The bridge to possible

Successful Migrations from Unified CM to Webex Calling

Johannes Krohn, Principal Technical Marketing Engineer

BRKCOL-2481

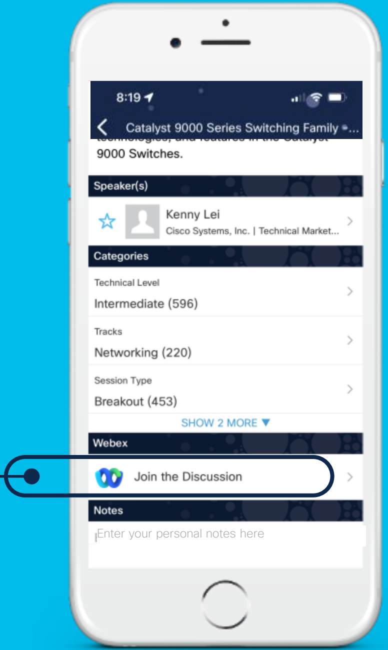
Cisco Webex App

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
until February 24, 2023.

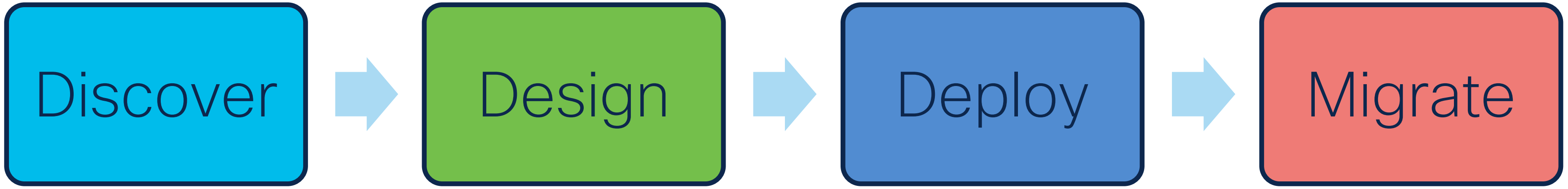
Agenda

- General Process
- Discover
- Design
- Deploy
- Migrate

… with some focus on programmability using Python

Webex Calling Migration

General Process



- Requirements
- Config assessment
- Inventory
  - users, devices, locations, …
- Feature utilization
- Integrations
- Validate network requirements

- Network requirements
- Feature mapping
- Migration batches
- Integrations
- Dial plan

- Infrastructure setup
- Base configuration
- Interworking setup
- Licensing

- Users
- Devices
- Features
- PSTN porting
- Acceptance test

General Process

Discover → Design → Deploy → Migrate

Discover

- Requirements
- ➡ Config assessment
- Inventory
  - ➡ users, devices, locations, …
- ➡ Feature utilization
- Integrations
- Validate network requirements

Design

- Network requirements
- ➡ Feature mapping
- Migration batches
- Integrations
- ➡ Dial plan

Deploy

- Infrastructure setup
- ➡ Base configuration
- ➡ Interworking setup
- Licensing

Migrate

- ➡ Users
- ➡ Devices
- ➡ Features
- PSTN porting
- Acceptance test

Discover

Analyze Existing Unified CM Setup

- Static analysis
  - Review configuration in admin GUI
  - Config export (bulk export)
  - Config export (AXL), thin/thick
  - CCUC
- Dynamic analysis
  - hard, based on trace analysis
  - UCM doesn't offer much directly
  - CCUC

Analytics – Usage Statistics

- Gather insights of existing installation
- Cloud Connected UC
  - Call volume
  - Registered endpoints
  - (CAC) locations
  - Trunk utilization
- RTMT
- …

Unified CM Data Extraction Options

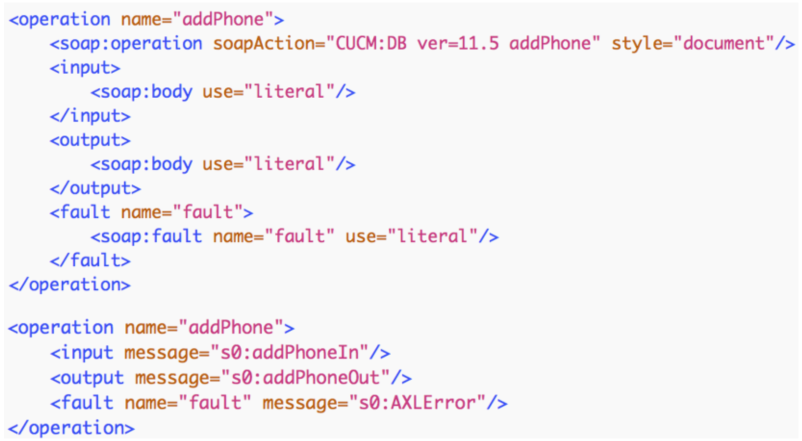
- AXL – Administrative XML
  - SOAP based provisioning API
- BAT – Bulk Administration Tool
  - CSV based
- Config Export
  - Single file Unified CM config export
- Third party: yarnlab wrangler_

AXL

- The **Administrative XML Web Service (AXL)** is an XML/SOAP based interface that provides a mechanism for inserting, retrieving, updating and removing data from the Unified Communication configuration database.
- https://developer.cisco.com/site/axl/
- Thick AXL – API defines specific objects that can be created, removed, queried, or updated
- Thin AXL – Provides a mechanism to perform direct SQL queries / updates

