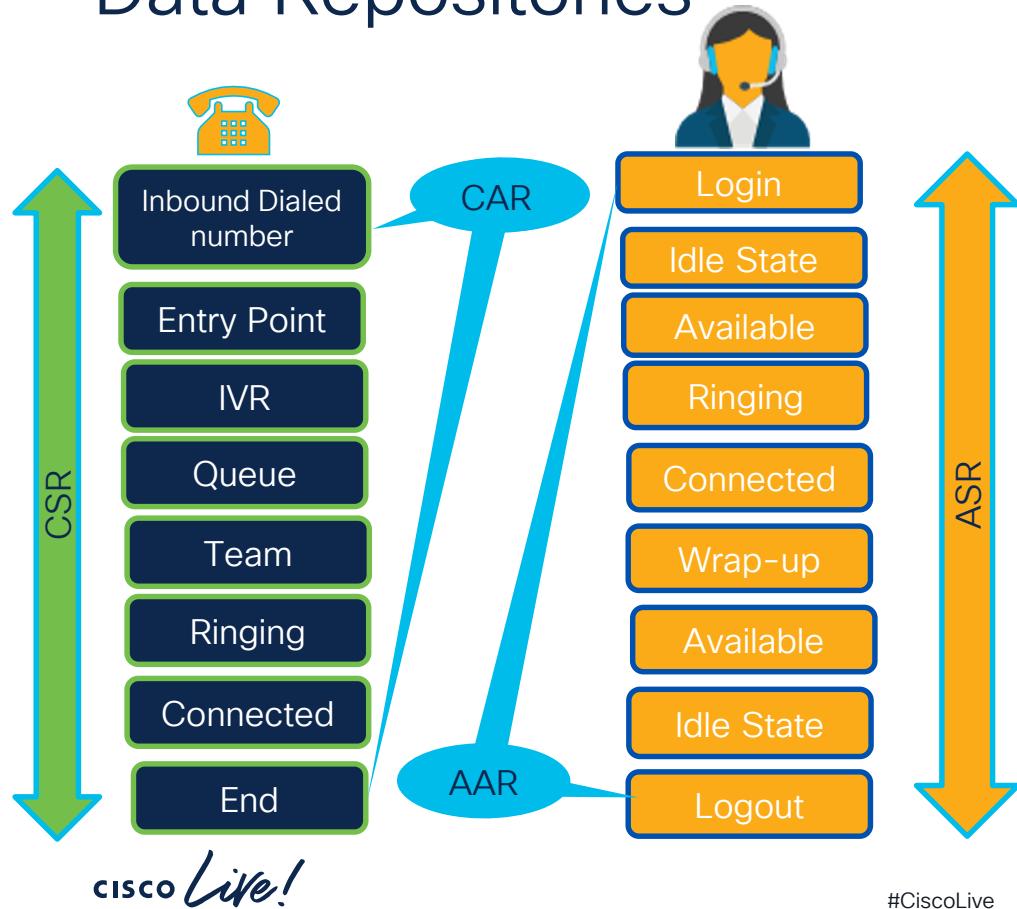


What is Analyzer?



- **Analyzer** is the main reporting tool of Webex Contact Center.
- **Analyzer** allows users to create visualizations (=reports) in various formats, combine them in dashboards and schedule report distribution via email.
- **Analyzer** provides both customer & agent data
- **Analyzer** contains a plethora of Stock reports to quickly onboard in addition to the option of creating a custom report from scratch.

Data Repositories



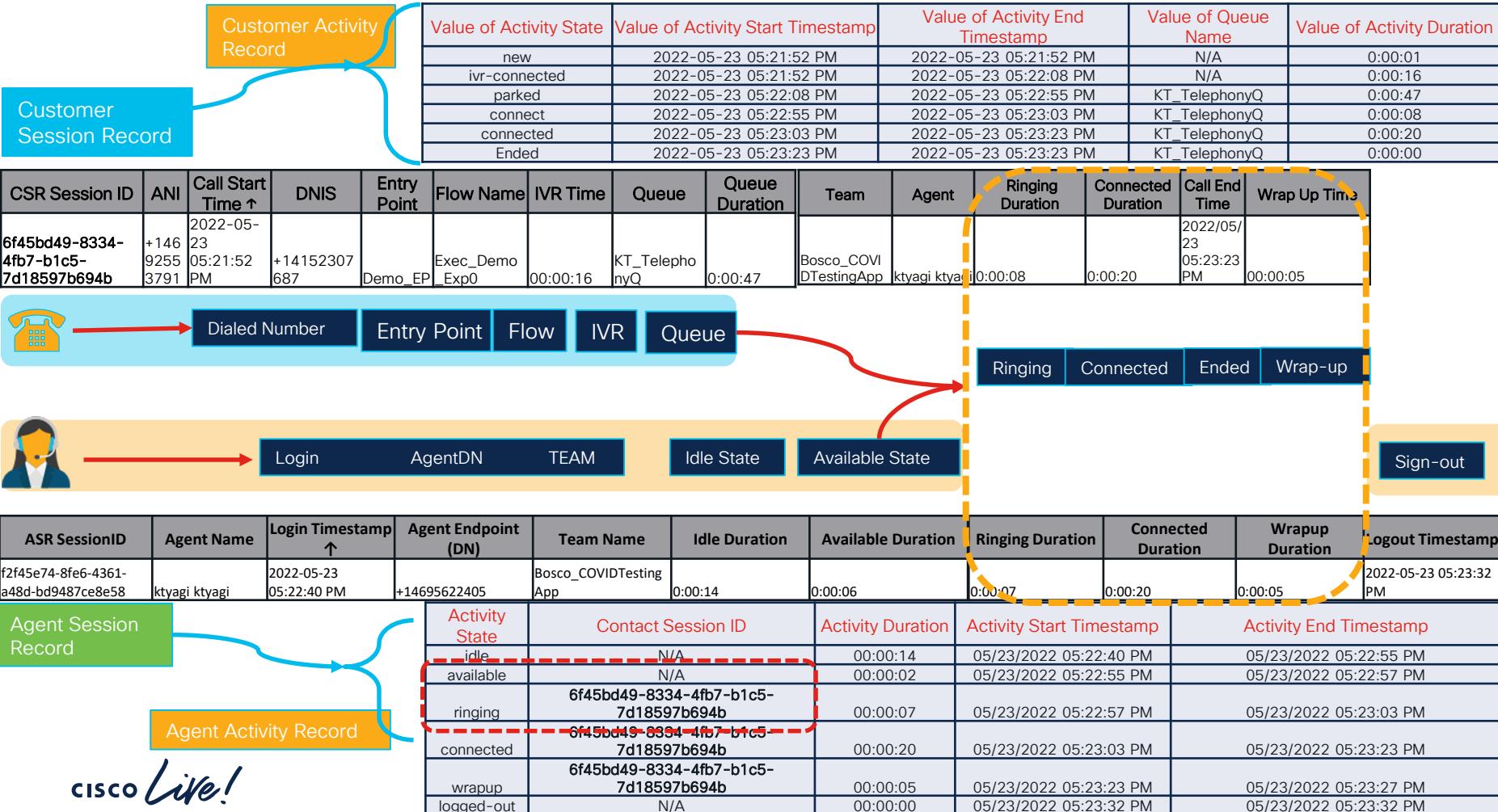
Customer

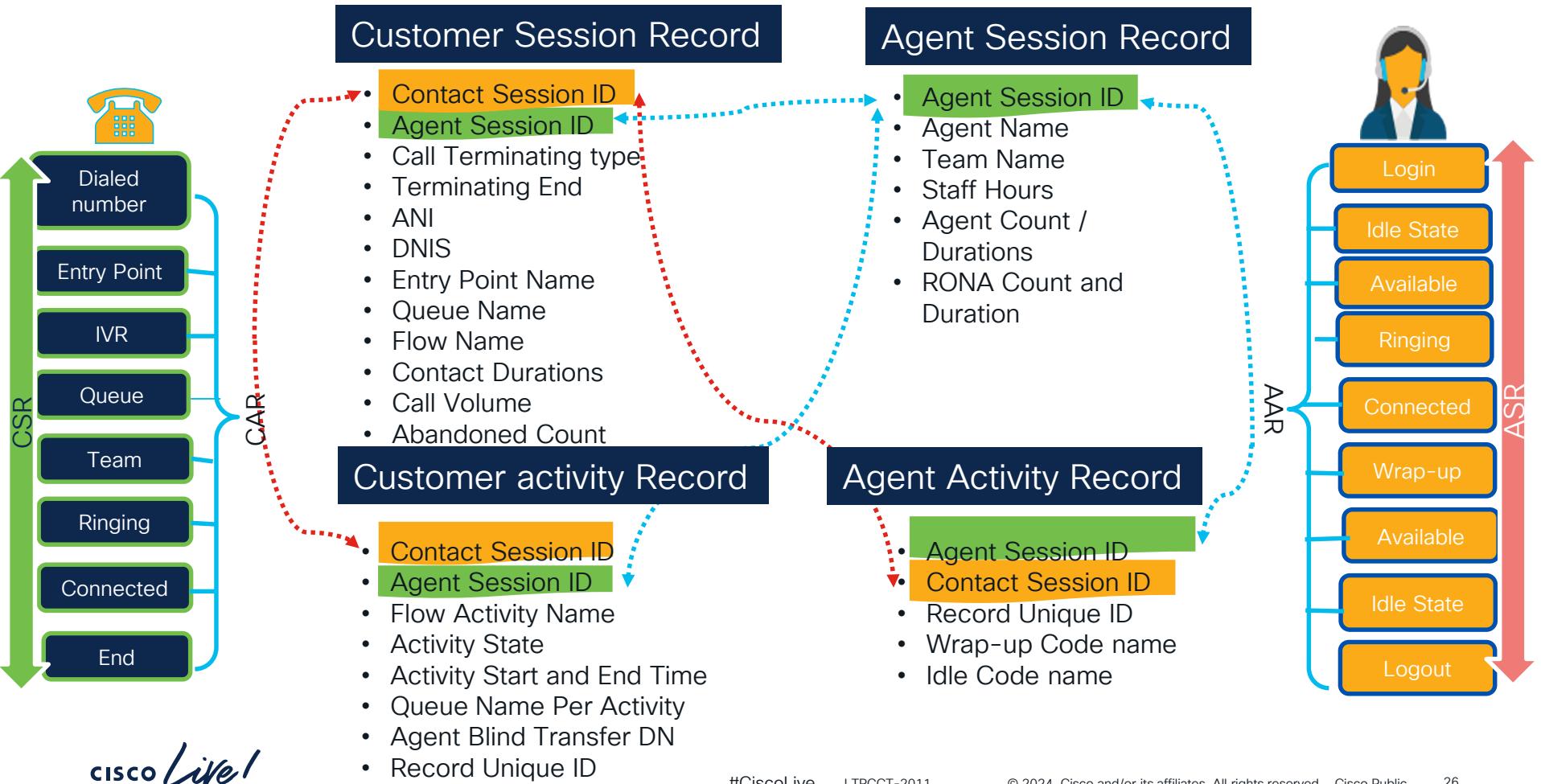
- Customer Session Record (CSR)
- Customer Activity Record (CAR)

Agent

- Agent Session Record (ASR)
- Agent Activity Record (AAR)

Data Creation and Storage





Data Schema

REFERENCE

You can find all the CSR and CAR fields with their description [here](#).

Standard CSR and CAR Fields and Measures

Customer Session Repository (CSR)

The standard fields and measures aggregated in the CSR are described in the following sections:

Column Name	Description	Field or Measure	Data Type
Is Current Session	Flag that indicates whether the session is an active session. Supported values are 0 and 1. The value 1 indicates that the session is active.	Measure	Integer
Activity Span	The amount of time, in seconds, that an agent was engaged in the activity during the specified interval.	Measure	Long
Agent Endpoint (DN)	The endpoint (number, e-mail, or chat handle) on which an agent receives calls, chats, or emails.	Field	String
Agent ID	A string that identifies an agent.	Field	String
Agent Leg Blob ID	String identifier for the blob that contains the recording of the agent's side of the call.	Field	String
Call Direction	Indicates if the call is an inbound call or an outbound call.	Field	String
Agent Login	Login name using which an agent logs in to the Agent Desktop.	Field	String
Agent Name	The name of the agent who answers customer calls, chats, and emails.	Field	String
Agent Session ID	A string that identifies an agent's login session.	Field	String
Agent System ID	A string that identifies an agent.	Field	String
ANI	ANI digits delivered with a call.	Field	String
Note Automatic Number Identification (ANI) is a service provided by the phone company that delivers the caller's phone number along with the call.			
Blind Transfer Count	The number of times a call was transferred by an agent to another agent or an external DN (Dial Number) via a blind transfer.	Measure	Integer
Contact End Timestamp	The time when the contact ended.	Measure	Long
Call Progress Detection	In case of outdial call, this field represents the CPD value returned from telephony.	Field	String

Standard ASR and AAR Fields and Measures

Agent Session Repository (ASR)

The standard fields aggregated in the ASR are described in the following table:

 **Note** The Disconnected Count field is currently not used and not populated in the ASR.

Column Name	Description	Field or Measure	Data Type
Is Current Session	Flag that indicates whether the session is an active session. Supported values are 0 and 1. The value 1 indicates that the session is active.	Measure	Integer
Activity Span	The amount of time in seconds that an agent was engaged in the activity during the specified interval.	Measure	Long
Agent Endpoint (DN)	The endpoint (number, e-mail, or chat handle) on which an agent receives calls, chats, or emails.	Field	String
Agent ID	A string that identifies an agent.	Field	String
Agent Login	The login name using which an agent logs in to the Agent Desktop.	Field	String
Agent Name	Name of an agent, that is, a person who answers customer calls/chats/emails.	Field	String
Agent Session ID	A string that identifies an agent's login session.	Field	String
Agent System ID	A string that identifies an agent.	Field	String
Logout Timestamp	The time when an agent logged out.	Measure	Long
Login Timestamp	The time when an agent logged in.	Measure	Long
Channel Type	The media type of the contact, such as telephony, email, or chat.	Field	String
Session Count	The number of agent sessions.	Measure	Integer
Current State	The current state of an agent.	Field	String
Historical Update Timestamp	The time when the historical process updated the record.	Measure	Long

Lab 1: Analyzer



10 min

Exercise 1.1: Analyzer User Interface

What did we learn?

- Navigate through the Analyzer Interface
- Change the report Timezone
- Access the Help Page
- Search a visualization by Name or ID
- Access Control

Lab Configuration

- Each student is given a unique ID to use for their logins, for each ID 3 users are created, an **Administrator**, a **Supervisor** & an **Agent** (they share same password).
 - Administrator: wxcclabs+admin_ID<StudentID>@gmail.com
 - Supervisor: wxcclabs+supvr_ID<StudentID>@gmail.com
 - Agent: wxcclabs+agent_ID<StudentID>@gmail.com
- Data already available to complete all exercises, test calls/chats are optional.
- Most of the features are pre-configured. Information for configurations in Appendix.
- Focus, Demo & Bonus Exercises
- Calling Devices to use:
 - Customer: Personal Device
 - Supervisor: Webex (with Webex Calling), already logged in Webex
 - Agent: WebRTC 
- Ensure that you are able to login into [Control Hub](#), [Desktop](#) and [Analyzer](#). Chrome is suggested to utilize profiles.

Lab Documents

Access Controls

Complete Information on permissions and access controls can be found in the [Analyzer User guide](#), under Introduction > Access Controls.

Resources	Roles	Restrictions
<ul style="list-style-type: none">• APS Reports on the Agent Desktop• Management Portal Dashboard• Browser Links	Administrators and supervisors with Cisco Contact Center disabled or with no associated Agent Profiles	Restrictions applied are based on the User Profiles
<ul style="list-style-type: none">• APS Reports on the Agent Desktop• Management Portal Dashboard• Browser Links	Administrators, supervisors with associated Agent Profiles and all agents	Restrictions applied for: <ul style="list-style-type: none">• Entry Points are based on the User Profiles• Queues are based on the Agent Profiles• Sites are based on the Agent Settings• Teams are based on the Agent Profiles
<ul style="list-style-type: none">• Analyzer Reports• Filters inside the Visualization create or edit page	All administrators and supervisors	Restrictions applied are based on the User Profiles

Types of Reports

Historical / Realtime

- Historical reports retrieve the data for a lengthy time range up to 12 months, within Last 13 months. e.g. **last 7 days**.
- Realtime reports provide a snapshot of the current status of the tenant looking back from a few minutes up to the start of the day. Realtime reports provide the option to refresh automatically every few seconds.

*Note: Both the time period and the type of data in a report are defined at the start of its creation and **cannot** be changed afterwards.*

Count / Value Based

- Count Based reports provide aggregated data (e.g. count, sum, avg) of the requested events in the report (*e.g. # of Agents in available state*)
- Value Based reports provide raw data for each unique activity in the report (*e.g. Name of each agent in available state*)

Stock Reports

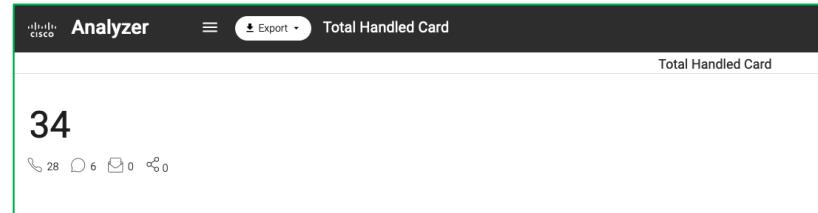
Stock Report



100+ Stock Reports!

Stock Reports (Simplified)

- Can be run directly or can be used as building blocks for your tailored custom visualizations.
- Use a simple search word (e.g. “Queue”, “abandoned”, “idle”)
- Many stock reports provide the same data with a different segmentation (e.g. “Site Idle Auxiliary”, “Team Idle Auxiliary”, “Agent Idle Auxiliary”)
- Card format stock reports



Stock Reports “Cheat Sheet”

- **Service Level & Queue Statistics** → CSQ All Fields Report (-1265), Contact Details in Queue (-1209), Contact Details in Queue – Today Real-Time (-1212), Queue Service Level Realtime (-152)
- **Self Service Statistics** → IVR & CVA Dialog Flow (-1251)
- **Abandoned Statistics** → Abandoned Contact Details (-1184), Queue Abandoned (-124), Abandoned Realtime (-148), Abandoned Call Detail Activity Report (-1258)
- **Agent/Team Status Tracking & Call Handling** → Agent Details (-100), Team (-105), Agent Trace (-108), Agent Interval Realtime (-1190)
- **Agent Not Ready Statistics** → Agent Idle Auxiliary (-109)
- **Callback Statistics** → Callback Report (-1249)
- **Troubleshooting Report (Call-By-Call)** → CSR Report – Yesterday (-1189)
- **NEW More (?) Queue Statistics** → Queue All Fields Report (-1268), queue Activity by Queue (-1269)

We “lied” to you – The 5th Repository (Queue Record)

Customer Session Record

Session ID	ANI	DNIS	Queue	Site	Team	Agent	Call Start Time ↑	Call End Time	Call Duration	IVR Time	Queue Name
ca3817cf-2b02-4d93-be06-36d46c190820	+14692553791	+15084337864	Q_Admin	WCCAnalyzerLab-Site	Team -1	cladm1user c	02/03/23 05:12:43	02/03/23 05:14:05	00:01:23	00:07	N/A

Customer Activity Record

Contact Session ID	Record Unique ID	Activity State	Entrypoint Name	Queue Name	Activity Start Timestamp	Activity End Timestamp
ca3817cf-2b02-4d93-be06-36d46c190820	ca3817cf-2b02-4d93-be06-36d46c190820-1675419163075-new	new	EP_WCCAnalyzerLab	N/A	02/03/2023 05:12:43 AM	02/03/2023 05:12:43 AM
ca3817cf-2b02-4d93-be06-36d46c190820	ca3817cf-2b02-4d93-be06-36d46c190820-1675419163507-treatment	ivr-connected	EP_WCCAnalyzerLab	N/A	02/03/2023 05:12:43 AM	02/03/2023 05:12:50 AM
ca3817cf-2b02-4d93-be06-36d46c190820	ca3817cf-2b02-4d93-be06-36d46c190820-1675419170776-connect	connect	EP_WCCAnalyzerLab	SBR_QV_Team1	02/03/2023 05:12:50 AM	02/03/2023 05:13:11 AM
ca3817cf-2b02-4d93-be06-36d46c190820	ca3817cf-2b02-4d93-be06-36d46c190820-1675419191192-con-to-agent-error	con-to-agent-error	N/A	SBR_QV_Team1	02/03/2023 05:13:11 AM	02/03/2023 05:13:11 AM
ca3817cf-2b02-4d93-be06-36d46c190820	ca3817cf-2b02-4d93-be06-36d46c190820-1675419207676-dequeued	dequeued	EP_WCCAnalyzerLab	Q_Admin	02/03/2023 05:13:27 AM	02/03/2023 05:13:27 AM
ca3817cf-2b02-4d93-be06-36d46c190820	ca3817cf-2b02-4d93-be06-36d46c190820-1675419217311-connect	connect	EP_WCCAnalyzerLab	Q_Admin	02/03/2023 05:13:37 AM	02/03/2023 05:13:43 AM

Queue Record

Value of Contact Session ID ↓	Value of CallLeg ID	Value of CallLeg Start Timestamp	Value of CallLeg End Timestamp	Value of Queue Name	Value of Agent Name	Value of Handle Type	Value of Service Level Threshold	Value of Is CallLeg Handled	Value of Is Within Service Level
ca3817cf-2b02-4d93-be06-36d46c190820	ca3817cf-2b02-4d93-be06-36d46c190820-callLeg-1675419163075	02/03/2023 05:12:43 AM	2023-02-03 05:13:27 AM	SBR_QV_Team1	Krishna Tyagi	dequeued	30.0	0.0	0.0
ca3817cf-2b02-4d93-be06-36d46c190820	ca3817cf-2b02-4d93-be06-36d46c190820-callLeg-1675419207676	02/03/2023 05:13:27 AM	2023-02-03 05:14:05 AM	Q_Admin	cladm1user c	normal	60.0	1.0	1.0

Queue Based Reporting

Challenge: Using CSR repository, we did not have access to critical duration and field metrics with respect to each queue a contact went through, as all information was aggregated on last queue.