AXL Challenge: Interface, Object Deserialization

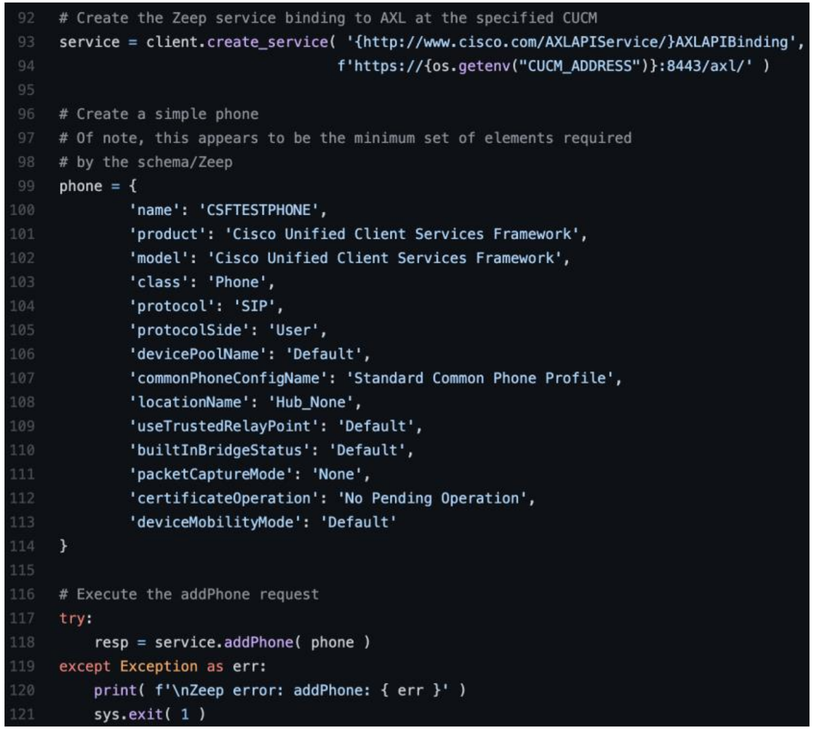
- SOAP defines interface signature (endpoint, parameters, return) in WSDL (Web Service Definition Language) files
- Idea: automatic interface and API layer creation based on WSDL
- Reality
  - Trying to avoid interface creation
  - Manual SOAP message templates
  - Tools like SoapUI simplify this.



```
<operation name="addPhone">
    <soap:operation soapAction="CUCM:DB ver=11.5 addPhone" style="document"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
    <fault name="fault">
        <soap:fault name="fault" use="literal"/>
    </fault>
</operation>

<operation name="addPhone">
    <input message="s0:addPhoneIn"/>
    <output message="s0:addPhoneOut"/>
    <fault name="fault" message="s0:AXLError"/>
</operation>
```

Solution: Zeep - Python SOAP Client

- “A fast and modern Python SOAP client”
- Python module to easily consume SOAP APIs
- “Zeep” (Dutch) = SOAP
- Consumes AXL WSDL and creates the Python interfaces

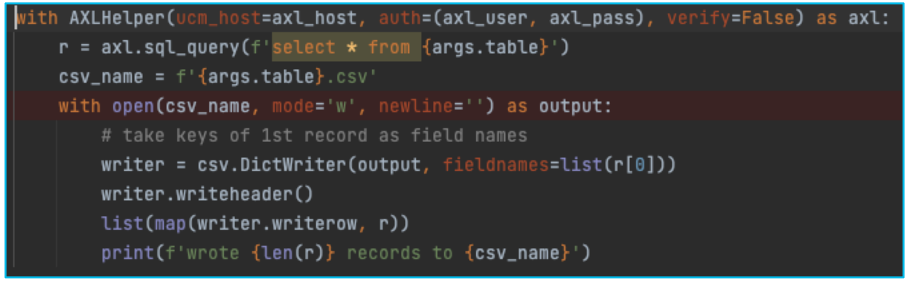


```python
# Create the Zeep service binding to AXL at the specified CUCM
service = client.create_service( '{http://www.cisco.com/AXLAPIService/}AXLAPIBinding',
                                f'https://{os.getenv("CUCM_ADDRESS")}:8443/axl/' )

# Create a simple phone
# Of note, this appears to be the minimum set of elements required
# by the schema/Zeep
phone = {
        'name': 'CSFTESTPHONE',
        'product': 'Cisco Unified Client Services Framework',
        'model': 'Cisco Unified Client Services Framework',
        'class': 'Phone',
        'protocol': 'SIP',
        'protocolSide': 'User',
        'devicePoolName': 'Default',
        'commonPhoneConfigName': 'Standard Common Phone Profile',
        'locationName': 'Hub_None',
        'useTrustedRelayPoint': 'Default',
        'builtInBridgeStatus': 'Default',
        'packetCaptureMode': 'None',
        'certificateOperation': 'No Pending Operation',
        'deviceMobilityMode': 'Default'
}

# Execute the addPhone request
try:
    resp = service.addPhone( phone )
except Exception as err:
    print( f'\nZeep error: addPhone: { err }' )
    sys.exit( 1 )
```

Examples: https://github.com/CiscoDevNet/axl-python-zeep-samples

Simple Use Case: Export Tables from Unified CM

- AXL can be used to execute SQL statements to export data from Unified CM
- Data dictionary: https://developer.cisco.com/docs/axl/#!14-cucm-data-dictionary

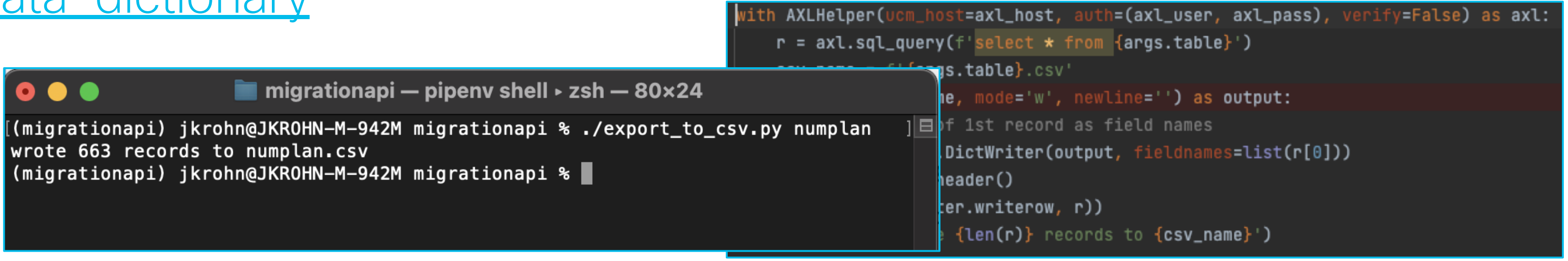


```python
with AXLHelper(ucm_host=axl_host, auth=(axl_user, axl_pass), verify=False) as axl:
    r = axl.sql_query(f'select * from {args.table}')
    csv_name = f'{args.table}.csv'
    with open(csv_name, mode='w', newline='') as output:
        # take keys of 1st record as field names
        writer = csv.DictWriter(output, fieldnames=list(r[0]))
        writer.writeheader()
        list(map(writer.writerow, r))
        print(f'wrote {len(r)} records to {csv_name}')
```

AXL helper: https://github.com/jeokrohn/ucmaxl
Tool: https://github.com/jeokrohn/migrationapi

Simple Use Case: Export Tables from Unified CM

- AXL can be used to execute SQL statements to export data from Unified CM
- Data dictionary: https://developer.cisco.com/docs/axl/#!14-cucm-data-dictionary



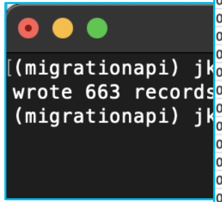
```
with AXLHelper(ucm_host=axl_host, auth=(axl_user, axl_pass), verify=False) as axl:
    r = axl.sql_query(f'select * from {args.table}')
                    s.table}.csv'
                    ne, mode='w', newline='') as output:
                    of 1st record as field names
                    DictWriter(output, fieldnames=list(r[0]))
                    header()
                    ter.writerow, r))
                    {len(r)} records to {csv_name}')
```

```
(migrationapi) jkrohn@JKROHN-M-942M migrationapi % ./export_to_csv.py numplan
wrote 663 records to numplan.csv
(migrationapi) jkrohn@JKROHN-M-942M migrationapi % 
```

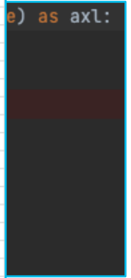
AXL helper: https://github.com/jeokrohn/ucmaxl
Tool: https://github.com/jeokrohn/migrationapi

Simple Use Case: Export Tables from Unified CM

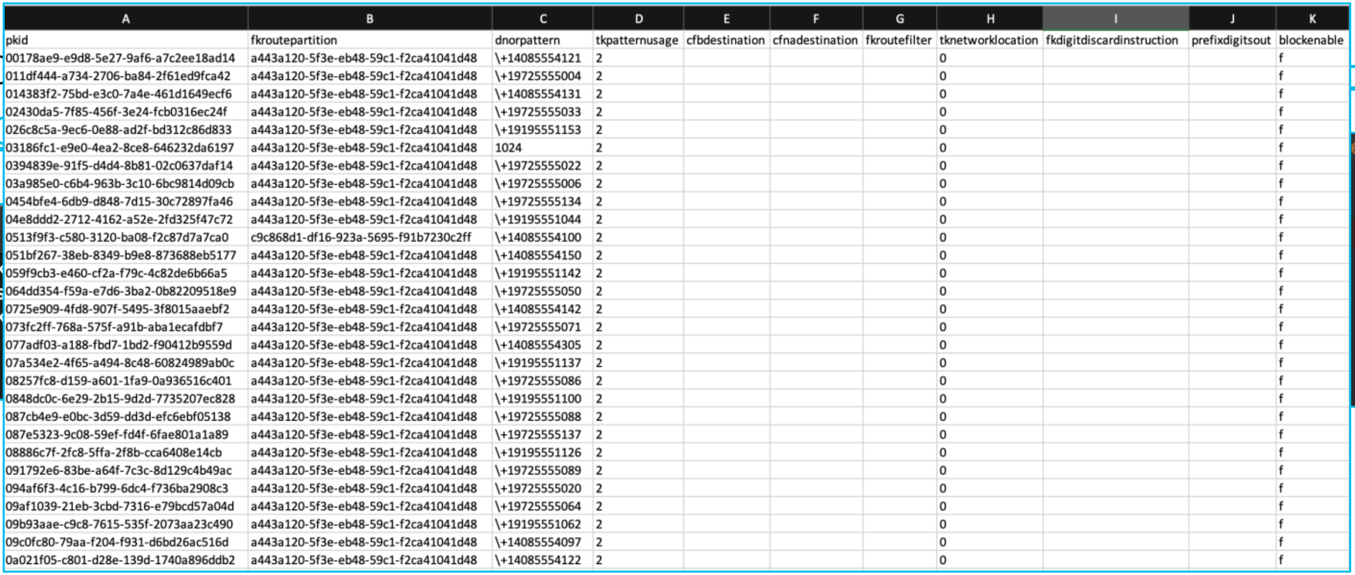
- AXL can be used to execute SQL statements to export data from Unified CM
- Data dictio
  data-dictio



```
[(migrationapi) jk
wrote 663 records
(migrationapi) jk
```



```
e) as axl:
```