- With Queue-Based Reporting (QBR), we are allowing for high visibility of reporting attributes at each single Queue level.
- 3 new stock reports added to familiarize yourselves with the new repository.

Total Queue Records
3,311

Interval	Queue Name	Contact Session ID	Calls Presented	New Calls Handled	Calls Abandoned	Calls Moved Out of Queue
11/06/2023 (3)	QUEUE_DATA (1)	249811f7-6a34-4ccb-8ccf-9e36dd19d410 (1)	1	1	0	0
	Queue_1 (1)	249811f7-6a34-4ccb-8ccf-9e36dd19d410 (1)	1	1	0	0
	Queue_Data_2 (1)	249811f7-6a34-4ccb-8ccf-9e36dd19d410 (1)	1	0	1	0

Interval	Queue Name	Contact Session ID	Calls Presented	New Calls Handled	Calls Abandoned	Calls Moved Out of Queue
11/06/2023 (3)	QUEUE_DATA (1)	249811f7-6a34-4ccb-8ccf-9e36dd19d410 (1)	1	1	0	0
	Queue_1 (1)	249811f7-6a34-4ccb-8ccf-9e36dd19d410 (1)	1	1	0	0
	Queue_Data_2 (1)	249811f7-6a34-4ccb-8ccf-9e36dd19d410 (1)	1	0	1	0

Home > Visualization > Stock Reports > Historical Reports > Queue Reports

Visualizations

Queue Activity by Queue
ID-1269
Temporal Scope Historical
Created By Cisco
Modified Nov 24, 2023 11:40:40 PM

Queue All Fields Report
ID-1268
Temporal Scope Historical
Created By Cisco
Modified Nov 24, 2023 11:40:40 PM

Queue Call Distribution Summary
ID-1270
Temporal Scope Historical
Created By Cisco
Modified Nov 24, 2023 11:40:40 PM

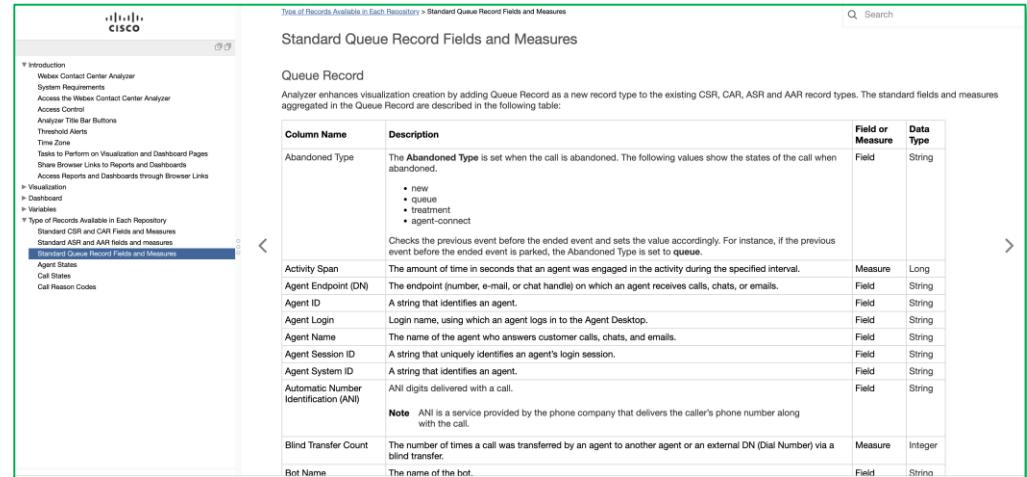
Major Metrics Introduced in QBR

- Calls Presented → Count of Queue Calls + Count of (Inbound) Consult To Queue + Count of Outdial Consult To Queue
- Calls Handled → Count of Connected Calls + Consult To Queue Handled Calls + Outdial Consult To Queue Handled Calls – Blind Transfer To Agent Calls – Agent Transferred In Calls
 - Blind Transfer To Agent Calls → The number of times an agent did a blind transfer to another agent
 - Agent Transferred In Calls → The number of times an agent did a consult followed by a transfer to an Entry Point, Queue or Agent (*Note: Metric does not increment in case of consult to Dialed Number*).
- Calls Abandoned → Count of Calls where handleType = abandoned
- QBR introduced few new handle type options:
 - TransferToDN → A call was transferred by an agent to a Dial Number (DN).
 - Dequeued → A call was moved out of a queue before being answered.

Data Schema (QBR)

You can find all the QBR fields with their description in the Help Page of Analyzer.

Since the feature is not yet Globally Available (GA), the Analyzer User guide does not contain this information yet, but you can find it by launching the Help page from Analyzer and searching for “Standard Queue Records Fields and Measures”.



The screenshot shows a browser window with the Cisco logo at the top. The main content area displays the 'Type of Records Available in Each Repository' help page, specifically the 'Standard Queue Record Fields and Measures' section. The title 'Standard Queue Record Fields and Measures' is centered above a table. The table has two columns: 'Column Name' and 'Description'. The 'Column Name' column lists fields like 'Abandoned Type', 'Activity Span', 'Agent Endpoint (DN)', etc., while the 'Description' column provides detailed explanations for each field. A search bar is visible at the top right of the page.

Column Name	Description	Field or Measure	Data Type
Abandoned Type	The Abandoned Type is set when the call is abandoned. The following values show the states of the call when abandoned. <ul style="list-style-type: none"> new queue treatment agent-connect Checks the previous event before the ended event and sets the value accordingly. For instance, if the previous event before the ended event is parked, the Abandoned Type is set to queue.	Field	String
Activity Span	The amount of time in seconds that an agent was engaged in the activity during the specified interval.	Measure	Long
Agent Endpoint (DN)	The endpoint (number, e-mail, or chat handle) on which an agent receives calls, chats, or emails.	Field	String
Agent ID	A string that identifies an agent.	Field	String
Agent Login	Login name, using which an agent logs in to the Agent Desktop.	Field	String
Agent Name	The name of the agent who answers customer calls, chats, and emails.	Field	String
Agent Session ID	A string that uniquely identifies an agent's login session.	Field	String
Agent System ID	A string that identifies an agent.	Field	String
Automatic Number Identification (ANI)	ANI digits delivered with a call.	Field	String
Note:	ANI is a service provided by the phone company that delivers the caller's phone number along with the call.		
Blind Transfer Count	The number of times a call was transferred by an agent to another agent or an external DN (Dial Number) via a blind transfer.	Measure	Integer
Bot Name	The name of the bot.	Field	String

Skill-Based Reporting (SBR)

- Skill-Based Reporting provides insight on the assigned skills both for a contact as well as for an agent.
- From a call perspective, we are able to see both the skill requirements of the call and the skills of the agent that ultimately answered the call.

- CSR / CAR / Queue Record Variables

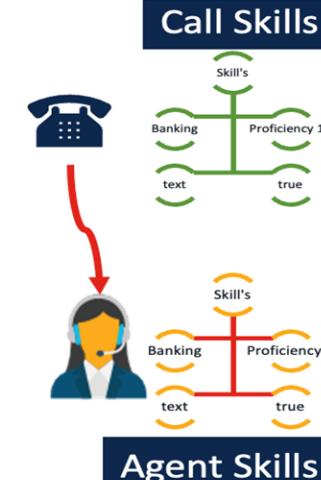
- Required Skills
- Matched Skills
- Skill Profile

- Similarly, we can see the skill profile & skills with which an agent logged in their shift:

- ASR / AAR Variables

- Skill Profiles
- Skills

Required Skills	Matched Skills	Skill Profile	Call #
▼ Loans=Auto loan (1)	▼ Loans=Auto loan (1)	▼ SkillProfile-GT1 (1)	2
▼ OS-Expert=Linux (1)	▼ OS-Expert=Linux (1)	▼ SkillProfile-GT1 (1)	1



Agent Skills	Skill Profile	Agent Name
▼ BooleanSkill-Expert-LIC Policy=True, French=8, Text=Test123 (1)	▼ GT2 (1)	▼ Agent& Chorus (1)
▼ Chinese=7, English-GT=7, French-GT=7, Enum-GT="Loans, Sales", Loans=Auto loan, OS-Expert=Linux (1)	▼ SkillProfile-GT1 (1)	▼ chorus-org-caus11 (1)
▼ Chinese=7, English-GT=7, French-GT=7, Enum-GT="Loans, Sales", OS-Expert=Linux (1)	▼ SkillProfile-GT1 (1)	▼ chorus-org-caus11 (1)

Transition Reports

- Cisco provides a smoother transition for UCCX customers with a set of stock reports that resemble the look & feel of major UCCX reports.
- In total, **9 stock reports** are available:
 - Abandoned Call Detail Activity Report
 - Agent Call Summary Report
 - Agent Detail Report
 - Agent summary Report
 - Application Summary Report
 - CSQ Activity Report by Window Duration
 - CSQ Agent Summary Report
 - CSQ All fields Report
 - Multi Channel Agent Summary Report
- The reports are not enabled by default, but after request through your Cisco contact.



Drill-Down

- Allows expanding a specific value in the table to see all the records involved in the computation.
- Drilldown-ception is possible (drill-down inside a drill-down)
- After drill down, user can add additional columns for more insight by clicking the variables on the left.
- Great troubleshooting tool, two usual cases:
 - Drill-Down a count of contacts to the contacts that comprise it
 - Drill-Down a contact Session ID to see a call's full path

The screenshot displays three tables illustrating drill-down functionality:

- Contact Details in Queue:** Shows a summary of contacts across different queues. A blue arrow points from the 'Channel Type' column of the first row to the expanded list below.
- Drill Down (Top):** An expanded view of the 'telephony' channel type. It shows a list of contact session IDs (1-7) and their corresponding queue names and session IDs. A blue arrow points from the 'Final Queue Name' column of the second row to the expanded list below.
- Drill Down (Bottom):** An expanded view of the contact session ID 'dead3a8c-c9bf-4477-a7de-6...'. It shows detailed activity states (new, flow-started, flow-activity, ivr-connected, ivr-done) and next states (N/A, N/A, N/A, connected, ended) for each step. A blue arrow points from the 'Activity State' column of the fourth row to the expanded list below.

Contact Details in Queue

Interval	Channel Type	Queue Name	# Contacts	Avg Queue Wait Time	Longest Contact in Queue	# Abandoned Contacts
01/24/2024 (3)	telephony (3)	CL2024_CCAL_Queue (1)	4	00:00:18	00:00:37	3
		CL2024_SBR_QV_Team51 (1)	7	00:00:01	00:00:05	0
		CL2024_SBR_QV_Team52 (1)	1	00:00:00	00:00:00	0

Drill Down

ID	Channel Type	Final Queue Name	Contact Session ID
1	telephony	CL2024_SBR_QV_Team51	c55e48d3-3dc0-4994-d79e-afc317d243c3
2	telephony	CL2024_SBR_QV_Team51	743c405d-4f82-49b2-97f0-19242d5ce619
3	telephony	CL2024_SBR_QV_Team51	dead3a8c-c9bf-4477-a7de-6466c59592d2
4	telephony	CL2024_SBR_QV_Team51	ed686931-04f0-8914-b1a3-21193c2f
5	telephony	CL2024_SBR_QV_Team51	9232b78e-578d-452b-b1c5-e1939ae05da
6	telephony	CL2024_SBR_QV_Team51	daa918e-6653-4e29-baa9-673b5a97489c
7	telephony	CL2024_SBR_QV_Team51	5ef020d-1190-4c7b-b1d5-663155242f38

Drill Down

ID	Contact Session ID	Queue ID	Activity State	Next State	Endpoint N
1	dead3a8c-c9bf-4477-a7de-6...	N/A	new	ivr-connected	EP_CL2024_DataL
2	dead3a8c-c9bf-4477-a7de-6...	N/A	flow-started	N/A	EP_CL2024_DataL
3	dead3a8c-c9bf-4477-a7de-6...	N/A	flow-activity	N/A	EP_CL2024_DataL
4	dead3a8c-c9bf-4477-a7de-6...	N/A	ivr-connected	ivr-done	EP_CL2024_DataL
5	dead3a8c-c9bf-4477-a7de-6...	N/A	flow-activity	N/A	EP_CL2024_DataL
6	dead3a8c-c9bf-4477-a7de-6...	N/A	flow-activity	N/A	EP_CL2024_DataL
7	dead3a8c-c9bf-4477-a7de-6...	N/A	ivr-done	N/A	EP_CL2024_DataL
8	dead3a8c-c9bf-4477-a7de-6...	f00a6b7c-7d74-40fe-b8db-a0...	connect	connected	EP_CL2024_DataL
9	dead3a8c-c9bf-4477-a7de-6...	f00a6b7c-7d74-40fe-b8db-a0...	connected	ended	EP_CL2024_DataL
10	dead3a8c-c9bf-4477-a7de-6...	f00a6b7c-7d74-40fe-b8db-a0...	bnr-started	bnr-ended	EP_CL2024_DataL

Run-Mode Filtering

Allows user to filter and visualize the data without editing the report, dynamically updating the report results upon selection.

You can only select the following as run-mode filters:

- **Duration** (available by default for all reports)
- All the variables added as **Row Segment** in that report.

Maximum of 5 filters can be applied.

The screenshot shows a reporting interface with a green border. At the top, there are four dropdown menus: 'Queue Name' (set to 'All'), 'Channel Type' (set to 'All'), 'Service Level Configur...' (set to 'All'), and 'Duration' (set to 'Last 7 days'). Below this is a section titled 'Show Filter on Run Mode' with a dropdown arrow. It contains the instruction 'Choose upto 5 from below' and a list of five filter options, each with a checked checkbox: 'Duration', 'Queue Name', 'Channel Type', and 'Service Level Configured (in se...)'.

Global Variables in Reporting

- Global Variables are configured in Management Portal and can be used in the flows.
- They can be used to store information for flow operations, to store custom information for Reporting or to provide additional context to agents.
- To be visible in Analyzer, they need to be set as **Reportable** during configuration.
- They can be of type **String**, **Integer**, **Date Time**, **Boolean** or **Decimal**.
- We can have up to maximum 100 Global Variables set as Reportable in a tenant.

The screenshot shows the 'Global Variable' configuration page. Under 'General Settings', a variable named 'CL2024_StudentID' is defined as a String type with no default value. It is currently active. In the 'Reporting Settings' section, there is a 'Make Reportable' button set to 'Yes', which is highlighted with a red box. A tooltip below the button states: 'This enables the variable to appear in Analyzer for reporting purposes. Each tenant has a limit of 100 variables that can be reported on.'

Value of Contact Session ID	CL2024_StudentID
8fc21eab-2d61-49c7-b251-0502547e5dc2	51
53af22b6-a6b6-431a-bda7-bd5d50ec9f0a	51
79ea065d-e412-4908-b106-e585c0a46b36	51
947cc7b3-06ba-484a-8fc2-08c489324fd6	51
b3b89fc1-4ac9-461e-9230-7edfb804d366	51
d8f6ec43-8768-4936-8554-1c79971a8603	51
d3614edf-4d3a-4ede-a8af-722521e25774	51
b0db5b47-eacf-444d-aa7f-b25afb45baae	51
fafac343-ed0b-4c82-aa94-186f03399999	52

Lab 1: Analyzer



15 min

Exercise 1.2: Using Stock Reports & Extending their Capabilities

What did we learn?

- *How to find & run Stock reports based on our target KPIs*
- *Report options: Edit, Save-As, Schedule Report, Export*
- *Report Capabilities: Skill-Based Reporting, Omnichannel Reporting, Transition Reports, Queue-Based Reporting*
- *Report Functionalities: Run-Mode Filtering, Drill-down, Global Variables*

Anatomy of a Visualization

The screenshot shows a visualization configuration interface with the following elements:

- Top Bar:** Type Agent Session Record, Visualization - 01/31/2024 21:01:00, Save button.
- Left Sidebar (Module):** Modules, Module1, Start Time (Last 7 days), If run today, Start Date: 2024-01-24, End Date: 2024-01-30, Including (All Days), Compute (3), Show Filter on Run Mode, Add Filter.
- Middle Section:** Row Segments, Profile Variables, Column Segments, Profile Variables, Agent Name (4).
- Output Type:** Table, Show summary, Table level, Customize.
- Table Data:** Team Name, Agent Name, Average of Connected Count.

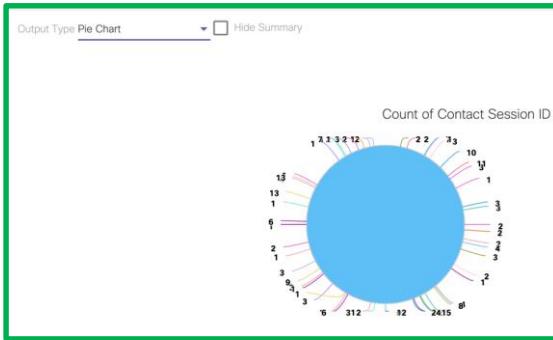
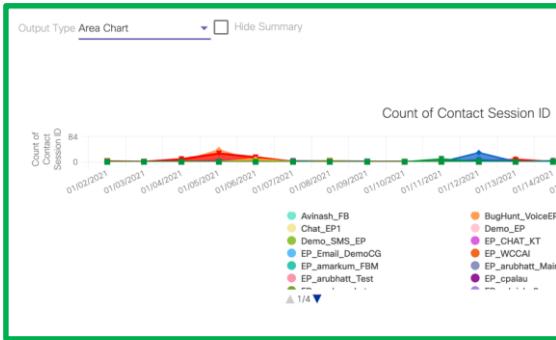
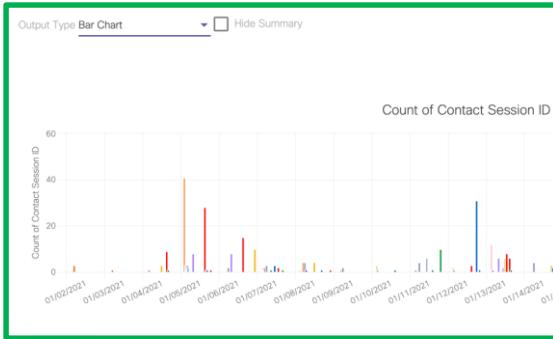
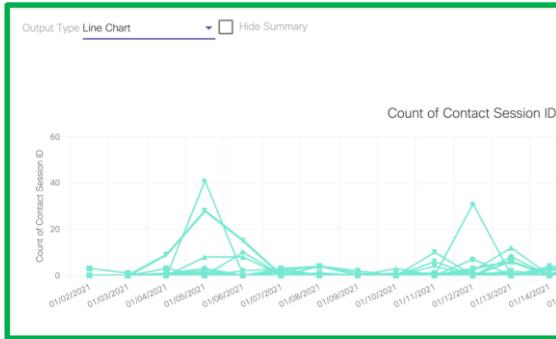
Callouts numbered 1 through 4 point to specific UI elements: 1 points to the top bar, 2 points to the Agent Name dropdown, 3 points to the Compute button, and 4 points to the table data row.

Team Name	Agent Name	Average of Connected Count
Team Name 1 (2)	Agent Name 1 (1) Agent Name 2 (1)	1038.00 7669.00
Team Name 2 (2)	Agent Name 1 (1) Agent Name 2 (1)	9777.00 2728.00

Visualization Scope & View

1. Select the repository
2. Select time period (Realtime/Historical)
3. Select Interval (Historical) or Duration & Refresh (Realtime)
4. Select the Data Output format

Data Output Formats



Supported Data Output Types:

1. Table
2. Heat Map
3. Row Heat Map
4. Column Heat Map
5. Line Chart
6. Bar Chart
7. Area Chart
8. Pie Chart
9. Sparkline Chart

Anatomy of a Visualization

The screenshot shows a visualization configuration interface with the following elements:

- Repository Selection (Step 1):** The top left corner shows "Type Agent Session Record" and "Visualization - 01/31/2024 21:01:00".
- Time Period (Step 2):** The "Start Time" dropdown is set to "Last 7 days" and "If run today".
- Data Output Format (Step 3):** The "Output Type" is set to "Table".
- Data Set Definition (Step 4):** The "Team Name" segment is selected. The table below shows the data structure:

Team Name	Agent Name	Average of Connected Count
Team Name 1 (2)	Agent Name 1 (1)	1038.00
	Agent Name 2 (1)	7669.00
Team Name 2 (2)	Agent Name 1 (1)	9777.00
	Agent Name 2 (1)	2728.00

- Profile Variables (Step 5):** The "Profile Variables" section is highlighted, showing "Average of Co..." and "Average of To...".