A	B	C	D	E	F	G	H	I	J	K
pkid	fkroutepartition	dnorpattern	tkpatternusage	cfbdestination	cfnadestination	fkroutefilter	tknetworklocation	fkdigitdiscardinstruction	prefixdigitsout	blockenable
00178ae9-e9d8-5e27-9af6-a7c2ee18ad14	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+14085554121	2				0			f
011df444-a734-2706-ba84-2f61ed9fca42	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555004	2				0			f
014383f2-75bd-e3c0-7a4e-461d1649ecf6	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+14085554131	2				0			f
02430da5-7f85-456f-3e24-fcb0316ec24f	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555033	2				0			f
026c8c5a-9ec6-0e88-ad2f-bd312c86d833	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19195551153	2				0			f
03186fc1-e9e0-4ea2-8ce8-646232da6197	a443a120-5f3e-eb48-59c1-f2ca41041d48	1024	2				0			f
0394839e-91f5-d4d4-8b81-02c0637daf14	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555022	2				0			f
03a985e0-c6b4-963b-3c10-6bc9814d09cb	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555006	2				0			f
0454bfe4-6db9-d848-7d15-30c72897fa46	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555134	2				0			f
04e8ddd2-2712-4162-a52e-2fd325f47c72	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19195551044	2				0			f
0513f9f3-c580-3120-ba08-f2c87d7a7ca0	c9c868d1-df16-923a-5695-f91b7230c2ff	\+14085554100	2				0			f
051bf267-38eb-8349-b9e8-873688eb5177	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+14085554150	2				0			f
059f9cb3-e460-cf2a-f79c-4c82de6b66a5	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19195551142	2				0			f
064dd354-f59a-e7d6-3ba2-0b82209518e9	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555050	2				0			f
0725e909-4fd8-907f-5495-3f8015aaebf2	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+14085554142	2				0			f
073fc2ff-768a-575f-a91b-aba1ecafdbf7	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555071	2				0			f
077adf03-a188-fbd7-1bd2-f90412b9559d	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+14085554305	2				0			f
07a534e2-4f65-a494-8c48-60824989ab0c	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19195551137	2				0			f
08257fc8-d159-a601-1fa9-0a936516c401	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555086	2				0			f
0848dc0c-6e29-2b15-9d2d-7735207ec828	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19195551100	2				0			f
087cb4e9-e0bc-3d59-dd3d-efc6ebf05138	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555088	2				0			f
087e5323-9c08-59ef-fd4f-6fae801a1a89	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555137	2				0			f
08886c7f-2fc8-5ffa-2f8b-cca6408e14cb	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19195551126	2				0			f
091792e6-83be-a64f-7c3c-8d129c4b49ac	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555089	2				0			f
094af6f3-4c16-b799-6dc4-f736ba2908c3	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555020	2				0			f
09af1039-21eb-3cbd-7316-e79bcd57a04d	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19725555064	2				0			f
09b93aae-c9c8-7615-535f-2073aa23c490	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+19195551062	2				0			f
09c0fc80-79aa-f204-f931-d6bd26ac516d	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+14085554097	2				0			f
0a021f05-c801-d28e-139d-1740a896ddb2	a443a120-5f3e-eb48-59c1-f2ca41041d48	\+14085554122	2				0			f

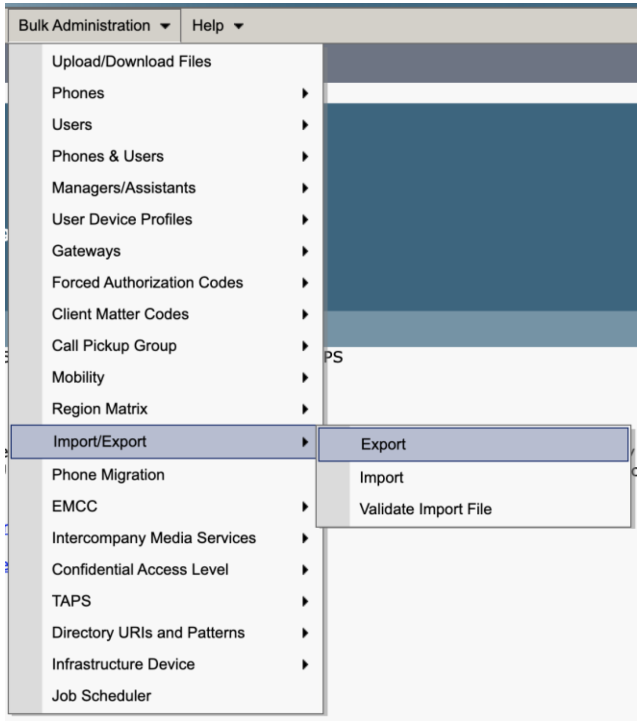
AXL helper: https://github.com/jeokrohn/ucmaxl
Tool: https://github.com/jeokrohn/migrationapi

Bulk Administration Tool (BAT)

- Main focus: simplify Unified CM provisioning
- .. but for migrations we are actually looking for the reverse
- Limited export capabilities: Users, Devices, User Device Profiles
- But then there is also config import/export