Visualization Scope & View

1. Select the repository
2. Select time period (Realtime/Historical)
3. Select Interval (Historical) or Duration & Refresh (Realtime)
4. Select the Data Output format

Define the Data Set

5. Choose the Profile Variables

Profile Variables: Visualization Building Blocks

Field

- Textual value of a variable, e.g. Agent Name, Contact Session ID, Queue Name
- It can be:
 - Count: # of times the variable appears
 - Value: the actual value of the variable
 - Cardinality: Total Number of unique records of the variable

Team Name	Agent Name	Cardinality Agent Name	Count Agent Name
akgosaing_team (2)	akgosaing agent1 (1)	1.0	2
	akgosaing sa,admin (1)	1.0	4
Summary		2	6

Value of Agent Name Count of Agent Name

Cardinality of Agent Name

Measure

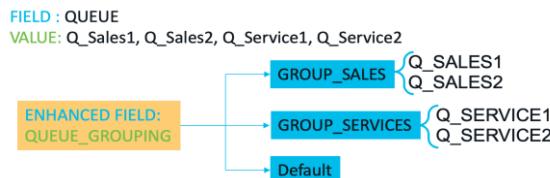
- Measures are computed values/aggregations (integer or decimal), e.g. Connected Duration, Hold Count
- It can be either the value or an arithmetic operation:
 - Sum
 - Average
 - Count
 - Value
 - Minimum
 - Maximum

Sum of Total Connected Duration
Average of Total Connected Duration
Count of Total Connected Duration
Minimum Total Connected Duration
Maximum Total Connected Duration

Profile Variables: Visualization Building Blocks

Enhanced Field

- Combine multiple values of a field in one or more groups.
- Enhanced Fields created in one report can be saved and re-used in other visualizations.



Formula

- Formulas can be created either from Fields or Measures.
- Arithmetic operations supported are +, -, × or ÷.
- Example: Average Speed of Answer = Average of Queue Duration + Average of Ringing Duration

A screenshot of a formula builder interface. The "Name" field is set to "Average Speed of Answer". The "Formula" field is set to "Arithmetic Expression". The expression "Average of Total Ringing Duration + Average of Queue Duration" is entered. A "Type a numeric value or select a column" input field is also visible.

Anatomy of a Visualization

The screenshot shows the configuration interface for a visualization. Key components include:

- Top Bar:** Type Agent Session Record, Visualization - 01/31/2024 21:01:00, Save button.
- Left Sidebar (Modules):** Module dropdown (Module1), Start Time (Last 7 days, If run today: 2024-01-24, End Date: 2024-01-30), Including (All Days), Compute button (3), Show Filter on Run Mode, Add Filter button.
- Middle Section:** Row Segments, Profile Variables (5), Column Segment (6, Profile Variables), Agent Name (6).
- Output Type:** Table (4), Show summary, Table level, Customize.
- Table Output:** Team Name, Agent Name, Average of Connected Count.

Team Name	Agent Name	Average of Connected Count
Team Name 1 (2)	Agent Name 1 (1)	1038.00
	Agent Name 2 (1)	7669.00
Team Name 2 (2)	Agent Name 1 (1)	9777.00
	Agent Name 2 (1)	2728.00

Visualization Scope & View

1. Select the repository
2. Select time period (Realtime/Historical)
3. Select Interval (Historical) or Duration & Refresh (Realtime)
4. Select the Data Output format

Define the Data Set

5. Choose the Profile Variables
6. Optionally, add segmentation

Row / Column Segments

- Segments allow to logically group the data under specific **Field only** variables.
 - Order of segments during edit mode is important.
 - Row Segments** are used more commonly compared to Column Segments.
- If we add an Interval to our visualization, this by default also becomes a row or column segment.

The screenshot shows a Cisco dashboard interface. On the left, there is a sidebar titled "Row Segments" containing the following items:

- Agent Name
- Interval (highlighted in orange)
- Site Name
- Team Name
- Agent Endpoint
- Channel Type

A green arrow points from the "Interval" item in the sidebar to a red box highlighting the "Interval" column in the main data table. The main table has the following columns:

Agent Name	Interval	Site Name	Team Name	Agent Endpoint (DN)	Channel Type	Login Time	Sign Out Time	Sign Out
Agent1_Lab (16)	01/05/2024 (4)	001_Site (4)	001_TeamA (4)	1000 (4)	chat (1) email (1) social (1) telephony (1)	01/05/2024 9:39:31 AM 01/05/2024 9:39:31 AM 01/05/2024 9:39:31 AM 01/05/2024 9:39:31 AM	01/05/2024 10:15:53 AM 01/05/2024 10:15:53 AM 01/05/2024 10:15:53 AM 01/05/2024 10:15:53 AM	3 3 3 1
	01/08/2024 (4)	001_Site (4)	001_TeamA (4)	1000 (4)	chat (1) email (1) social (1) telephony (1)	01/08/2024 4:52:43 AM 01/08/2024 4:52:43 AM 01/08/2024 4:52:43 AM 01/08/2024 4:52:43 AM	01/08/2024 10:04:42 AM 01/08/2024 10:04:42 AM 01/08/2024 10:04:42 AM 01/08/2024 10:04:42 AM	18 18 18 6
	01/09/2024 (4)	001_Site (4)	001_TeamA (4)	1000 (4)	chat (1) email (1) social (1) telephony (1)	01/09/2024 9:16:44 AM 01/09/2024 9:16:44 AM 01/09/2024 9:16:44 AM 01/09/2024 9:16:44 AM	01/09/2024 10:32:51 AM 01/09/2024 10:32:51 AM 01/09/2024 10:32:51 AM 01/09/2024 10:32:51 AM	3 3 3 3
	01/10/2024 (4)	001_Site (4)	001_TeamA (4)	1000 (4)	chat (1) email (1) social (1) telephony (1)	01/10/2024 4:25:10 AM 01/10/2024 4:25:10 AM 01/10/2024 4:25:10 AM 01/10/2024 4:25:10 AM	01/10/2024 5:28:31 AM 01/10/2024 5:28:31 AM 01/10/2024 5:28:31 AM 01/10/2024 5:28:31 AM	3 3 3 1
Agent100_Lab (40)	01/10/2024 (8)	CL2024_Site (8)	CL2024_CCA1_Team (4)	webrtc-601bf588-87ec-4c6b-98f9-5815...	chat (1) email (1) social (1) telephony (1)	01/10/2024 2:18:03 PM 01/10/2024 2:18:03 PM 01/10/2024 2:18:03 PM 01/10/2024 2:18:03 PM	01/10/2024 4:34:02 PM 01/10/2024 4:34:02 PM 01/10/2024 4:34:02 PM 01/10/2024 4:34:02 PM	5 5 5 1
			CL2024_ChartTeam (4)	webrtc-601bf588-87ec-4c6b-98f9-5815...	chat (1)	01/10/2024 2:03:19 PM	01/10/2024 2:14:38 PM	5

Count or Value?

- The most common source of confusion for report users -> "Do I use a Count or Value based variable?"
- By **Count**, we mean either Count or Cardinality for **Field** variables or the arithmetic operations (Count, Sum, Avg, Min, Max) for **Measure** variables.

Count of Contact Session ID
Value of Contact Session ID
Cardinality of Contact Session ID

Sum of Connected Duration
Average of Connected Duration
Count of Connected Duration
Minimum Connected Duration
Maximum Connected Duration
Value of Connected Duration

Ask Yourself:

- Do I want to see aggregations of multiple entities -> **Count based** (e.g. # of Calls handled by a team, # of different Agents that handled a call for a specific Queue)
- Do I want to see the raw data (each in a separate row) for each unique call or agent separately -> **Value based** (e.g. all session IDs of the calls handled, the name of each agent that handled a call for a specific queue)

Count or Value?

Count of Final Queue Name	Count of Contact Session ID
26	62

Value of Queue Name	Value of Contact Session ID
CL2024_Queue1	29eb9d9f-04d0-42b0-8462-edb62ee22dc2
CL2024_SBR_QV_Team51	8fc21eab-2d61-49c7-b251-002547e5dc2
CL2024_SBR_QV_Team51	53af22b6-a6b6-431a-bda7-bd5d50ec9f0a
CL2024_SBR_QV_Team51	79ea065d-e412-4908-b106-e585c0a46b36
CL2024_SBR_QV_Team51	947cc7b3-05ba-484a-8fc2-08c489324fb6
CL2024_SBR_QV_Team51	b3b89fc1-4ac9-461e-9230-7edfb804d366
CL2024_SBR_QV_Team51	d8f6ec43-8768-4936-8554-1c79971a8603
CL2024_SBR_QV_Team51	d3614edf-4d3a-4ede-a8af-722521e25774
CL2024_Data_ChatQueue	b0db5b47-eacf-444d-aa7f-b25afb45bae
CL2024_SBR_QV_Team52	c2cbc415-1227-4265-9c90-815b8e98bf16
	fafac343-ed0b-4c82-aa94-186f03399999

- Count based reports will provide the total tenant's values unless segmented.
 - Segments allow us to combine value fields (e.g. Queue Name) with aggregation statistics.
 - We cannot use Segments in value-based reports.

Note: A report is considered count or value based on what we selected on the 1st profile variable we added. After that, all profile variables we try to add will only show Count or Value options respectively.

Final Queue Name	Count of Contact Session ID
CL2024_CCAL_Queue (1)	5
CL2024_ChatQ (1)	3
CL2024_Data_ChatQueue (1)	1
CL2024_SBR_QV_Team51 (1)	13
CL_DEMO (1)	4
N/A (1)	36

Anatomy of a Visualization

The screenshot shows a visualization configuration interface with the following components and numbered callouts:

- Top Bar:** Type Agent Session Record, Visualization - 01/31/2024 21:01:00, Save button.
- Left Sidebar (Module):** Modules tab selected, Module1 dropdown, Start Time (Last 7 days), If run today, Start Date: 2024-01-24, End Date: 2024-01-30, Including (All Days), Compute button (callout 3), Show Filter on Run Mode (callout 7), Add Filter button (callout 7).
- Central Area:** Click to add title, Row Segments, Profile Variables (callout 5), Column Segment (callout 6), Team Name (callout 2), Agent Name (callout 6), Output Type Table (callout 4), Show summary, Table level, Customize, Data table.
- Data Table:** Team Name, Agent Name, Average of Connected Count.

Team Name	Agent Name	Average of Connected Count
Team Name 1 (2)	Agent Name 1 (1)	1038.00
	Agent Name 2 (1)	7669.00
Team Name 2 (2)	Agent Name 1 (1)	9777.00
	Agent Name 2 (1)	2728.00

Visualization Scope & View

1. Select the repository
2. Select time period (Realtime/Historical)
3. Select Interval (Historical) or Duration & Refresh (Realtime)

4. Select the Data Output format

Define the Data Set

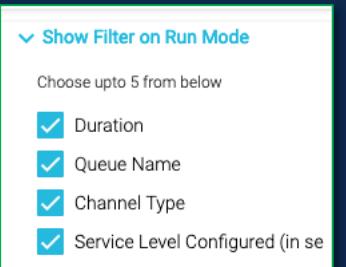
5. Choose the Profile Variables
6. Optionally, add segmentation
7. Optionally, create filters

Filters

- There are 3 different places we can setup a filter inside a visualization.

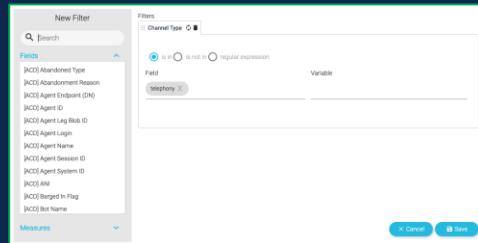
Run-Mode Filtering

- ❖ Set in “Edit Mode”, can be Duration or any Row Segment (up to 5).
- ❖ It can be updated dynamically during report execution.



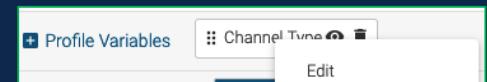
Report Filter

- ❖ When set, it applies for the whole report.
- ❖ It can be inclusive (**is in**), exclusive (**is not in**), an operation (**>**, **=**, **!=**, **between**) or a RegEx.



Column Filter

- ❖ It is configured the same way as a Report Filter, but it only applies to a specific column.
- ❖ It is configured by right-clicking on a variable and selecting “Edit”.



The Power of RegEx

- Regular Expression (RegEx) is a sequence of characters that specifies a match pattern in text.
 - For example, asterisk “.” is used to match all possible characters.
 - Thus, if I used RegEx string *StudentID..* to match a text, I would match all strings like **StudentID51**, **StudentID52**, **StudentID53** etc (but not **StudentID100** for example, as it has more than 2 characters).
- As filters in Webex CC Analyzer allow to use RegEx, we can create powerful generic filters. Some common use cases include:
 1. `.*` -> This filter specifies that I want to match any character (“.”) infinite times (“*”). Basically, as this filter matches all strings, it is a great way to filter out N/A (empty) results in a report.
 - For example, if we use “.*” as filter in **Queue Name** variable, it will return all contacts that had a queue assigned (so not the calls that ended before reaching queue).
 2. `((31)|(32)).*` -> This filter specifies that I want to match any string which starts with 31 or (“|”) 32 and then can have an infinite number of characters. RegEx allows us to use this OR operator to match multiple scenarios in a single pattern.
 - With this example, we can filter out all ANIs that come from the Netherlands (+31 Country Code) or Belgium (+32 Country Code).
 3. `.*Sales.*` -> Similarly with example 2, we can also use any string between `*` to find all the values that include the word “Sales” in them, for example if we want to fetch all the Sales teams.

Anatomy of a Visualization

The screenshot shows a visualization configuration interface with the following components and numbered callouts:

- Top Bar:** Type Agent Session Record, Visualization - 01/31/2024 21:01:00, Save button.
- Left Sidebar (Module):** Modules tab selected, Module1 dropdown, Start Time (Last 7 days), If run today, Start Date: 2024-01-24, End Date: 2024-01-30, Including (All Days), Compute button (3), Show Filter on Run Mode (7), Add Filter button (7).
- Central Area:** Click to add title, Row Segments, Profile Variables (7), Column Segment (6), Average of Connected Count (5), Team Name (6), Agent Name (6), Output Type Table (4), Show summary, Table level (8), Customize.
- Table Preview:** Shows data for Team Name 1 (2) and Team Name 2 (2).

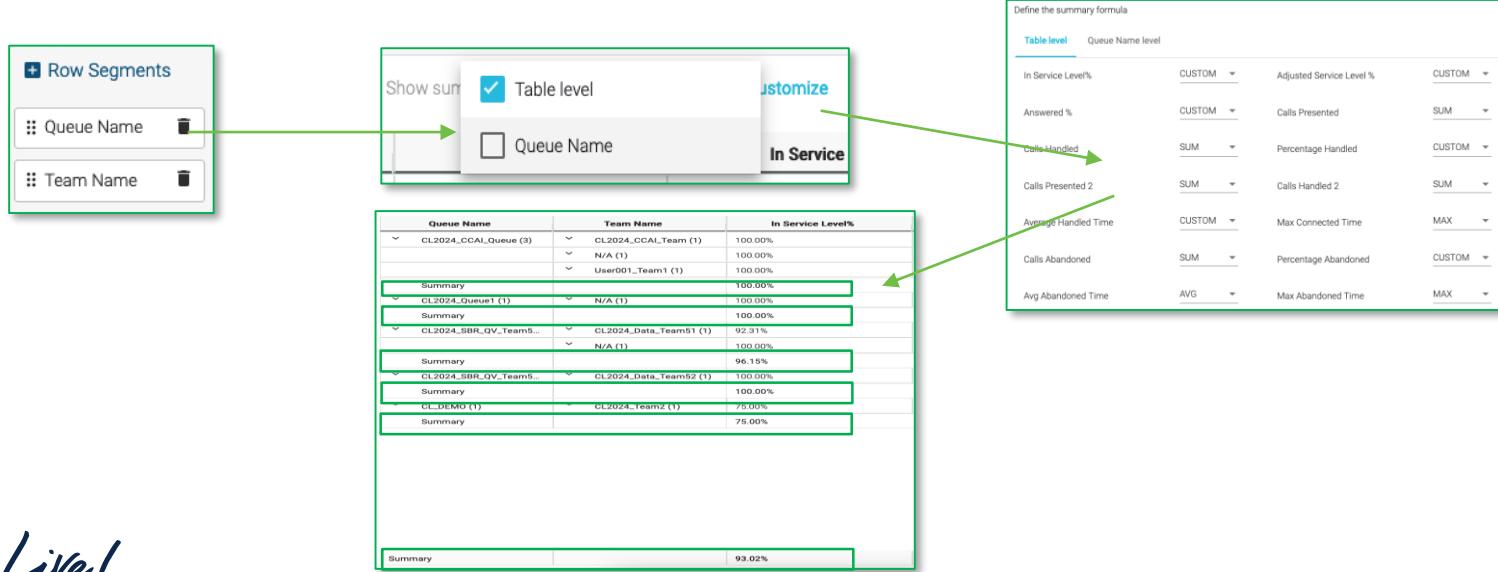
Team Name	Agent Name	Average of Connected Count
Team Name 1 (2)	Agent Name 1 (1)	1038.00
	Agent Name 2 (1)	7669.00
Team Name 2 (2)	Agent Name 1 (1)	9777.00
	Agent Name 2 (1)	2728.00

Visualization Scope & View

1. Select the repository
2. Select time period (Realtime/Historical)
3. Select Interval (Historical) or Duration & Refresh (Realtime)
4. Select the Data Output format
5. Choose the Profile Variables
6. Optionally, add segmentation
7. Optionally, create filters
8. Optionally, customize report summary

Customize Report Summary

- Report Summary allows us to define the formula used to summarize the results of a specific column in a report. Report Summary is available on:
 - Table Level (for all Reports)
 - Top-Level Row Segment (for reports with at least 2 Row Segments)



Anatomy of a Visualization

The screenshot shows a visualization configuration interface with the following numbered components:

- 1: Type Agent Session Record
- 2: Start Time (Last 7 days)
- 3: Compute button
- 4: Output Type Table
- 5: Average of Connected Count metric
- 6: Team Name and Agent Name segmentation
- 7: Save button
- 8: Show summary and Customize options
- 9: Module selection (Module1)

The main area displays a table with the following data:

Team Name	Agent Name	Average of Connected Count
Team Name 1 (2)	Agent Name 1 (1)	1038.00
	Agent Name 2 (1)	7669.00
Team Name 2 (2)	Agent Name 1 (1)	9777.00
	Agent Name 2 (1)	2728.00

Visualization Scope & View

1. Select the repository
2. Select time period (Realtime/Historical)
3. Select Interval (Historical) or Duration & Refresh (Realtime)
4. Select the Data Output format
5. Choose the Profile Variables
6. Optionally, add segmentation
7. Optionally, create filters
8. Optionally, customize report summary
9. Lastly, you can create multiple modules

Compound Visualizations (Modules)

- A compound visualization (Module) allows users to have different views for the same visualization.
- Each module can have differing date ranges, intervals, and filters.
- Compound visualizations cannot be scheduled or exported.

Queue Name	CSQ All Fields					Last Month			
	Last Week					In Service Level%	Calls Presented	Calls Handled	Percentage Handled
	In Service Level%	Calls Presented	Calls Handled	Percentage Handled					
CL2024_CCAL_Queue (1)	100.00%	9	4	44.44%	100.00%	23	11	47.83%	
CL2024_Queue1 (1)	100.00%	1	0	0.00%	100.00%	5	0	0.00%	
CL2024_SBR_QV_Team5...	92.86%	28	6	21.43%	92.86%	28	6	21.43%	
CL2024_SBR_QV_Team5...	100.00%	1	0	0.00%	100.00%	1	0	0.00%	
CL_DEMO (1)	75.00%	4	1	25.00%	94.44%	36	8	22.22%	
CL_Outdial_Queue (1)					50.00%	2	0	0.00%	
User33_VoiceQueue (1)					80.00%	10	0	0.00%	

Module 1 Module 2

Analyzer Variables Tab

- Not the variables we have talked about so far!
- Variables tab allows to group & create a predefined set of values, that can then be used in report filters.
- We can define the scope of these variables:
 - USER:** Only to be used by the creator
 - GLOBAL:** Available across the organization



New Variable

*Name: CL2024_Variable_Teams *Associated Column: INTERACTION-Team Name

Values: CL2024_Data_Team51 X, CL2024_Data_Team52 X, CL2024_Data_Team53 X

Description:

Scope: User Global

Cancel Save

New Filter

team

Fields: [ACD] Callback Team Name, [ACD] Team ID, [ACD] Team System ID

Measures

Filters: Team Name

Is in Is not in Regular expression

Field: Variable

Please select/enter a value

\$CL2024_Variable_Teams

\$ME

Lab 1: Analyzer



20 min

Exercise 1.3: Create a Visualization from scratch

What did we learn?

- *Create a complete report from zero while taking advantage of all the advanced features Analyzer offers*

Dashboards

- Allows to add multiple visualizations under the same view.
- Users can combine reports of different type (historical/realtme, count/value based, different repositories).
- Run-mode filtering is possible and will be applied to all visualizations at the same time.