Config Export

- Specific BAT option
- Allows full or partial Unified CM config export
- Result: TAR file with 172 files
  - One CSV for each config object type
- Examples
  - callpark.csv
  - callpickupgroup.csv
  - directedcallpark.csv
  - enduser.csv
  - huntlist.csv
  - huntgroup.csv
  - …



Working with Config Export Tar and CSV Files

- Challenge: large column count in some CSVs
  - Flat DB schema
  - phone.csv has the same 122 columns for each line on the phone
  - … and if there is a single phone with 11 lines then this alone leads to 1342 columns
- Manual analysis can/will be painful
- Automation
  - Excel: import, hide columns, filter operations, …
  - programmatic

Working with CSV Files using Python

- Csv: Python standard module
  - DictReader: parse CSV file and read into Python dictionaries
- But then, how to process the data?
  - List of dicts directly
  - Pandas: Python data analysis tool (think of it as programmatic Excel)
  - Parse CSVs into Python objects and “unflatten” CSVs (e.g. create line objects from data in phone.csv)

Example: Working with CSV Files using Python

- GitHub repository: https://github.com/jeokrohn/ucmmigration
- `ucmexport.Proxy` implements "pythonic" way to access object within a tar file.
  - Not the most memory efficient ☺
  - … but plays nicely with Python IDEs (auto completion, interactive debugger, …)
- Proxy also has logic to determine phone ownership, map DN/partition to user, …

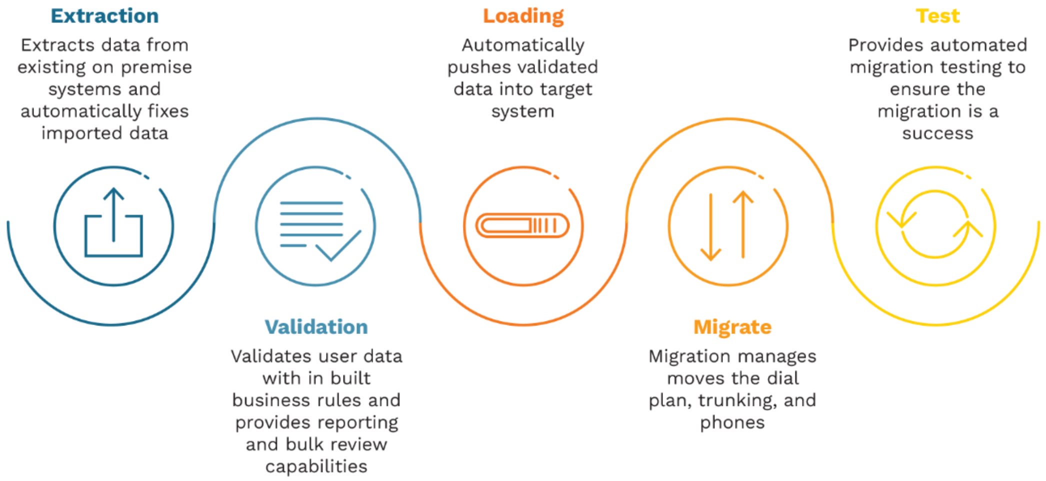
wrangler_

yarnlab

Feature Summary

- Data extraction v10.5+ from CUCM/UCXN
- Data Validation
  - Normalization
  - Transformation
  - Correction
- Configuration mapping to WbC-MT
- Validation issues for configuration changes
- Data load via APIs
- Multiple source clusters
- Phased Migration in batches
  - Inter site dependency reporting
- Dial plan analysis and connection

Migration Process - Automated Webex Calling Migration

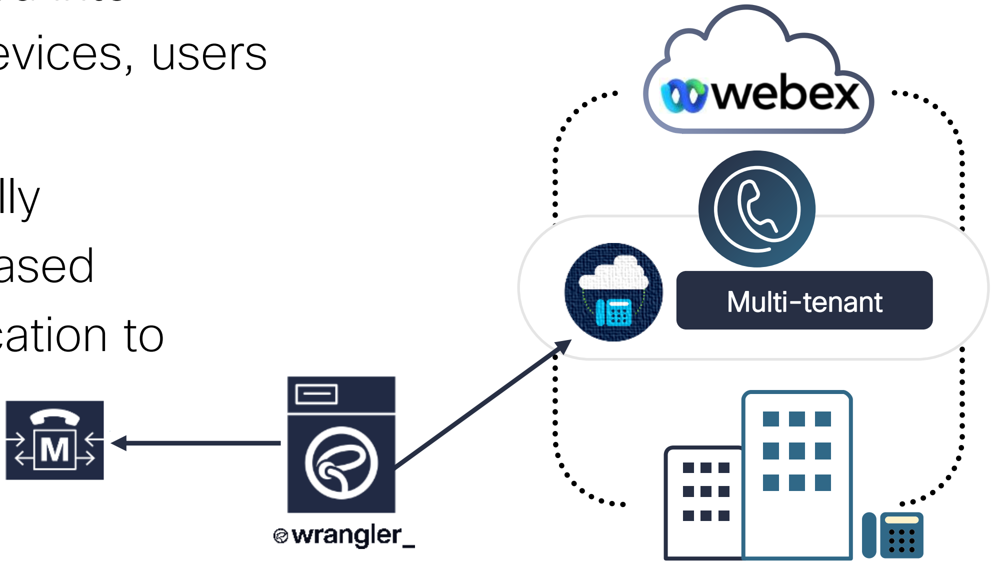

Extraction
Extracts data from existing on premise systems and automatically fixes imported data
Validation
Validates user data with in built business rules and provides reporting and bulk review capabilities
Loading
Automatically pushes validated data into target system
Migrate
Migration manages moves the dial plan, trunking, and phones
Test
Provides automated migration testing to ensure the migration is a success

CISCO Live!

yarnlab

Data Extraction

- Configuration is imported from the source and target organisation and imported into wrangler_ including all lines, devices, users and dial plan
- Configuration can be periodically rediscovered if required for phased migrations using change notification to minimize data freezes



User Migration Batches

Migration Batches

Which users have to be migrated together?

Migration Batches

Which users have to be migrated together?

- Dependencies between Users
  - Monitoring each other on BLFs
  - Used in the same hunt pilot
  - Shared lines
  - Call pick-up
  - Using the same call-park numbers
  - Intercom
  - Shared DN on phones owned by different users
  - …
- Need to make sure that users w/ dependencies are migrated together
- This information is available in the Unified CM config export
- .. But somewhat hard to extract

Example: User Relations based on Hunt Pilots

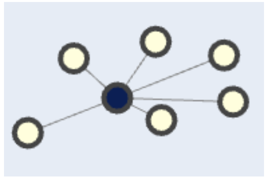
- Start with `huntpilot.csv`
- Look for ”HUNT LIST 1” column; this is a reference to “NAME” in `huntlist.csv` table
- In `huntlist.csv` looks at “LINE GROUP x” columns; reference to “NAME” in linegroup.csv
- Collect DNPs from “DN OR PATTERN x” and “ROUTE PARTITION x” columns in `linegroup.csv`
- Find phones with these DNPs in `phones.csv`; look at “Directory Number X” and “Route Partition X” columns
- Look at “Owner User ID” and “User ID x” columns to collect user IDs
- → Definitely needs to be done programmatically

Example: User Relations based on Hunt Pilots

- Start with `huntpilot.csv`
- Look for "HUNT LIST 1" column; this is a reference to "NAME" in `huntlist.csv` table
- In `huntlist.csv` looks at "LINE GROUP x" columns; reference to "NAME" in linegroup.csv
- Collect DNPs from "DN OR PATTERN x" and "ROUTE PARTITION x" columns in `linegroup.csv`
- F nd "F
- L
- → Definitely needs to be done programmatically

Alternative:
- Parse DN and partition from "PRIMARY EXTENSION" in `enduser.csv`
- And match these with DNPs collected from `linegroup.csv`
- .. If primary extension is populated for all users