The screenshot shows a dashboard application interface. At the top, there are tabs for "Modules" and "Formatting", and buttons for "Save", "Preview", and "Delete". A sidebar on the left lists various modules: "DataLab Admin Only", "Stock Reports", "Queue Activity by Queue", "Queue All Fields Report", "Queue Call Distribution Sum...", "Abandoned Call Detail Activit...", "Agent Call Summary Report", "Agent Detail Report", and "Agent Summary Report". Below the sidebar is a title input field with placeholder text "Click to add title". A modal window titled "Agent Summary Report" displays a table with columns "Agent Name" and "Calls". The table contains three rows: "Agent Name 1 (1)" with 9609 calls, "Agent Name 2 (1)" with 7179 calls, and a summary row with 7126 calls. A green arrow points from the "Agent Summary Report" title to the modal. Another green arrow points from the "Agent Summary Report" title to the "Move & Resize" handle in the bottom right corner of the modal. In the main content area, there is a chart titled "Agent - Chart" showing "Contacts Handled" over time. Below the chart is a table titled "Agent" with columns for "Agent Name", "Interval", "Multi Media Profile Type", "Channel Type", "Skill Profile", "Skills", "Log In Count", and "Contact Handled". The table has two main sections: "Agent1 Lab (4)" and "Summary". The "Agent1 Lab (4)" section shows data for four intervals in February 2024, while the "Summary" section shows data for March 2024. A green arrow points from the "Agent - Chart" title to the "Move & Resize" handle in the bottom right corner of the chart area.

Save Preview Delete

Click to add title

Agent Summary Report

Agent Name	Calls
Agent Name 1 (1)	9609
Agent Name 2 (1)	7179
Summary	7126

Drag & Drop

Move & Resize

Agent Historical Dashboard

Agent - Chart

Contacts Handled

Agent Name	Interval	Multi Media Profile Type	Channel Type	Skill Profile	Skills	Log In Count	Contact Handled
Agent1 Lab (4)	02/02/2024 (4)	BLENDED (4)	chat (1)	N/A (1)	N/A (1)	3	1
			email (1)	N/A (1)	N/A (1)	3	0
			social (1)	N/A (1)	N/A (1)	0	0
			telephony (1)	N/A (1)	N/A (1)	1	0
Summary	01/30/2024 (4)	BLENDED (4)	chat (1)	N/A (1)	N/A (1)	15	2
			email (1)	N/A (1)	N/A (1)	15	0
			social (1)	N/A (1)	N/A (1)	0	0
			telephony (1)	N/A (1)	N/A (1)	3	0
	01/31/2024 (4)	BLENDED (4)	chat (1)	N/A (1)	N/A (1)	5	1

CISCO Live!

Stock Dashboards “Cheat Sheet”

- **Service Level & Queue Statistics** → Contact Centre Overview – Historical (-1210), Contact Centre Overview – Realtime (-1215)
- **Entry Point Statistics** → Entry Point Historical Dashboard (-1166)
- **Abandoned Statistics** → Abandoned Contacts (-1181), Interval Abandoned Dashboard (-1172)
- **Agent/Team Status Tracking & Call Handling** → Agent Historical Dashboard (-1160), Team State Dashboard (-1237)

Analyzer Beta

- The future of Analyzer: A new look with enhanced capabilities and a more powerful backend architecture.
- New concepts are introduced:
 - All reports are treated as **widgets**
 - Label** structure instead of folders
- Today, only stock Historical reports are available in Analyzer Beta, with more options coming soon.

The screenshot shows the Cisco Webex Analyzer Beta interface. At the top, there is a green banner with the text "Experience the Analyzer Beta" and a "Try Now" button. Below this, a modal window titled "Experience the Analyzer Beta" displays a cartoon character of a person with a speech bubble, followed by the text: "Explore the all new Analyzer Beta with new features and enhanced user-interfaces. Any edits or changes made in the Analyzer Beta will not affect the existing version." There is a checkbox for "Do not show this message again" and two buttons: "Cancel" and "Launch". A green arrow points from the "Launch" button to the main report list below. The main interface shows a dark-themed dashboard with a search bar and a navigation bar with tabs: "All (75)", "Recent (10)", "Favorite (1)", and "Stock (75)". The "Stock" tab is selected. The report list table has columns: "Name", "Labels", "Last Edited By", "Last Edited", "Created By", and "Actions". The reports listed are: Abandoned Call Detail Activ..., Agent Call Summary Report, Agent Detail Report, Agent Summary Report, Application Summary Report, CSQ Activity Report by Win..., CSQ Agent Summary Report, CSQ All Fields Report, Multichannel Agent Summary, and Queue Abandoned Historica... . Each report has a star icon, a timestamp (09 May 2023), and a "More" actions button.

Name	Labels	Last Edited By	Last Edited	Created By	Actions
Abandoned Call Detail Activ...	Stock Transition	Cisco	09 May 2023	Cisco	⋮
Agent Call Summary Report	Stock Transition	Cisco	09 May 2023	Cisco	⋮
Agent Detail Report	Stock Transition	Cisco	09 May 2023	Cisco	⋮
Agent Summary Report	Stock Transition	Cisco	09 May 2023	Cisco	⋮
Application Summary Report	Stock Transition	Cisco	09 May 2023	Cisco	⋮
CSQ Activity Report by Win...	Stock Transition	Cisco	09 May 2023	Cisco	⋮
CSQ Agent Summary Report	Stock Transition	Cisco	09 May 2023	Cisco	⋮
CSQ All Fields Report	Stock Transition	Cisco	09 May 2023	Cisco	⋮
Multichannel Agent Summary	Stock Transition	Cisco	09 May 2023	Cisco	⋮
Queue Abandoned Historica...	Historical Multichannel Stock	Cisco	09 May 2023	Cisco	⋮

Lab 1: Analyzer



10 min

Exercise 1.4: Analyzer Beta

Exercise 1.5: Dashboards (Bonus)

What did we learn?

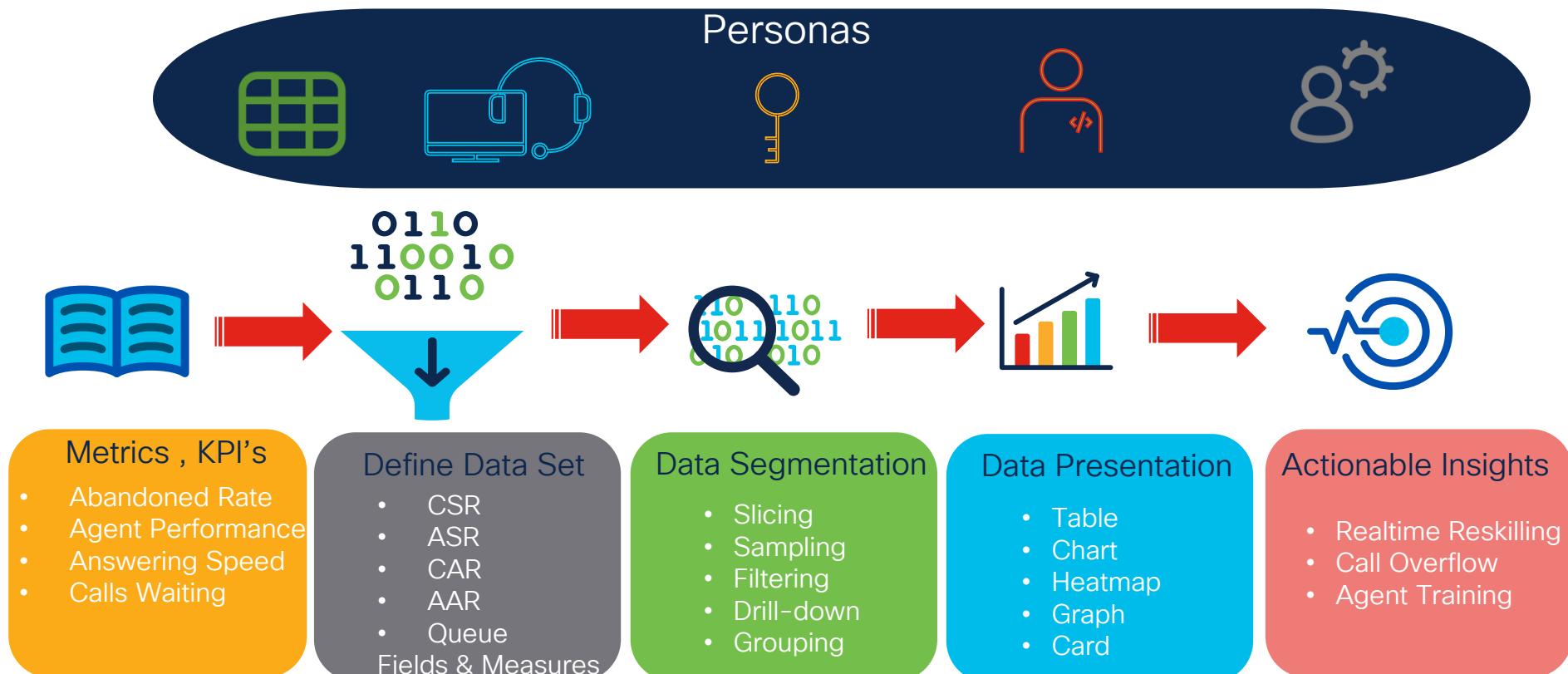
- *Future Analyzer looks much cooler!*
- *New Analyzer will bring (1) new look & feel on reports (2) additional capabilities and (3) better efficiency & speed due to new architecture.*
- *Dashboards are very easy to make and they allow users to combine reports of different types and repositories in a single view.*

Analyzer Beta – What's Next?

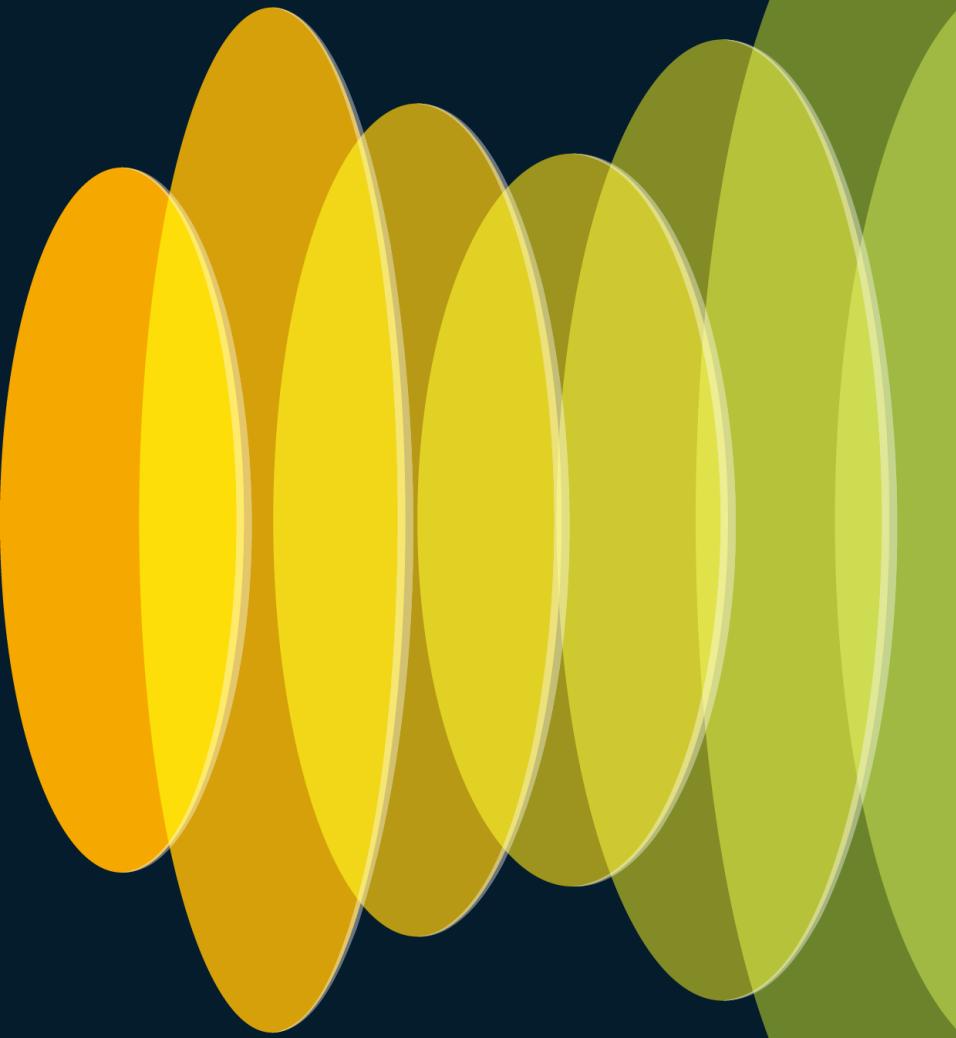
- Realtime Stock & Custom Dashboard Creation
- Quick & Easy Permissions assignment per dashboard
- Multi-Level Formulas
- Custom Card format Dashboard creation
- Colour Coding Thresholds

The Analytics Approach

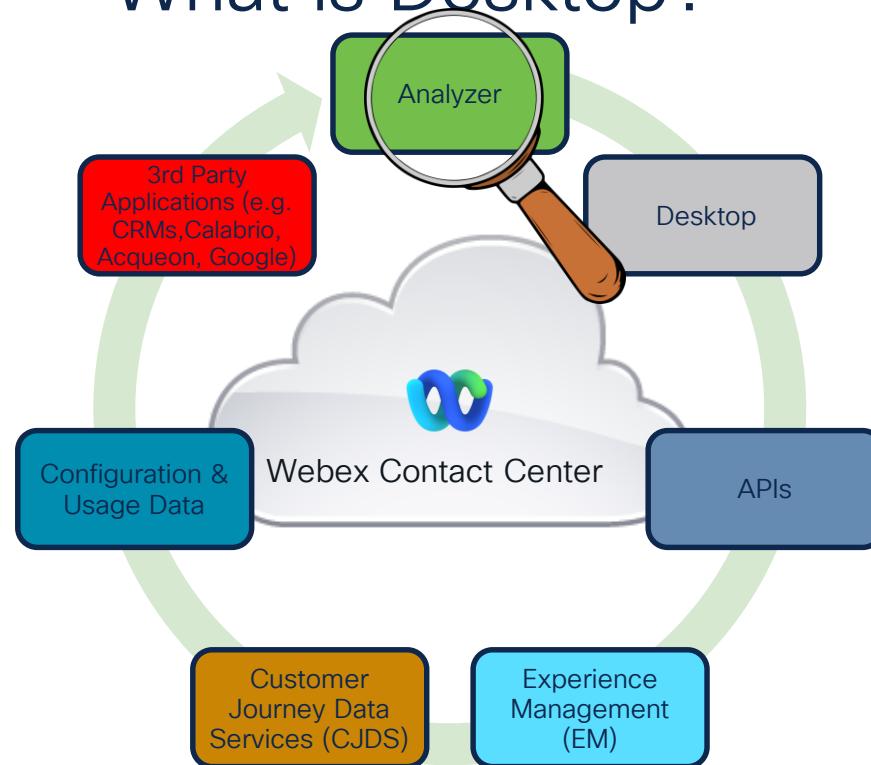
REFERENCE



Desktop



What is Desktop?



- **(Agent) Desktop** is the interface in which agents handle contacts & **(Supervisor) Desktop** includes all agent capabilities alongside extended Team monitoring options for supervisors.
- We will focus on the **out-of-the-box** data points & dashboards, but advanced custom widgets can connect to almost any external database and provide all kinds of visibility.

[Cisco Desktop Widget Samples Repository](#)

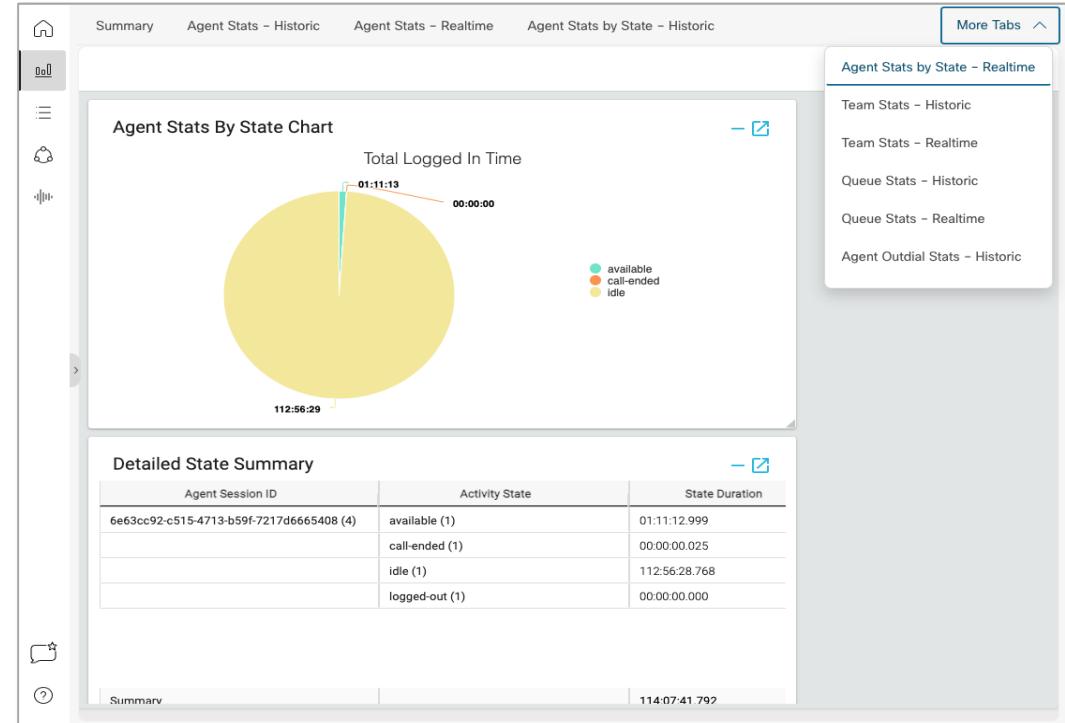
Agent Popover & Interaction Panes

- Administrators have the capability to present to Agents additional variables during both the ringing stage of an incoming call as well as after the call has been answered.
- This is part of the standard Agent Desktop, configured in the **Flow Designer**.
- We can present any data to the customer that has been created or fetched during the flow.



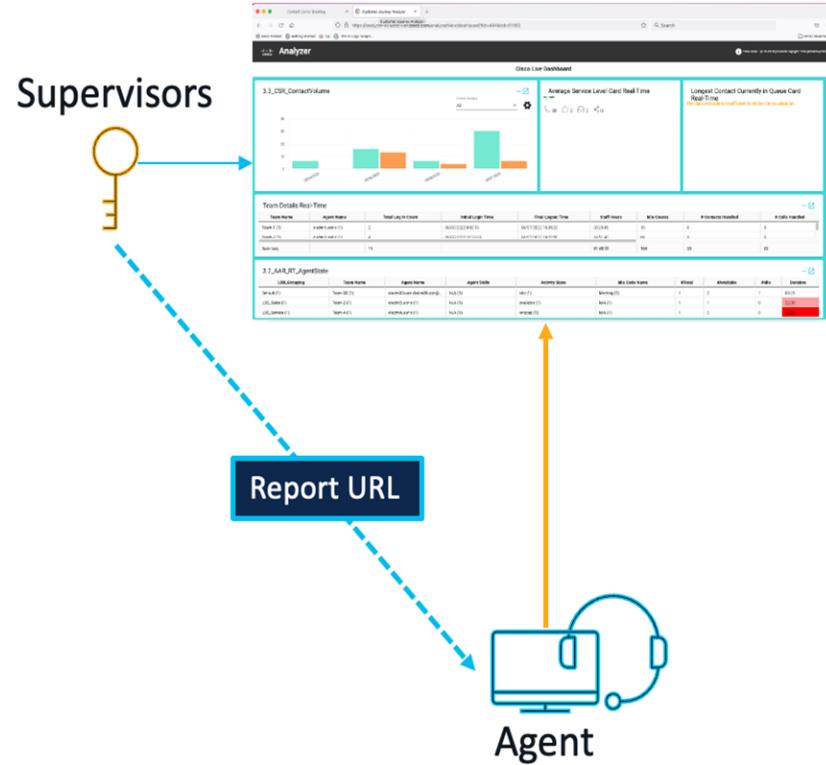
Agent Performance Statistics (APS)

- APS offers multiple Dashboards with Agent, Queue or Team statistics (Historical or Realtime), available both in Agent & Supervisor Desktop.
- Access to APS is defined in the **Desktop Profile** of the user.
- APS reports are not editable, we can only apply **persistent filters**.
- APS dashboards do not exist in Analyzer as stock reports, but we can cross-launch them from Desktop in a new window.



Browser Link Report Access

- Quick option for Supervisors to share a specific visualization with Agents
- Browser link access possible for both standard and premium agents
- Runs in browser timezone
- Note: Drilldown functionality is not available via browser link report access



<https://analyzer-v2.wxcc-us1.cisco.com/analyzer/view/visualization?tid=105&rid=60420>

Embedding the Report into the Agent Desktop

[Desktop Developer Documentation](#)

The screenshot shows the "iFrame Widgets" section of the documentation. It includes a bulleted list of widget types and a note about custom widgets. Below this is a detailed explanation of the iFrame widget, mentioning Content Security Policy (CSP) requirements and an example code snippet:

```
Content-Security-Policy: frame-ancestors 'self' https://*.cisco.com;
```

Below the code, there is a note about Google Chrome throttling impact and a warning about using Web Workers for timers.

Get Started



[Analyzer Iframe Widget Sample \(GitHub\)](#)

The screenshot shows the GitHub repository structure for the "Analyzer_Iframe_Widget_Layout.json" file. It also includes a preview of the "Analyzer Reporting Widget" and its pre-requisites:

Pre-requisites:

- Before you begin, you must create an Analyzer Report on WebeCC Analyzer.
- The URL is in the format: https://analyzerwcc-___.cisco.com where ___ is the datacenter (us1,eu1,eu2,etc.)
- Once the report is created, click on run. It will be in the format: <https://analyzer-v2.wcc-us1.cisco.com/analyzer/view/visualizations/11111111111111111111111111111111>, where 11111111111111111111111111111111 is the org ID and id is the report ID.
- Copy the URL of your custom report, and into the Desktop Layout. In the attached JSON, the desktop layout has an iFrame Navigation tab widget seen in lines 102-128.
- Change the source ("src") tag for the iFrame element inside the Desktop Layout JSON and point it to the Analyzer Report.



The screenshot shows the Cisco Live Agent Desktop interface. On the left, there's a navigation bar with icons for Home, Dial, and Agent. The main area is titled "Cisco live! Agent Desktop". On the right, there's a "Analyzer" module with a table showing agent status:

Agent Name	State	Idle Code	Duration
Tenant Admin PS	idle	Meeting	00:04:28

Lab 2: Desktop



10 min

Exercise 2.1: Agent Personal Statistics (APS)

Exercise 2.2: Agent Direct URL Access

Exercise 2.3: Embedding the Report into the Agent Desktop (Bonus)

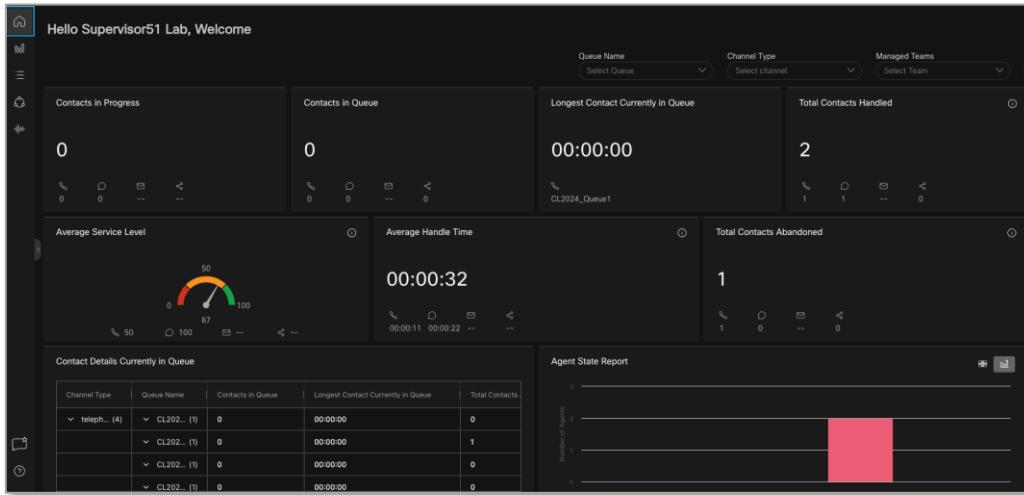
What did we learn?

- All Agents are offered –by default- a tab full of uneditable reports to provide them insights for their daily tasks.
- If they need additional visibility, supervisors can share with them custom reports via URL.
- If the need for a report is recurring, it can be embedded to their desktop.

Supervisor Desktop

NEW

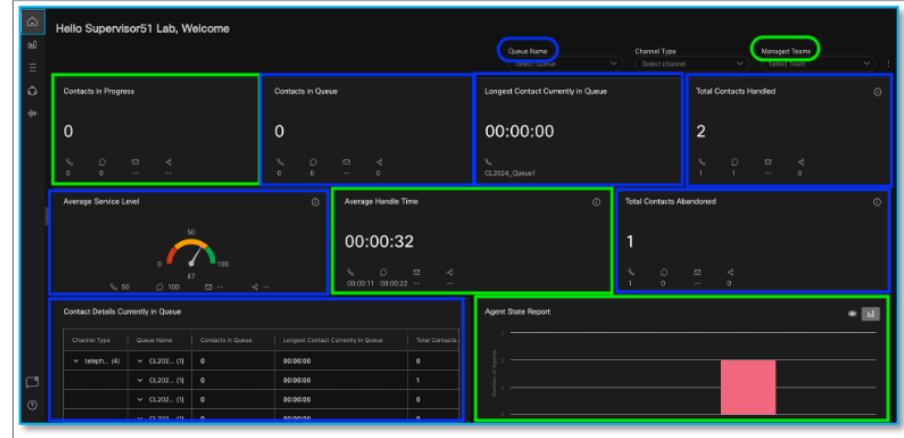
- Supervisor Desktop Home offers a comprehensive dashboard with Queue, Team & Agent Statistics.
- Supervisors can filter the dashboards, remove unwanted widgets but cannot edit existing ones today.
- Team Performance Details tab offers Supervisors a real-time view of all their managed agents, their current status and details.



Team Performance Details									
Displaying 1 Agent									
Agent Name	Agent State	Agent State Duration	Phone Number	Site	Team	Contact Status	Time in Contact State	Total C	Actions
SA Admin Cisco	Meeting	24:07:33	Desktop	CL2024_Site	CL2024_CCAL_Team	-	-	-	
Agent51 Lab	Meeting	10:03:28	Desktop	CL2024_Site	CL2024_Data_Team51	-	-	-	

Supervisor Desktop Tips

1. Desktop allows to filter on either Queue or Team. Out of the Desktop's 9 Widgets:
 - Contacts in Progress, Average Handle Time and Agent State Report metrics depend on Teams.
 - All the rest results are based on selected Queues filter.



2. Customers sometimes have confusion when trying to compare the KPI cards of Desktop with Analyzer. Thus, Cisco create a dedicated [article](#) to explain all the necessary details when comparing.

Lab 2: Desktop



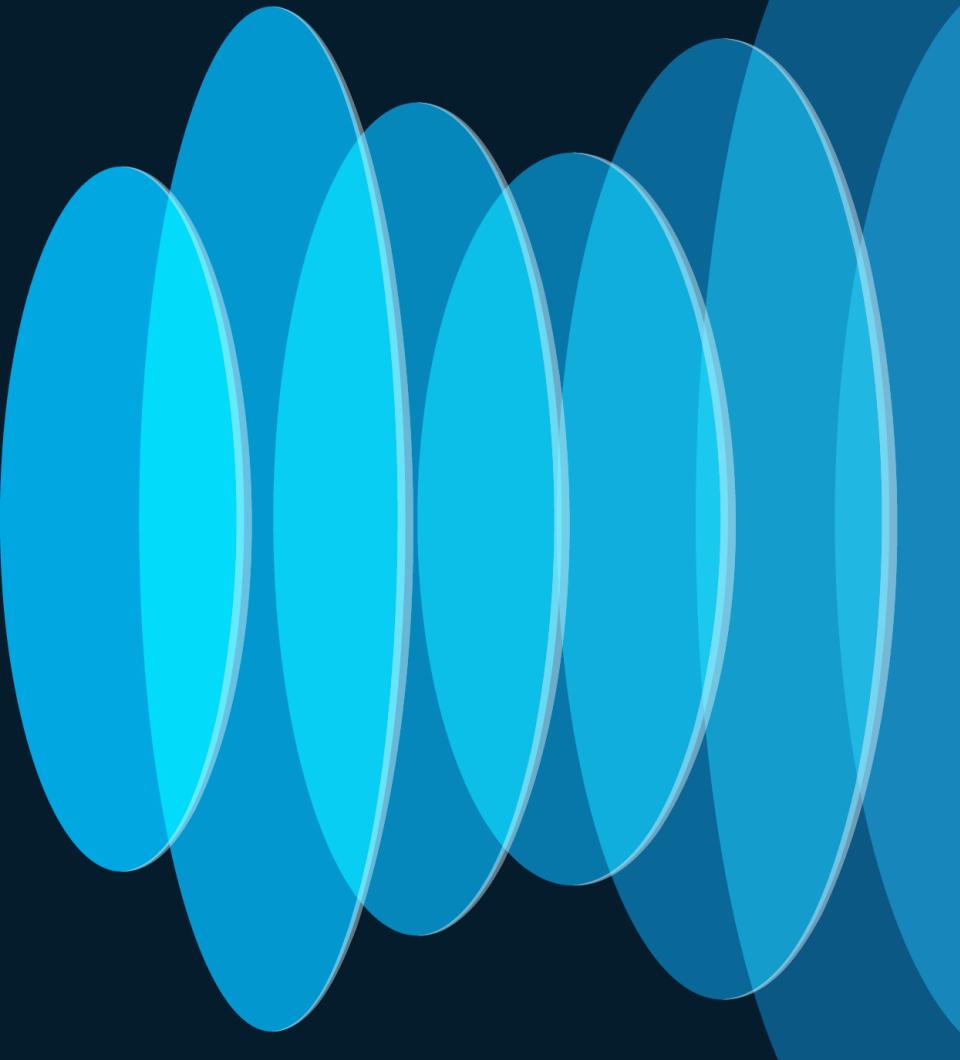
3 min

Exercise 2.4: Supervisor Desktop Dashboard (Demo)

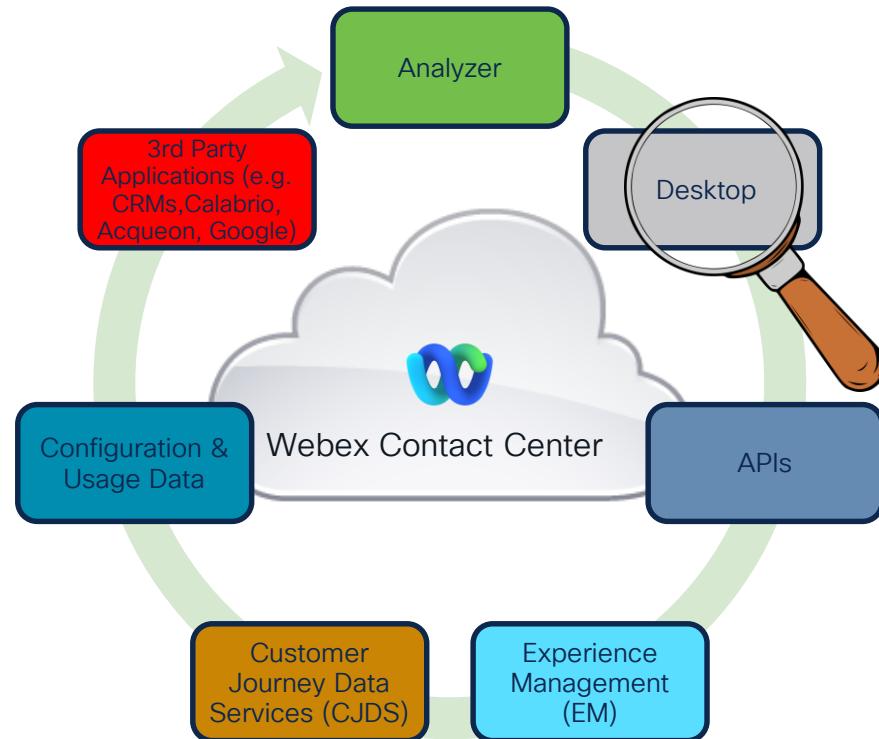
What did we learn?

- Supervisor Desktop offers various reporting widgets and team monitoring capabilities for Supervisors.
- Our goal is to make Supervisor Desktop the only portal the Supervisor needs to access to perform all their daily tasks.

APIs



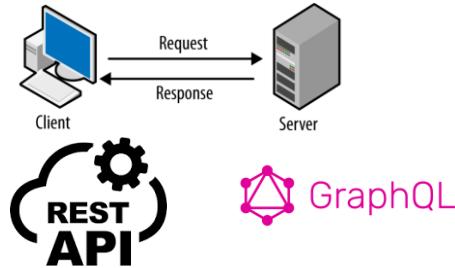
What are APIs?



- API is the mechanism that enables two software components to communicate with each other using a set of definitions and protocols.
- As API is a broad subject with many different technologies and capabilities (even amongst the different Webex CC APIs), in this lab we will focus only on the [Search API](#), i.e. Webex Contact Center Reporting API.

APIs for Beginners

- I ask you for something -> API Request
You give it to me -> API Response
- In what language do I ask? -> Concept/Protocol
- Where is it? -> Endpoint/URL
- How do I want it? -> Header
- What do I want? -> Parameters
- Optionally, do I need to give you more information to understand what I want? -> Body
- Is it okay for you to give it to me? -> Authorization



A screenshot of a developer tool interface. At the top, there are tabs for 'Sample Code' (highlighted in blue) and 'Try Out'. Below the tabs, a 'POST /search' button is shown next to a 'curl' dropdown menu. A code block displays a curl command:

```
1 curl --request POST \
2   --url 'https://api.lwccc-us1.cisco.com/search?orgId=97cdbf45-ebe2-468
3   --header 'Accept: application/json' \
4   --header 'Authorization: Bearer YOUR_TOKEN' \
5   --header 'Content-Type: application/json' \
6   --data '{"query": ""}'
```



Search API GraphQL GitHub Documentation



- Cisco created a comprehensive [Guide](#) & [GitHub repo](#) with definitions, examples & use cases.
- Data Dictionary provided also for every single available variable in the API.

Getting started with the Search API using GraphQL

For a quick overview of the `search` API and how to use our documentation, refer the video below:

Watch Getting Started With the Search API

Watch: Getting Started with the /search API

Introduction

A GraphQL API enables clients to construct queries to retrieve data. The queries are defined by the API server in the form of a GraphQL schema, which acts as a contract between the server and the client.

● Note: For an introduction to GraphQL, refer: [GraphQL - 101](#)

The API allows access to the following:

1. Task related data stored as Customer Session Records (CSRs) and Customer Activity Records (CARs), accessible using the `taskDetails` graphql query.
2. Agent related data stored as Agent Session Records (ASRs) and Agent Activity Records (AARs), accessible using the `agentSession` graphql query.
3. Queue related data stored as Call Leg Records (CLRs), accessible using the `taskLegDetails` graphql query

Based on the operation, a query can be of two types:

1. Query to get the list of records (CSR, ASR, CLR) based on the filter criteria.

2. Query to get the details of a specific record (CSR, ASR, CLR) based on the ID.

Contents

- Hitting data
- Sorting Support
- Pagination support for fetching raw data
- Pagination Support
 - Inner Pagination / Paginating CAR and AAR records
 - Pagination Support for Aggregation with Group By
- Global Variables Support
 - Filtering based on global variables
 - Performing Aggregations on global variables
 - Performing group by operation on global variables
- Restrictions
- Recommendations and best practices
- Sample Usecases and queries
- Data Dictionary
- Support

CSR Data Dictionary

Customer Session Record (CSR) - It represents the customer workflow, consisting of a sequence of customer activities. The following fields are available in a CSR record.

Note - These fields are applicable only for the API and not for Analyzer UI.

Field Name	Field Type	Description	Is Aggregation allowed?	Is GroupBy allowed?	Is Filter allowed? (use <code>filter</code> argument for all the fields)	Is Sortable?
id	String	A unique string that identifies the contact session.	Yes	Yes	Yes	Yes
status	String	Current state of the contact. E.g: available, idle, ringing, wrapup, etc.	Yes	Yes	Yes	Yes
channelType	String	The media type of the contact, such as telephony, email, or chat.	Yes	Yes	Yes	Yes
createdTime	Long	Time when the contact started.	Yes	Yes	No	Yes
endedTime	Long	Time when the contact ended.	Yes	Yes	Yes	Yes

Developer Portal

- *Every API is only as good as its documentation.*
- Developer Portal contains the information, examples and testing for all Webex Contact Center APIs, including Search API.
- Developer Portal provides information on all the required parameters to construct the API query, as well as the expected response format.
 - Also, it allows to directly test APIs from the portal, without the need to write code or use an external tool.

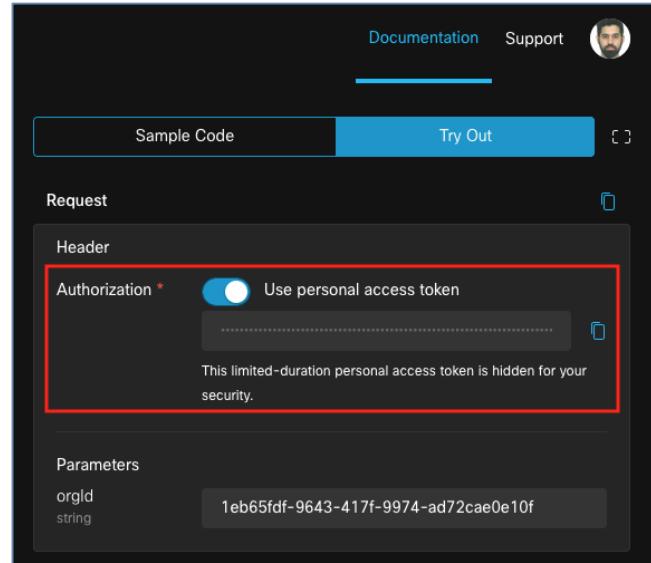
The screenshot shows the 'Search Tasks' page in the Webex Contact Center for Developers API documentation. The left sidebar lists various API categories like Overview, Getting Started, App Submission Process, Authentication, Common API Errors, Integrations, Introduction to APIs, Rate Limiting, and Sandbox. The main content area is titled 'Search Tasks' and describes the /search GraphQL endpoint. It explains that the search API enables customers to fetch data from WxCC. Parameters are passed as part of the POST request body. It details query types (task, taskDetails, taskLegDetails, agentSession), task and taskLegDetails supports a max page size of 500, whereas taskDetails and agentSession supports a max page size of 250. It also covers mandatory parameters for queries, optional filters, and aggregation input parameters. The right side of the interface includes a 'Sample Code' section with a curl command, a 'cURL' dropdown, and a 'Try Out' button. Below the code is a 'Response' section showing a JSON object representing a task record.

```
1 curl --request POST \
2   --url "https://api.wxcc-us1.cisco.com/search?orgId=97cdbf45-ebe2-468
3   --header 'Accept: application/json' \
4   --header 'Authorization: Bearer YOUR_TOKEN' \
5   --header 'Content-Type: application/json' \
6   --data '{"query":""}
```

```
1 {
2   "data": {
3     "task": {
4       "tasks": [
5         {
6           "id": "fb53f6d1-5535-4ac8-b081-53834e17d6f5",
7           "channelType": "telephony",
8           "createdTime": 1629450000000,
9           "endedTime": 1630380980406,
10          "captureRequested": true,
11          "isActive": false,
12          "status": "ended",
13          "queue": [
14            {
15              "id": "e434a654-df4c-42dc-908b-3d9d0206a616",
16              "name": "cb_outdial_queue"
17            }
18          ],
19          "owner": {
20            "name": "callbackorg2User1 callbackorg2User1",
21            "id": "74ab6507-a32a-479c-bda7-15ff0b6c6c3c"
22          }
23        }
24      ]
25    }
26  }
27 }
```

API Authentication

- All Webex CC APIs use OAuth2 for authentication.
 - OAuth2 requires user to use a token instead of a password to get access to API resources.
- An **Organization full Admin** role is needed for the user to be able to get access and use the APIs.
- Quick Solution -> Try Out from Developer Portal
 - An Access Token (lasting ~12 hours) is automatically generated and can be instantly used.



API Authentication – Continuous Access

- [Authentication](#) process for continuous access (e.g. application, bot) requires the following steps:
 1. Create a Webex integration to obtain a ClientID and a client secret.
 2. Use them to obtain an one-time short-live **Authorization Code**.
 3. Exchange the Authorization Code for an **Access Token**.

You can now use this access token (similar to the portal one) to access the APIs. However, this token will expire in ~12 hours, so you need to do steps 1-3 again. To avoid that:

4. In previous step, alongside the access token, you will receive a **Refresh Token**.
5. Use the refresh token to renew your access token (get a new one).

API Authentication – Continuous Access

REFERENCE

- [Authentication](#) process for continuous access (e.g. application, bot) requires the following steps:
 1. Create a Webex integration to obtain a ClientID and a client secret.

Congratulations! 🎉

Your integration has been created. Use the OAuth credentials below to finish building your integration.

CL_DataLab

OAuth settings

Client ID
C2aa4324f490052708a9fc8cc888fcc7b798a91bc96ceebf689b33716d2bc20d7 [Copy](#)

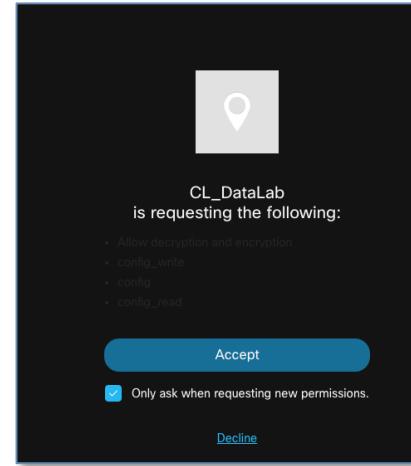
Client Secret
baf853ccc04d92a6d41c4f7a70087ec9d804196fd5e38a28ba91756a6fbe97a [Copy](#)

OAuth Authorization URL
You can use the URL below to initiate an OAuth permission request for this app. It is configured with your redirect URI and app scopes. Be sure to update the state parameter.
[https://webexapis.com/v1/authorize?](https://webexapis.com/v1/authorize?client_id=C2aa4324f490052708a9fc8cc888fcc7b798a91bc96ceebf689b33716d2bc20d7&response_type=code&redirect_uri=https%3A%2F%2Fgoogle.com&scope=spark%3Akms%20cj%3Aconfig_write%20cj%3Aconfig%20cj%3Aconfig_read%20cjds%3Aadmin_org_read%20cjds%3Aadmin_org_write&state=set_state_here)

API Authentication – Continuous Access

REFERENCE

- [Authentication](#) process for continuous access (e.g. application, bot) requires the following steps:
 1. Create a Webex integration to obtain a ClientID and a client secret.
 2. Use them to obtain an one-time short-live Authorization Code.



https://www.google.com/?code=ZGM3MjlkNDItNGZINS00M2YzLWEzMmEtNWU0NmFmOGNjN2MyNjA3ZTkyZDUtOGIx_P0A1_e56f00d4-98d8-4b62-a165-d05a41243d98&state=set_state_here

API Authentication – Continuous Access

REFERENCE

- Authentication process for continuous access (e.g. application, bot) requires the following steps:
 1. Create a Webex integration to obtain a ClientID and a client secret.
 2. Use them to obtain an one-time short-live Authorization Code.
 3. Exchange the Authorization Code for an Access Token.