Real life examples: Hunt Pilots

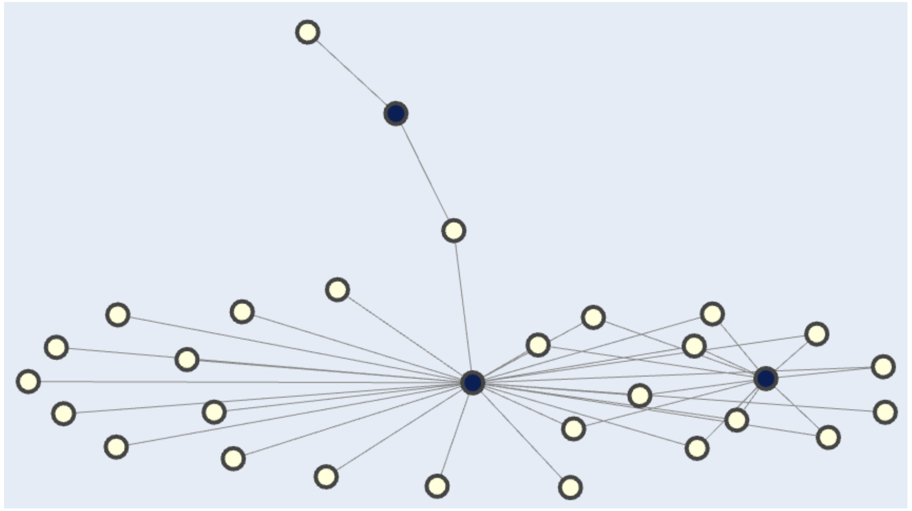


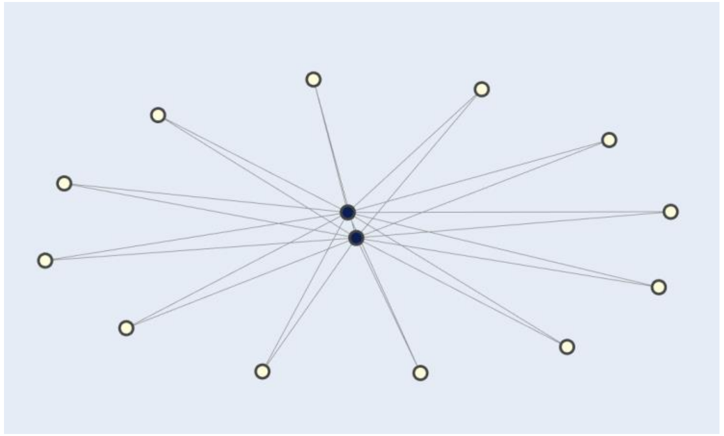
Simple

○ User

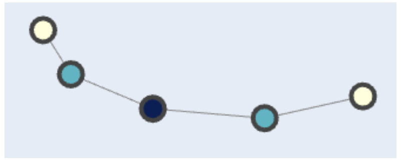
● Hunt Pilot

Complex





Real life examples: Shared Lines

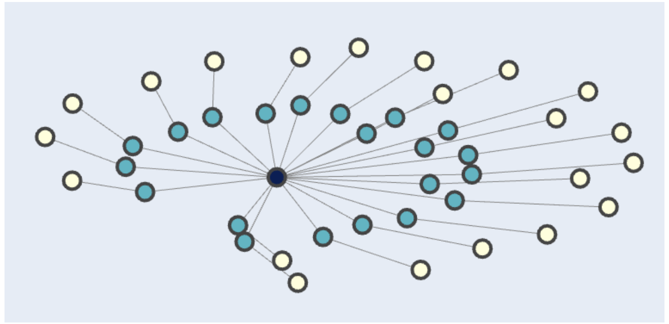


DN shared on two phones owned by two users

○ User

● DN

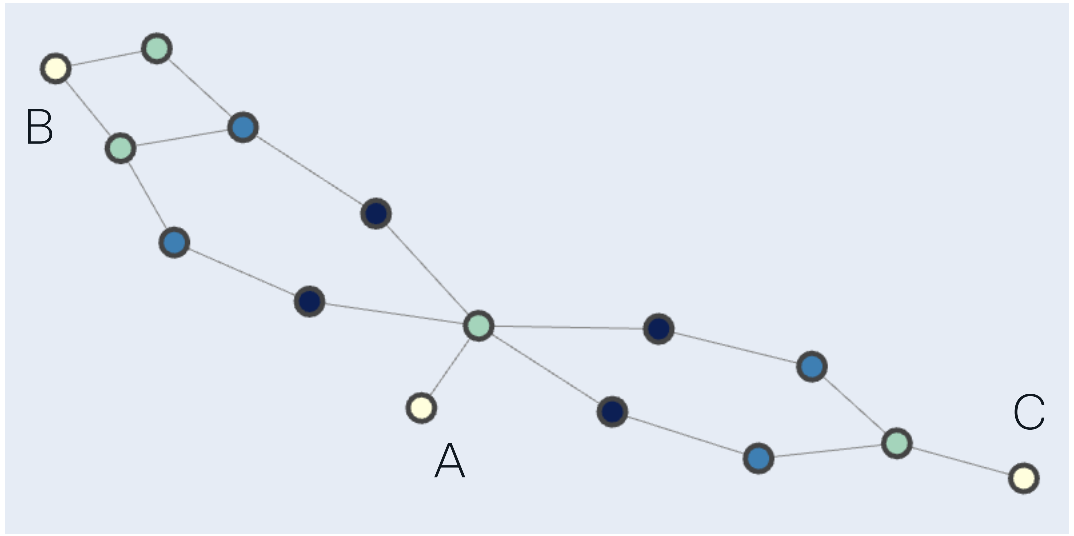
● Phone