You can now use this access token (similar to the portal one) to access the APIs. However, this token will expire in ~12 hours, so you would need to do 1-3 again. To avoid that:

4. In previous step, alongside the access token, you will receive a Refresh Token.

Step 2: Exchanging the Authorization Code for an Access Token

Your app should take the one-time short-lived code obtained in the previous step and exchange it for an access token. To get an access token, please make a request to https://webexapis.com/v1/access_token with the parameters as shown below:

```
curl --location --request POST 'https://webexapis.com/v1/access_token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=authorization_code' \
--data-urlencode 'client_id=<client_id>' \
--data-urlencode 'client_secret=<client_secret>' \
--data-urlencode 'redirect_uri=<redirect_uri>' \
--data-urlencode 'code=<code>'
```

The screenshot shows a Postman interface with a POST request to https://webexapis.com/v1/access_token. The 'Body' tab is selected, showing form-data parameters:

Key	Value	Description
grant_type	authorization_code	
client_id	Cf1ae118a6d6e955d0e7688374a928716b24e7aaec04086288d1c...	
client_secret	30b128c7e24c7504bf333ed17396719983e69202beefef307131af0...	
redirect_uri	https://auth.pstmn.io/v1/callback	
code	ZjAwWJUIMYYIOTBjOC00YjM0LTIM2QYTC10DFIODY1MWNmYjINTK2...	

The response body shows the JSON structure of the access token:

```
1 { "access_token": "N2ZY5104WYyN0D10500BD11W11ZGhzZ", "token_type": "Bearer", "expires_in": 1269599, "refresh_token": "06MT1CryYzt20Yzly0MThzT1L0GtYVAYTEjNjRkZjM40tg3ZmTwZatH02_P0A1_e56f00d4-98d8-4b62-a165-d05a1243d98", "refresh_token_expires_in": 775999, "scope": "spark:cams c:jconfig_write c:jconfig c:pconfig c:pconfig:read" }
```

API Authentication – Continuous Access

REFERENCE

- Authentication process for continuous access (e.g. application, bot) requires the following steps:
 1. Create a Webex integration to obtain a ClientID and a client secret.
 2. Use them to obtain an one-time short-live Authorization Code.
 3. Exchange the Authorization Code for an **Access Token**.

You can now use this access token (similar to the portal one) to access the APIs. However, this token will expire in ~12 hours, so you need to do steps 1-3 again. To avoid that:

4. In previous step, alongside the access token, you will receive a **Refresh Token**.
5. Use the refresh token to renew your access token (get a new one).

Step 4: Using the Refresh Token

After the access token expires, APIs will no longer accept requests with the token. If your users stay logged in for a long time or your app needs to make background requests on behalf of a user, your app must use a `refresh_token` to get a new `access_token` before the current `access_token` expires.

To get a new access_token, make a POST request to https://webexapis.com/v1/access_token, by setting grant_type to "refresh_token" as follows:

```
curl --location --request POST 'https://webexapis.com/v1/access_token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=refresh_token' \
--data-urlencode 'client_id=<client_id>' \
--data-urlencode 'client_secret=<client_secret>' \
--data-urlencode 'refresh_token=<refresh_token>'
```

The screenshot shows a Postman interface with a POST request to https://webexapis.com/v1/access_token. The 'Body' tab is selected, showing the following JSON payload:

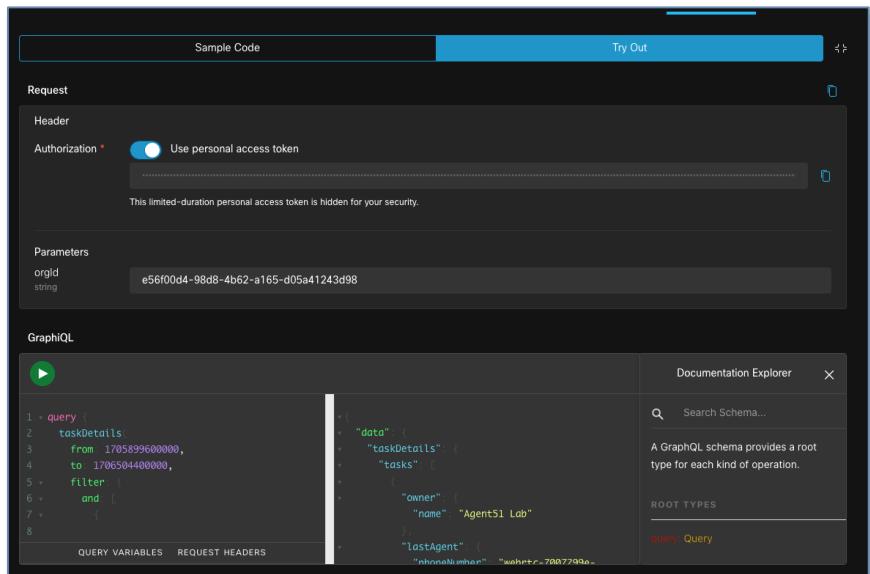
```
{  
  "grant_type": "refresh_token",  
  "client_id": "CfHe1lsade955de7888374a9287fb24e7ee0c4086288d1cb...",  
  "client_secret": "30812bc7b24c7504bf8f333e6d17396719983e892028eef307131eb0...",  
  "refresh_token": "ZmITIVTczYztGvzytD0MTMzLTHOGHYZVhTE2NjhZM40tg3mf..."  
}
```

The response status is 200 OK, and the response body is:

```
{  
  "access_token": "9C13mfJNcc0M0dlyRflnJLTg1YjgtNnZvZtX1TTZxZUyMrLz1M0gtH2",  
  "expires_in": 1248599,  
  "refresh_token": "2d11fc1c72720ry80fTHzLTL10GtV1yTE2hJnZ",  
  "refresh_token_expires_in": 7775785,  
  "token_type": "Bearer",  
  "scope": "https://api.cisco.com/cpsrconfig/write cpsrconfig/jar/config/read"  
}
```

GraphQL for Search API

- GraphQL is a query language for declarative data fetching.
- Relationship between Analyzer repositories and GraphQL queries:
 - CSR / CAR -> **taskDetails**
 - ASR / AAR -> **agentSession**
 - CLR (Queue Records) -> **taskLegDetails**
- Two types of queries:
 - Query to fetch **raw data**
 - Query to **perform aggregations**



The screenshot shows a GraphQL playground interface. At the top, there's a "Sample Code" button and a "Try Out" button. Below that is a "Request" section with a "Header" tab and an "Authorization" field set to "Use personal access token". A note says "This limited-duration personal access token is hidden for your security." Under the "Parameters" tab, there's an "orgId" field with the value "e56f00d4-98d8-4b62-a165-d05a41243d98". In the "GraphQL" section, a green play button is followed by a code editor containing the following GraphQL query:

```
1 - query
2   taskDetails
3     from 1705899600000,
4     to 1706504400000,
5     filter {
6       and: [
7         {
8           ...
```

Below the code editor are "QUERY VARIABLES" and "REQUEST HEADERS" buttons. To the right, there's a "Documentation Explorer" panel with a search bar and a note: "A GraphQL schema provides a root type for each kind of operation." It also lists "ROOT TYPES" and "query: Query".

Anatomy of a GraphQL Query (Raw Data)

1. Specify the data repository
2. Query Range (in epoch)
3. Set filters (optional)

```
{  
  taskDetails( ①  
    from: 1698848027000  
    to: 1701320400000 ②  
    filter: {  
      and: [{ totalDuration: { gt: 60000 } }, { totalDuration: { lt: 120000 } }] ③  
    }  
  ) {  
    tasks { ①  
      id  
      channelType  
      lastAgent {  
        id  
      }  
      lastQueue {  
        id  
      }  
      totalDuration  
      connectedDuration  
      holdDuration  
    }  
    pageInfo {  
      hasNextPage  
      endCursor  
    }  
  }  
}
```

The diagram illustrates the components of a GraphQL query. It shows a code snippet with three numbered annotations:

- Annotation 1: Points to the top-level fields "taskDetails" and "tasks".
- Annotation 2: Points to the "from" and "to" parameters within the "taskDetails" field.
- Annotation 3: Points to the "filter" field within the "taskDetails" field.

Annotations also point to specific parts of the query results, such as "Range of the query" and "Filters" for the "taskDetails" field, and "Fields requested" for the "tasks" field.

GraphQL Query - Filters

- **Filters** allow users to specify the results they want to fetch besides the query time period.

- Syntax:

```
filter : {  
    fieldName : { operator : value }  
}
```

- Operator on scalar (textual) fields can be any of the following: **equals, notequals, match, contains**.
 - This means that filters allow both for exact matching or partial matching with the use of RegEx.
 - For numerical fields, we can use **equals, notequals, gt, gte, lt, lte**.
 - We can also use **and & or** operators to create multiple filter criteria:

```
filter: {  
  and: [  
    { channelType: { equals: telephony } }  
    {  
      or: [  
        { status: { equals: "ended" } }  
        { status: { equals: "created" } }  
      ]  
    }  
  ]  
}
```

Anatomy of a GraphQL Query (Raw Data)

1. Specify the data repository
2. Query Range (in epoch)
3. Set filters (optional)
4. Specify the fields you want to fetch

The diagram shows a GraphQL query with the following structure:

```
{  
  taskDetails(  
    from: 1698848027000  
    to: 1701320400000  
    filter: {  
      and: [{ totalDuration: { gt: 60000 } }, { totalDuration: { lt: 120000 } }]  
    }  
  ) {  
    tasks {  
      id  
      channelType  
      lastAgent {  
        id  
      }  
      lastQueue {  
        id  
      }  
      totalDuration  
      connectedDuration  
      holdDuration  
    }  
    pageInfo {  
      hasNextPage  
      endCursor  
    }  
  }  
}
```

- Annotation 1: Points to the `taskDetails` field.
- Annotation 2: Points to the `from` and `to` arguments within the `taskDetails` field. A callout indicates this is the "Range of the query".
- Annotation 3: Points to the `filter` argument within the `taskDetails` field. A callout indicates this is the "Filters".
- Annotation 4: Points to the list of fields requested under the `tasks` field. A callout indicates this is the "Fields requested".

Anatomy of a GraphQL Query (Raw Data)

1. Specify the data repository
2. Query Range (in epoch)
3. Set filters (optional)
4. Specify the fields you want to fetch
5. Add Pagination details

The diagram shows a GraphQL query with the following structure:

```
{  
  taskDetails(  
    from: 1698848027000  
    to: 1701320400000  
    filter: {  
      and: [{ totalDuration: { gt: 60000 } }, { totalDuration: { lt: 120000 } }]  
    }  
  ) {  
    tasks {  
      id  
      channelType  
      lastAgent {  
        id  
      }  
      lastQueue {  
        id  
      }  
      totalDuration  
      connectedDuration  
      holdDuration  
    }  
    pageInfo {  
      hasNextPage  
      endCursor  
    }  
  }  
}
```

- 1: Task Details field.
- 2: Range of the query (from/to).
- 3: Filters applied to the query.
- 4: Fields requested (id, channelType, lastAgent.id, lastQueue.id, totalDuration, connectedDuration, holdDuration).
- 5: Pagination details (hasNextPage, endCursor).

GraphQL Query - Pagination

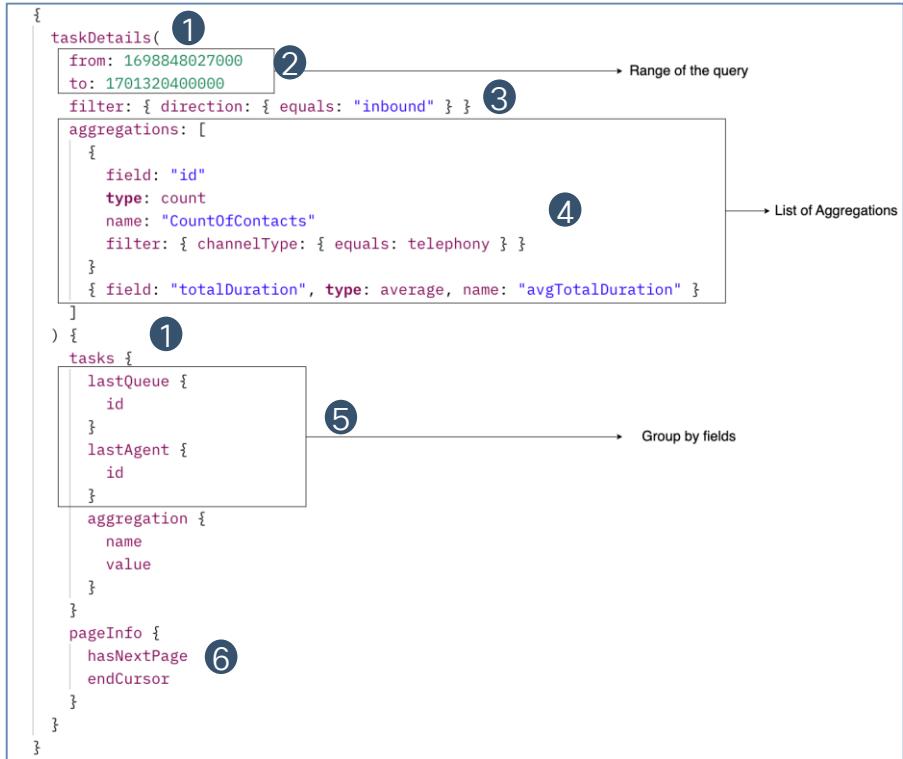
- Since it is possible that an API request may return a large dataset, [pagination](#) allows to divide the dataset into smaller chunks (pages).
- Search API uses following fields to be able to support fetching subsequent sets of records:
 - `hasNextPage`: Boolean field that determines if more data exists in the given query
 - `endCursor`: String field that acts as identifier for the next page
 - `Pagination`: Accepts the cursor argument, which indicates which page to retrieve

```
{  
  taskDetails(  
    from: 1690964812000  
    to: 1691224012000  
    pagination: { cursor: "<VALUE-OF-endCursor-FROM-PREVIOUS-QUERY" }  
  ) {  
    tasks {  
      id  
    }  
    pageInfo {  
      hasNextPage  
      endCursor  
    }  
  }  
}
```

Anatomy of a GraphQL Query (Aggregations)



1. Specify the data repository
2. Query Range (in epoch)
3. Set filters (optional)
4. Specify the aggregations you want to perform
5. Select the Group
6. Add Pagination Details



GraphQL Query – Aggregations & Group By

- Aggregations allow users to perform and retrieve the results of arithmetic operations of the requested data.

- Syntax:

- **Field**: The requested variable we want to perform aggregation on
- **Type**: The kind of operation we want to perform. Possible values: count, min, max, sum, average, cardinality
- **Name**: A user friendly identifier string for the result of the aggregation operation
- Optionally, we can define the grouping we want to perform on the results (similar to Row Segments).
- Every aggregation operation should conclude with the **aggregation** part, which is basically used to fetch the results.

```
aggregations: [  
  {  
    field: "totalDuration"  
    type: max  
    name: "maxDurationTelephony"  
  }  
]
```

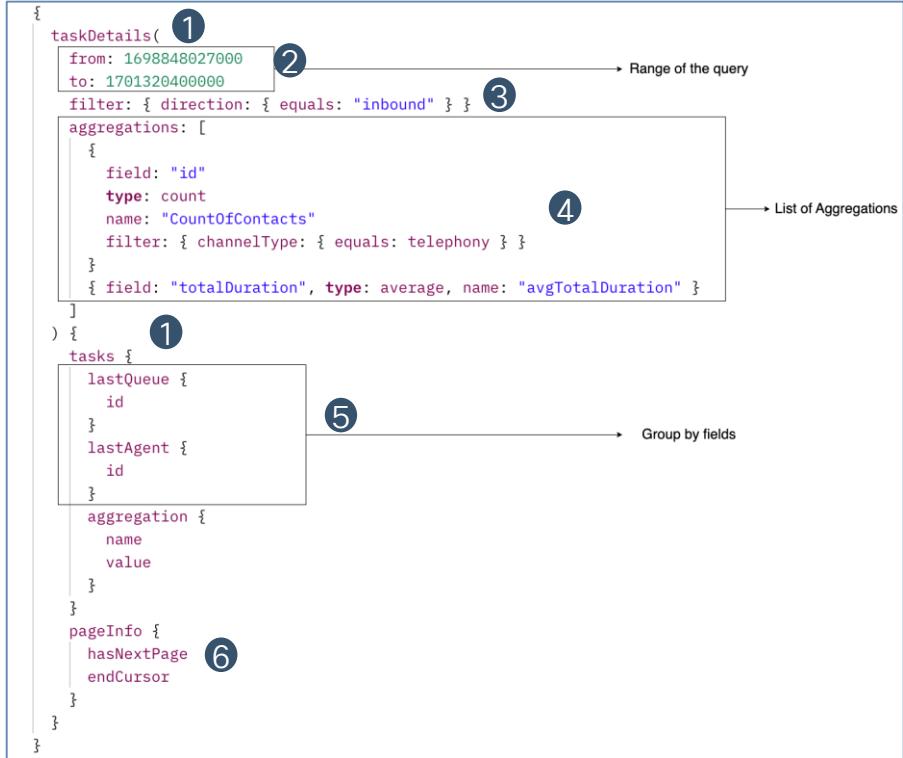
```
tasks {  
  lastAgent {  
    id # Group by on lastAgent.id  
  }  
}
```

```
aggregation {  
  name  
  value  
}
```

Anatomy of a GraphQL Query (Aggregations)



1. Specify the data repository
2. Query Range (in epoch)
3. Set filters (optional)
4. Specify the aggregations you want to perform
5. Select the Group
6. Add Pagination Details



Lab 3: APIs

 15 min

Exercise 3.1: Build a Search API from scratch

Exercise 3.2: API Authorization (Bonus)

Exercise 3.3: Extract the GraphQL Query using the browser

What did we learn?

- Authenticate using OAuth2, get an access & refresh token
- The building blocks of a GraphQL query, how to build an API based on our needs
- How to take advantage of Analyzer UX Refresh & browser tools to build an API query based on a report

Analyzer & APIs: Extract the Query from Portal

NEW

- Can we extract the schema of a report to directly use an API instead of building it?
- Each Analyzer version is utilizing a different architecture concept:
 - Analyzer -> REST APIs X
 - Analyzer Beta -> GraphQL APIs ✓
- This means we can only use UX Refresh reports.
- We can use the Network tab from the Browser Developer tools to inspect the report query ([guide](#)).
- Note: The extracted JSON is not a ready drag&drop GraphQL query, we need to add information to the query for a successful API request. Thus, this capability is only used as a guideline on building a query, not as a direct fetch of an Analyzer report via API.

The screenshot displays a web application interface for 'Agent Detail Report'. At the top, there are tabs for 'Stack' and 'Transition'. Below the tabs, there are sections for 'Agent Detail Report' and 'Duration' (set to 'Last Week'). The main area shows a table of data with columns: Agent Name, Extension, Call Start Time, Call End Time, Duration, Called No., Call All., Call Route., Other CSQs, Talk Time, Hold Time, Work Time, and Call Direct. The data lists various agents (Agent100 Lab, Agent100 Lab, Agent100 Lab, etc.) with their respective call details. Below the table, the 'Network' tab of the browser's developer tools is open. It shows a list of network requests, with one specific request highlighted in red. The highlighted request is a POST to 'www.cisco.com/west-us-1/api/1.0/reporting/agent/100/refresh'. The 'Headers' tab shows the request headers, and the 'Request' tab shows the complex JSON payload being sent. The payload includes fields like 'query', 'from', 'to', 'filters', 'sort', 'groupby', 'pageinfo', and 'meta'. The 'Response' tab shows the server's response, which is a large JSON object containing detailed report data.

How to transform the extracted Query to an API

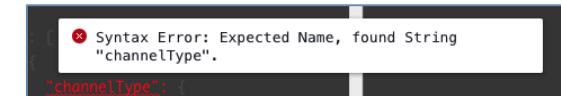
REFERENCE

- As we can see in the [video guide](#), after extracting the Query from the Network tab, it is not in proper format to be run as an API.
- There are two major changes we need to make to the query to fix it:
 - All the **fields** have quotes (" ") around them that we need to remove. Quotes should only remain to the values, not the fields.



- There is a string `"extFilter": {}`, in the query, which we need to delete.

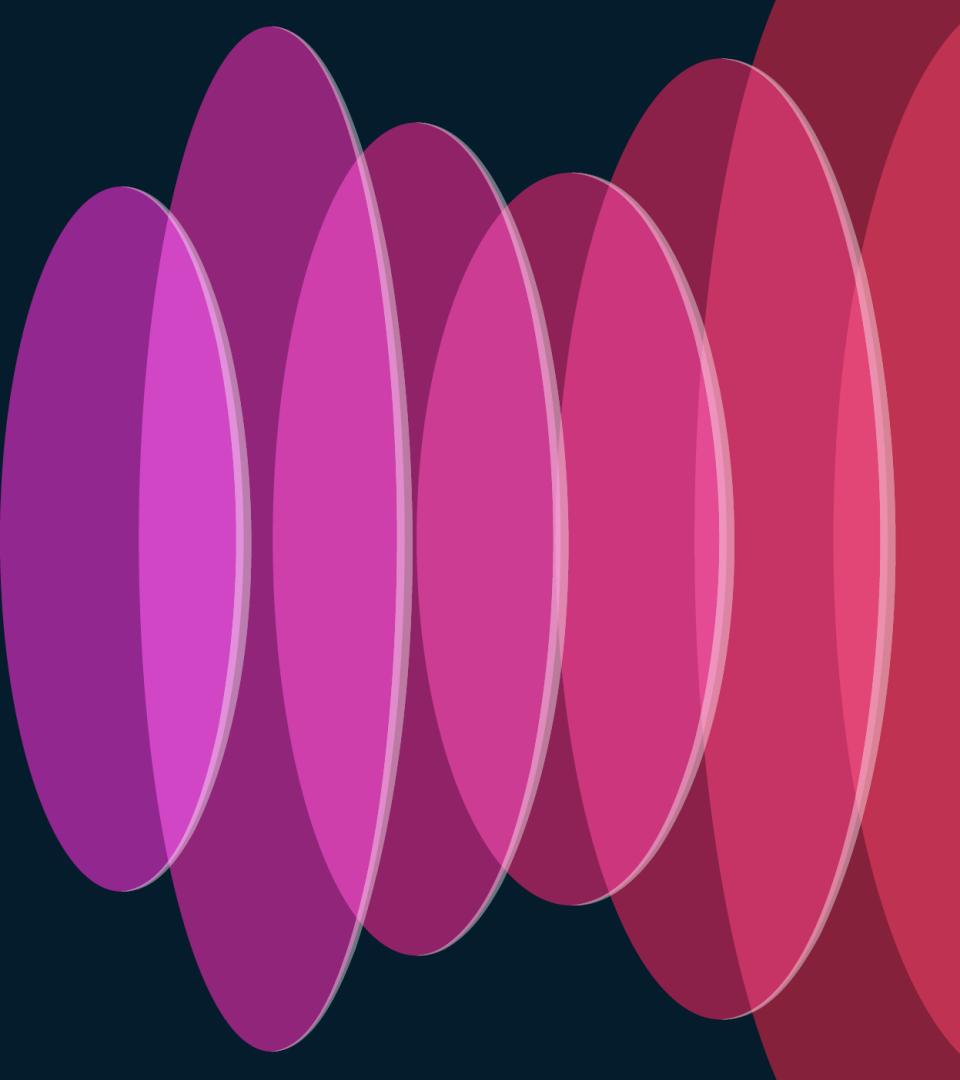
Keep in mind that if you test the API using the Developer portal Try Out option, you will be able to get descriptive error messages for all the things you need to correct.



If you have access to a GPT-powered solution, you can ask it to make the transformation for you using the following question:

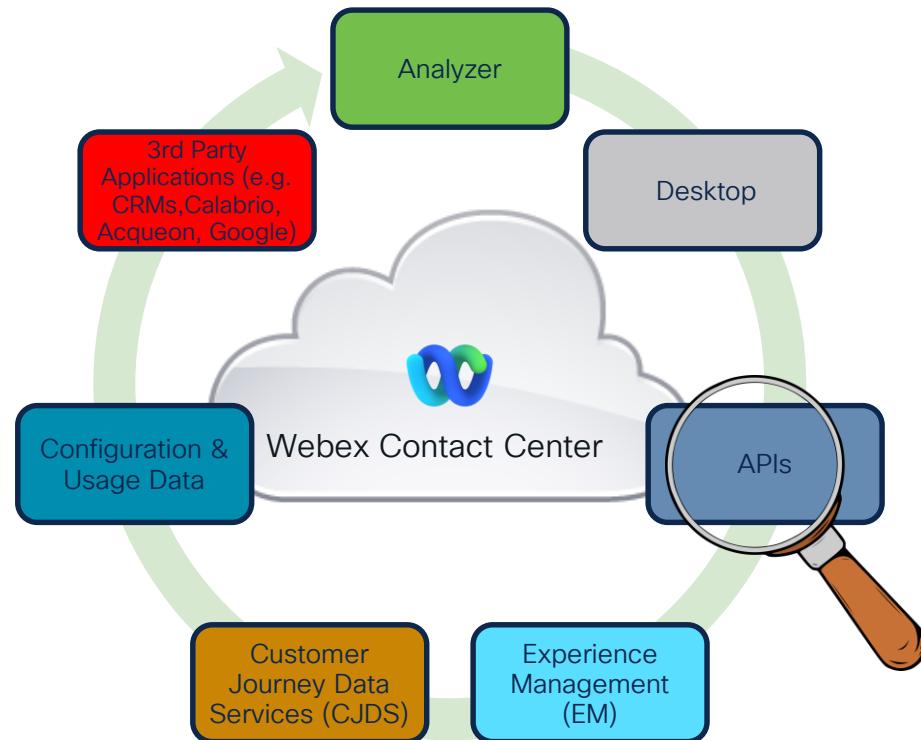
"Take this raw query with variables and construct the GraphQL query from it by replacing all the \$variables using variables values in the variable section. The goal is to be able to paste the prepared GraphQL query directly into a GraphQL editor.
<PasteYourExtractQuery>"

Experience Management



NEW

What is EM?

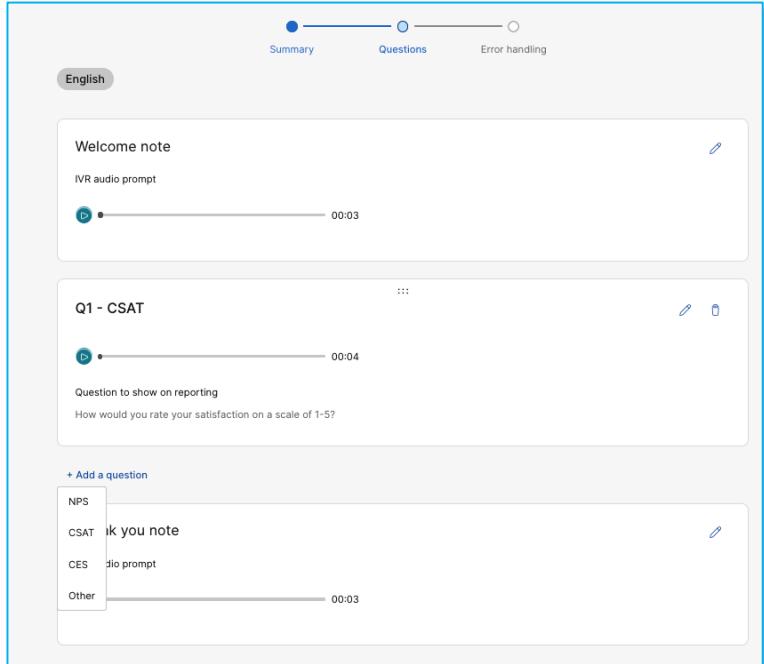


- **Experience Management (EM)** is a feature suite aimed to effortlessly survey customers for their feedback across all channels.
- Today, we have the option to create post call [IVR surveys](#) to grab customer's feedback via DTMF after the agent interaction has ended.
- Feature currently only available in US, coming in EU in the next few weeks.
- Part of standard Webex CC, no additional cost for IVR Surveys.

Survey Builder



- Survey Builder in Control Hub includes:
 - A welcome prompt
 - One or more survey questions
 - A thank you note
 - A prompt for invalid input or timeout
- Responses can be **downloaded** from the **Control Hub** in a CSV file.
- Flow Designer provides control on if we will provide customer with a survey or not (Global Variable **Global_FeedbackSurveyOptIn**).





Future of Experience Management

- IVR Survey Global Rollout
- Storing survey results in Analyzer
- Release APIs for survey data access
- Digital Surveys
- Conditional Surveys

Lab 4: Experience Management

 5 min

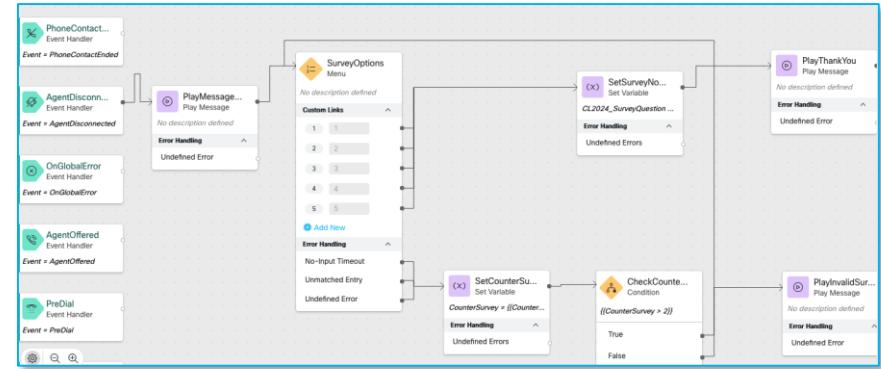
Exercise 4.1: Explore the new Experience Management Survey Builder (Demo)

What did we learn?

- How to configure a survey (prompts, questions)
- Downloading survey results
- Use Analyzer stock reporting to gather survey statistics

But I want IVR Surveys today!

- We can create “manual” IVR surveys directly in Flow Designer:
 - Use a **Menu** node with all the scoring options
 - Add the question (e.g. NPS score) as a prompt to Menu node
 - Store the response in a reportable Global Variable using the **Set Variable** node.
 - This sub-flow needs to be added in the **AgentDisconnected** Event Flow.
- Since the response is simply a Global Variable, we can create custom reports based on them in Analyzer.
- In all surveys (both EM & manual ones), it is mandatory that **Agents disconnect the call from the Desktop, not from their device**.



Lab 4: Experience Management



5 min

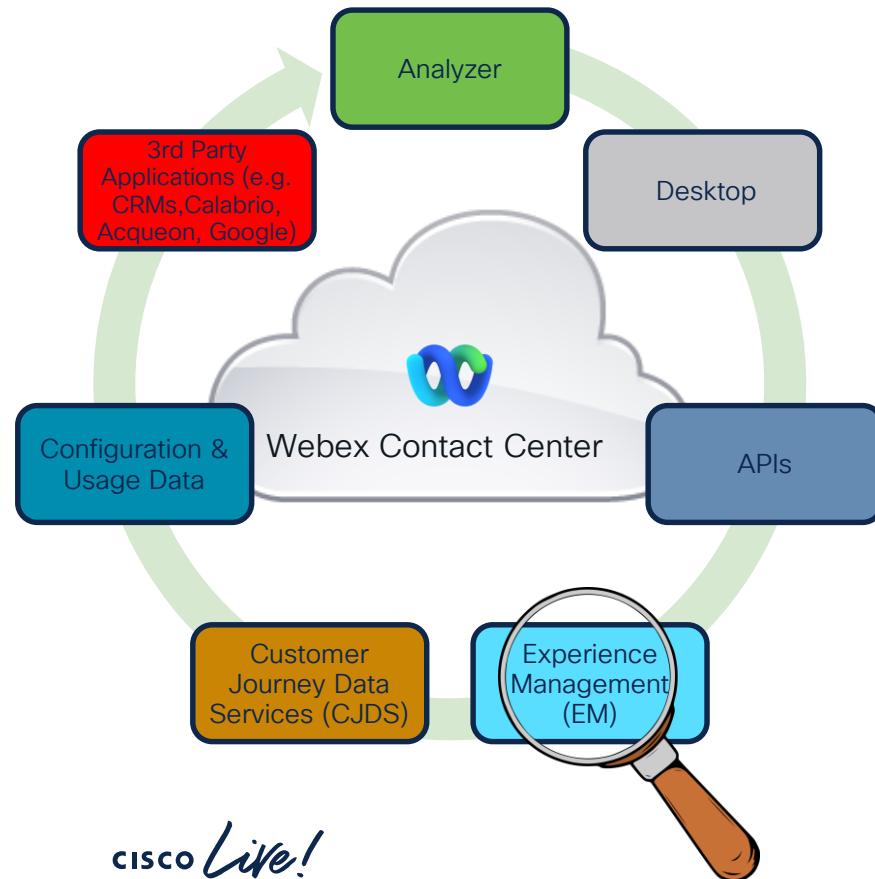
Exercise 4.2: “Manual” surveys using the Flow Designer

What did we learn?

- It is possible to create a type of Post Call Survey today without using EM
- Although it provides more reporting options today, it is only meant as a workaround until EM survey data become available in Analyzer.

Customer Journey Data Services (CJDS)

What is CJDS?



- **Journey Data Services (JDS)** is an API-first customer journey management service that allows users to capture a customer's journey across multiple channels or applications, identify insights and take real-time actions.
- Limited Availability in US, coming in EU in the near future alongside Experience Management.
- No additional cost, part of standard agent offering.
- Customer Journey Data [FAQ](#)

NEW

CJDS Developer Portal

Journey - Getting Started

webex Contact Center for Developers Documentation Support

Skill Profile Subscriptions SureyToken Tasks Team User Profiles Users Work Types Quicks AI Contact Control APIs Desktop Digital Transcript Json Details Getting Started With Search ... Journey - Getting Started Using Webhooks HELP FAQ Devnet Community

Journey - Getting Started

How does CJDS work?

- CJDS is an API-first service that enables organizations to:
 - Listen: Integrate with any data source or third-party applications to listen to disparate data sources.
 - Identify: Create a dynamic customer profile capturing propensity drivers, such as a customer's preferred mode of communication or preferred language.
 - Analyze: Apply different aggregation techniques to all customer data collected.
 - Act: Use the data and insights within CJDS to dynamically change the flow within Webex Contact Center Flow Control and personalize the customer experience at a granular level. These insights are visible to customer-facing teams in real time through Agent Desktop.

CJDS Overview Vidcast

- <https://app.vidcast.io/share/889c2cbf-51b2-4cc9-94fb-9143078dca83>

How to set up CJDS?

- CJDS is currently in Limited Availability (for US-only), please fill out this form to be onboarded. The Cisco team will assist you with CJDS instance setup.

Technical Details

Authentication for CJDS APIs

- CJDS uses Bearer Auth for all APIs. You will need to use CI generated Bearer Token in the 'Authorization' header. Prefix token with 'Bearer'. Please look at the Authentication section on how to generate bearer token. You can also get the Bearer token from <https://developer.webex-cx.com> login with your admin credentials. It requires a full administrator role, CJDS appropriate cjp:config_read or cjp:config_write scopes.

Contents

- How does CJDS work?
- CJDS Overview Vidcast
- How to set up CJDS?
- Technical Details
- Authentication for CJDS APIs

webex Contact Center for Developers Documentation Support

Journey

Customer Journey Data service is a next-generation customer journey management service that enables businesses to capture customer journeys across any channel or application, identify insights, and take real-time actions to provide an excellent customer experience. This service is currently only available for our US-based customers. Please refer to the getting started document for more information.

Get All Journey Actions

`GET /admin/v1/api/journey-actions/workspace-id/{workspaceId}`

Get all Journey Actions in JDS. It requires an administrator role and the appropriate cjp:config_read or cjp:config_write scopes

Get All Journey Actions For A Template

`GET /admin/v1/api/journey-actions/workspace-id/{workspaceId}/template-id/{templateId}`

Get all Journey Actions for a template in JDS. It requires an administrator role and the appropriate cjp:config_read or cjp:config_write scopes

Create A New Journey Action

`POST /admin/v1/api/journey-actions/workspace-id/{workspaceId}/template-id/{templateId}`

Create a new Journey Action in JDS. It requires an administrator role and the appropriate cjp:config_write scope

Get Specific Journey Action By Action Id

`GET /admin/v1/api/journey-actions/workspace-id/{workspaceId}/template-id/{templateId}`

Get a specific Journey Action by Action Id

Contents

- Call Monitoring
- Captures
- Contact Number
- Contact Service Queue
- Desktop Layout
- Desktop Profile
- Dial Number
- Dial Plan
- Entry Point
- Estimated Wait Time
- Files
- Global Variables
- Holiday List
- Journey
- Multimedia Profile
- Outdial ARI
- Overrides
- Progressive Dialer
- Questionnaire
- Questions
- Queues

REST APIs

- Get All Journey Actions
- Get All Journey Actions For A Template
- Create A New Journey Action By Action Id
- Get Specific Journey Action By Name
- Add One/more Identities To A Person
- Merges Identities To A Primary Identity
- Get All Or A Specific Person Details
- Create A Person
- Search For An Identity Via Aliases
- Delete Specific Person By Id
- Add/Remove/Replace Details Of A Person
- Get All Template Details
- Create Template
- Get A Specific Template Searched By

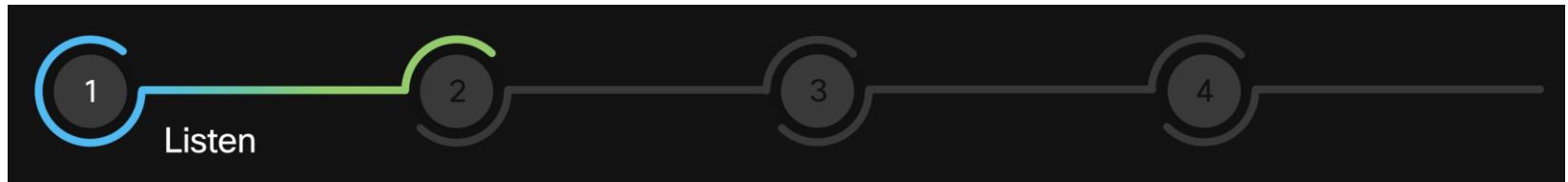
4 Stages of JDS - Listen

Listen: Integrate with any data source or 3rd party app to listen for customer events.

Project / Workspace: A uniquely identified entity that includes a specific set of JDS configurations and data sources.

- At any given moment, only one project can be active per tenant (i.e. listen to events).

Subscription: The events that JDS listens on, e.g. all the agent activity state changes/logins/logout are one default JDS subscription.



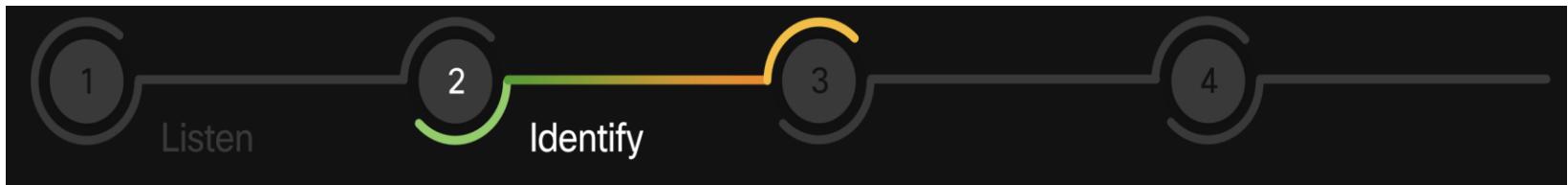
4 Stages of JDS - Identify

Identify: Create a customer profile capturing propensity drivers.

Identity / Person: A unique customer, all the events that the same customer (e.g. call, chat, email, visit website) creates are marked under the same identity.

Alias: Different ways we can identify the same customer/person (e.g. email, phone number, Customer ID). Customer must have at least one alias.

Id	First Name	Last Name	Email Addresses	Phone Numbers	Customer Ids
	John	Smith	johnsmith@gmail.com	12224440000 12224440001	111
	Jane	Doe	janedoe@gmail.com janedoe2@gmail.com	12224440002	222

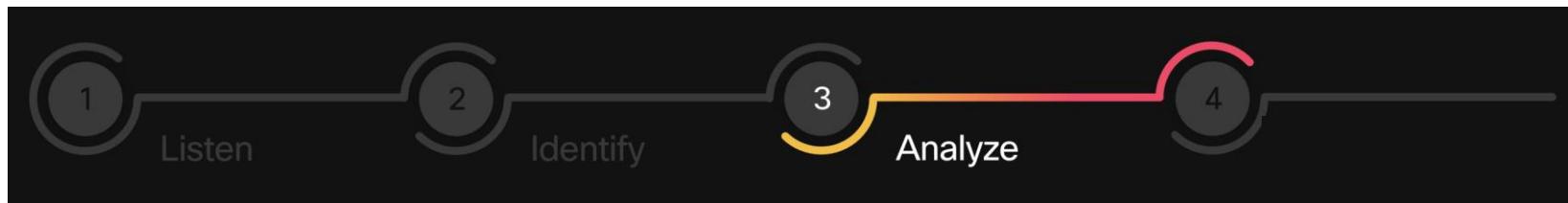


4 Stages of JDS - Analyze

Analyze: Apply aggregation techniques to the collected data.

Profile Template: A profile template defines the kind of aggregation technique we want to see for a customer (e.g. contacts within last 24 hours).

Progressive Profile: The values that correspond to an identity's profile template at that particular moment of time (e.g. contacts within last 24 hours = 10).

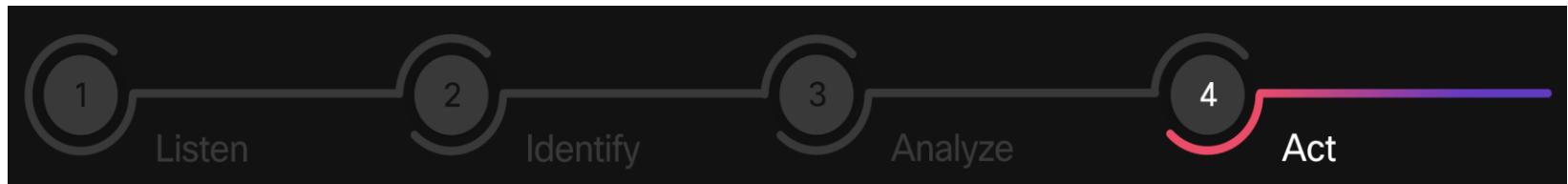


4 Stages of JDS - Act

Act: Use the gathered insights to dynamically change the flow and personalise the experience. These insights are visible to customer-facing teams in real time through Agent Desktop via the journey widget.

Actions: Actions can mean two different things in the concept of JDS:

- The logical concept of differentiating the experience based on the result of Analyze (e.g. Call priority for repeated callers).
- An automated programmatic response to events (e.g. send an SMS to a repeated caller).



CJDS Widget

The screenshot displays the Cisco Customer Journey Data System (CJDS) Widget. On the left, a chat window shows a conversation between Anuj and a user. The user has sent two messages: "Welcome to CL2024 Data Chat!" and "Welcome to Cisco Live Data & Analytics lab chat demo.". Below the messages are four input fields: Name (Anuj), Email (anubhati@cisco.com), Type (Credit), and StudentID (62). At the bottom of the chat window, there are icons for a microphone, a camera, a transfer option, and a message input field that says "Write a message to anubhati@cisco.com". On the right, a sidebar titled "Customer Journey Widget" shows a timeline of customer interactions. It includes sections for "Customer Journey" and "Contact History". The "Customer Journey" section shows "Contacts within last 10 days" and "Contacts within 24 hours". The "Contact History" section is filtered by "All" and "All Time", showing interactions from "Today 5/19/2024" and "Monday 5/6/2024". Each interaction is represented by a blue speech bubble icon and details like the time, queue, and wrap-up reason. For example, on May 19th, there was an "Inbound Chat" at 02:47 PM in the "CL2024_Data_ChatQueue" queue, and another at 02:44 PM with a "Wrap Up Reason: Sale". On May 6th, there were two more "Inbound Chat" sessions at 04:40 PM and 08:24 PM, both with a "Wrap Up Reason: Sale".

[JDS Widget Installation](#)

CJDS Widget

The screenshot shows the CJDS Widget interface. On the left, there's a chat window with a message from 'Anuj' and a form with fields for Name, Email, Type, and StudentID. On the right, there's a 'Customer Journey Widget' showing a timeline of events. Red arrows point to specific parts of the interface:

- An arrow points to the 'Identity' section, which displays the user profile: Anuj Bhatia.
- An arrow points to the 'Profile Templated / Progressive Profile Values' section, which shows contact history filters: 'Contacts within last 10 days' and 'Contacts within 24 hours'.
- An arrow points to the 'Journey Events' section, which lists a series of inbound chats with wrap-up reasons: Sale.

Customer Journey Widget

Customer Journey Contact History

Anuj Bhatia

Identity

Profile Templated / Progressive Profile Values

Journey Events

Today 5/19/2024

- Inbound Chat Ongoing 02:47 PM Queue: CL2024_Data_ChatQueue
- Inbound Chat 02:44 PM Wrap Up Reason: Sale

Tuesday 5/7/2024

- Inbound Chat 04:40 PM Wrap Up Reason: Sale

Monday 5/6/2024

- Inbound Chat 08:24 PM Wrap Up Reason: Sale
- Inbound Chat 08:21 PM Wrap Up Reason: Sale
- Inbound Chat 11:01 AM

Transfer Conference End

Name Anuj

Email anubhati@cisco.com

Type Credit

StudentID 62

Write a message to anubhati@cisco.com

[JDS Widget Installation](#)

Lab 5: Customer Journey Data Services (CJDS)

 10 min

Exercise 5.1: Familiarize yourself with JDS Desktop Widget

Exercise 5.2: Configure JDS Widget in Desktop Layout (Bonus)

Exercise 5.3: Explore the CJDS APIs (Demo)

What did we learn?

- *The basic concepts of CJDS*
- *How to setup & use the CJDS Widget in Agent Desktop to retrieve customer information in real-time*
- JDS is currently an API-first solution, with most of the capabilities being available via API

NEW

JDS API Collection

- Download the [JDS Postman Collection](#) and go play around!

The screenshot shows the Postman application interface for the JDS CiscoLive API collection. The left sidebar lists various API endpoints categorized under 'JDS CiscoLive'. The main workspace shows a 'GET' request for the endpoint `/{{baseUri}}/admin/v1/api/person/workspace-id/{{workspaceId}}/aliases/{{{identity}}}`. The 'Headers' tab is selected, displaying the following configuration:

Key	Value	Description
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.36.1	
Accept	/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Content-Type	application/json	

The 'Response' section contains a small cartoon character icon and the placeholder text 'Click Send to get a response'.

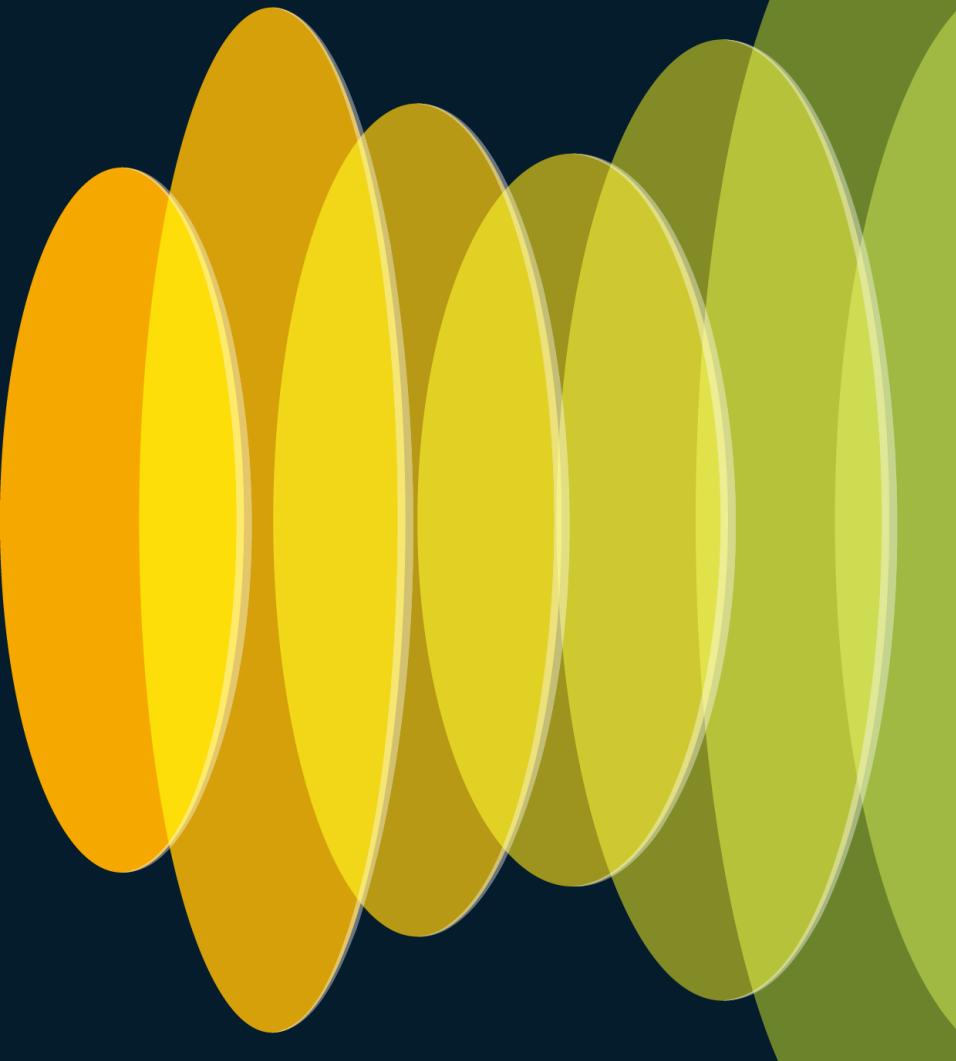
Deep Dive into JDS

- Cisco provides a series of use cases with video walkthrough and the configuration available to allow users to ramp up to more complex scenarios.
- Use Cases
 - Scenario #1: Directing Customer Input ([guide](#) / [video walkthrough](#))
 - Scenario #2: Repeat Caller Identification ([guide](#) / [video walkthrough](#))
 - Scenario #3: Efficiently Handle Repeat Callers ([guide](#) / [video walkthrough](#))
- Guides
 - JDS Postman Collection ([video walkthrough](#))
 - Create identities via API ([video walkthrough](#))
 - Merging identities & adding aliases ([video walkthrough](#))

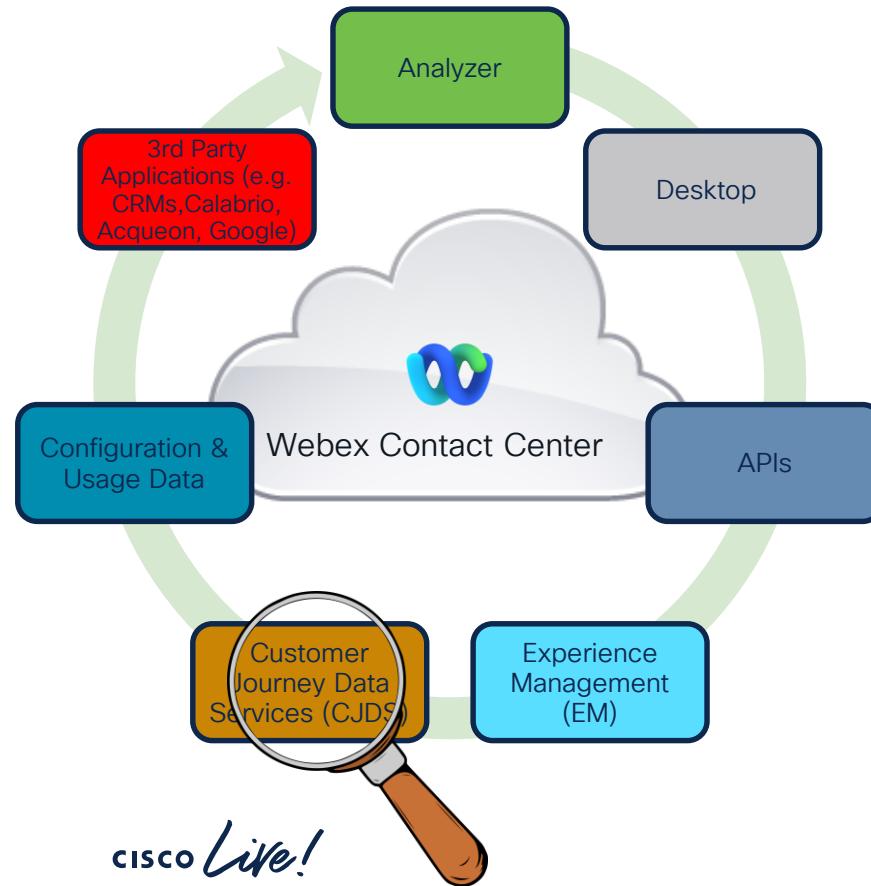
Where does the Journey continue?

- CJDS Global Rollout with widget 10.0
- Interaction Intent and Summary for Journey Widget
- Customer Sentiment Insights for Intelligent Routing
- Ingest EM Data into CJDS

Configuration & Usage Data



What are Configuration & Usage Data?

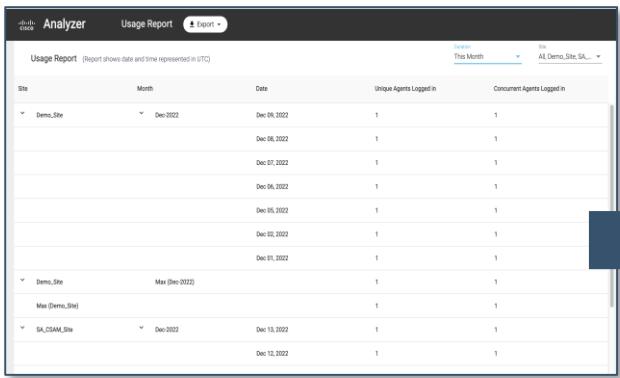


- **Usage Data** allows administrators to see the amount of **agent & IVR license usage** during the current billing cycle.
- **Configuration data** refers to all the created objects in the tenant (e.g. Users, Entry Points, Teams etc.). Administrators often need to fetch in bulk and quickly review the existing tenant configuration.

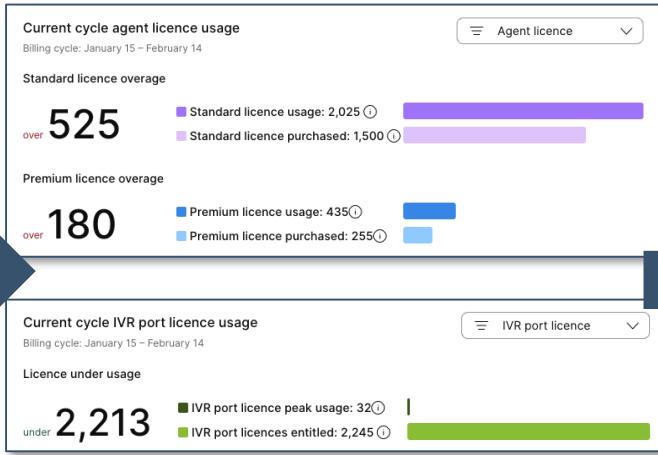
Usage Information: A Short Timeline

NEW

Usage Report



License Usage Cards



Dec 2023

Today

Coming Soon

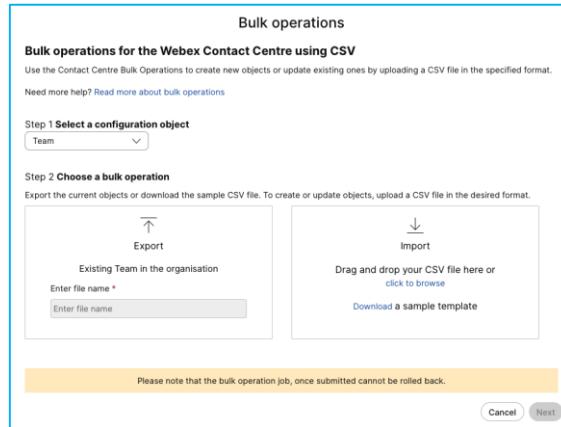
Enhancements

- Global Availability
- Detailed Daily View
- Add-on Usage (Digital Channels, Google CCAI, Acqueon, Calabrio)
- Granular Usage View in Analyzer

Configuration Data: Bulk Operations

- Bulk Operations are used to import (create & edit) or export configuration objects.
- Almost all objects inside Webex CC can be retrieved via Bulk Operations.
- A CSV file is generated when export job is done and can be downloaded at any point after the job is finished.

<ul style="list-style-type: none">• Entry Point• Dial Plan• Queue• Outdial Entry Point• Outdial Queue• Site• Team• Call Recording Schedules• Call Monitoring Schedules	<ul style="list-style-type: none">• Users• User Profiles• Work Types• Auxiliary Code• Desktop Profiles• Desktop Layouts• Multimedia Profiles• Address Book• Threshold Rules	<ul style="list-style-type: none">• Outdial ANI• Skill Definition• Skill Profile• Entry Point Mappings• Audio Files• Overrides• Business Hours• Holidays
--	---	---



Configuration Data: Configuration APIs

- [Developer Portal](#) offers comprehensive guides on how to get, create, edit or delete various configuration objects.
- Although bulk operations is an easier process if the goal is to only retrieve a configuration object type, APIs allow developers to create automations that further manipulated the fetched objects.
- [API Collection Download](#)

The screenshot shows the 'webex Contact Center for Developers' API documentation. On the left, there's a sidebar with a navigation menu including: Estimated wait time, Global Variables, Holiday List, Journey, Multimedia Profile, Notification, Outdial ANI, Overrides, Progressive Dialer, Queues, Search, Site, Skill, Skill Profile, Subscriptions, Tasks, Team, User Profiles, Users, and Work Types. The main content area is titled 'cjpconfig_read scopes.' and contains five sections: 'List Teams' (GET /organization/{orgId}/team), 'Create A Team' (POST /organization/{orgId}/team), 'Get A Team By ID' (GET /organization/{orgId}/team/{id}), 'Update A Team By ID' (PUT /organization/{orgId}/team/{id}), and 'Delete A Team By ID' (DELETE /organization/{orgId}/team/{id}). To the right of the main content is a 'Contents' sidebar listing: REST APIs, List Teams, Create A Team, Get A Team By ID, Update A Team By ID, Delete A Team By ID, and Bulk Export Teams.

Lab 6: Configuration & Usage Data

 3 min

Exercise 6.1: License Usage Cards (Demo)

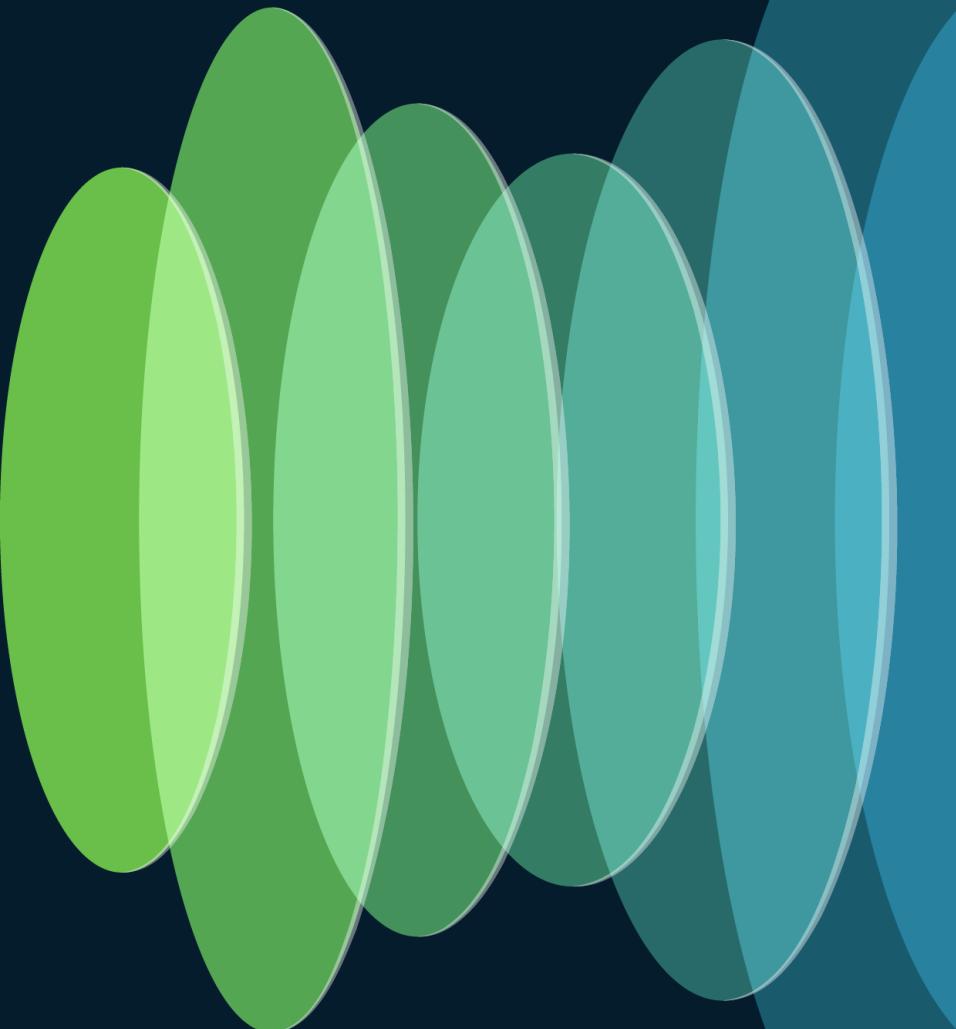
Exercise 6.2: Import/Export Configuration data from Control Hub (Bonus)

Exercise 6.3: Use APIs to Export Configuration (Bonus)

What did we learn?

- Simple but accurate license usage view, no granularity yet
- Bulk Operations -> Quick process to retrieve configuration objects
- APIs -> Sophisticated solution when we want to further manipulate the fetched data programmatically

Closing Thoughts



So this is you (after this Lab)

What did you learn?

- ✓ How to use 6 major Webex CC tools to extract data & insights
- ✓ Who are the data key stakeholders, what information they require and how to provide it to them
- ✓ Measureable Actions -> Ability to adapt, improve & differentiate your experience



What is next?

- ✓ Review this presentation, lots of material you can deep dive on
- ✓ Take advantage of the lab & the Webex space
- ✓ Get ready to party!!!

Let's meet again!



Wafik Hert



welherte@cisco.com



[Whert](#)



Anuj Bhatia



anubhati@cisco.com



[Anuj-Bhatia](#)



Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to **win 1 of 5 full conference passes** to Cisco Live 2025.



Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.



Level up and earn **exclusive prizes!**



Complete your surveys in the **Cisco Live mobile app**.



Continue your education

CISCO Live!

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand



Scan the QR code
to get started

CISCO Live!

\$25 for 5 Minutes?

Your feedback will help make our products better

Leave a Webex Contact Center product review to unlock a \$25 Visa gift card



The bridge to possible

Thank you

cisco *Live!*

#CiscoLive