Single DN shared on multiple phones

Both cases are problematic for a migration to WxC

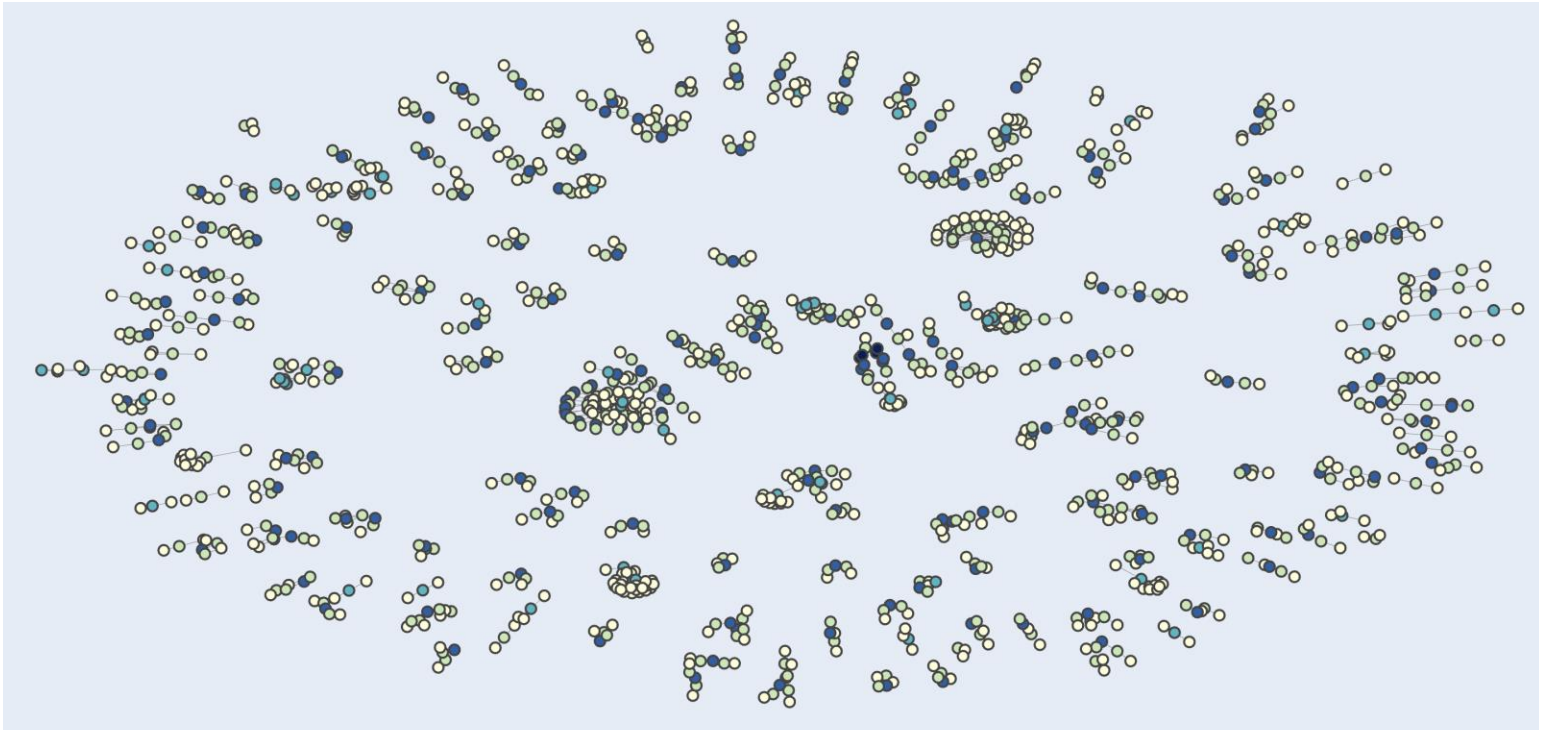
Real life examples: BLF



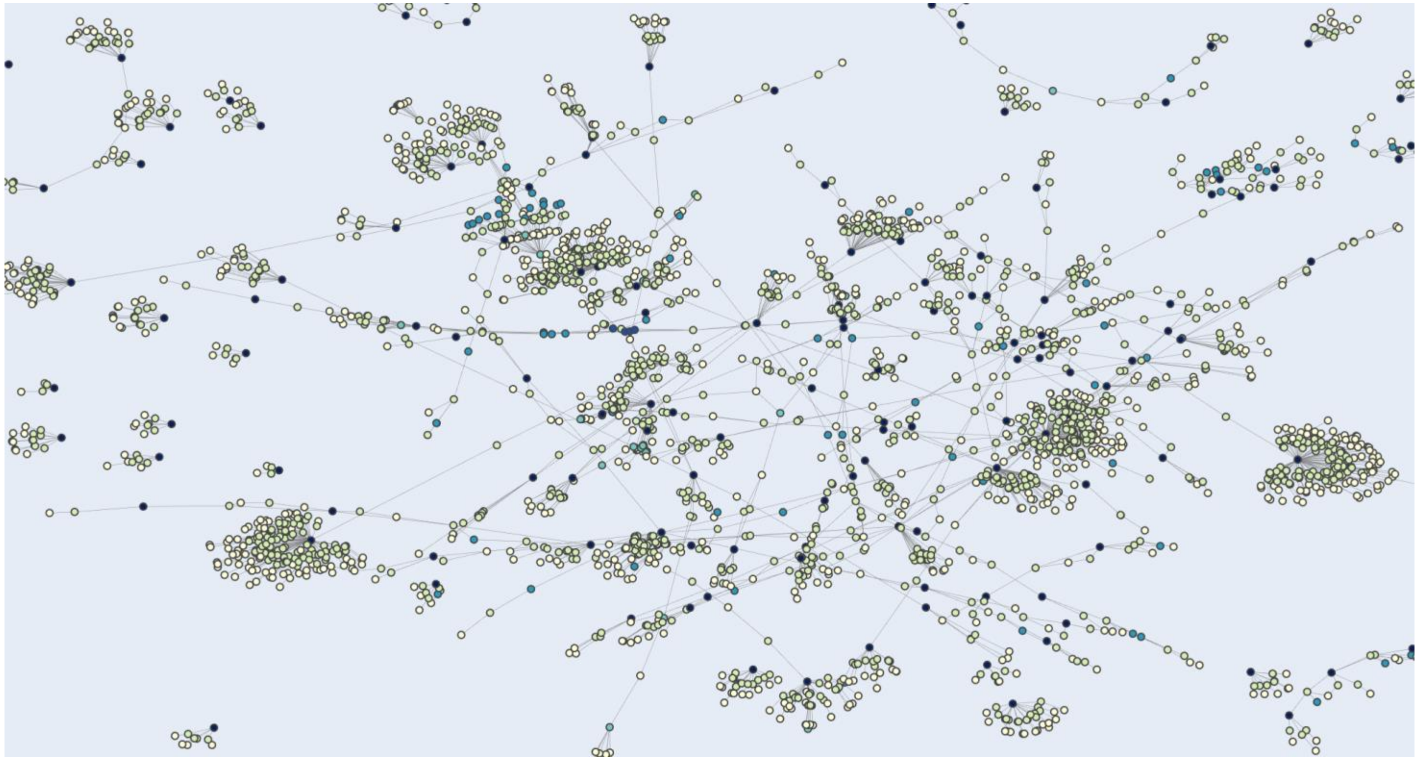
- ○ User
- ● Phone
- ● DN
- ● BLF

User A on one phone has four BLFs monitoring DNs on phones of two users B and C

Real life example: combined

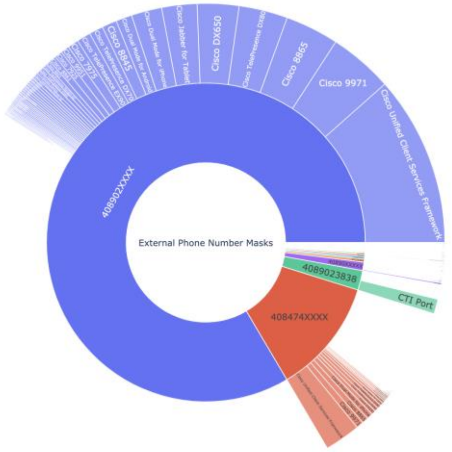


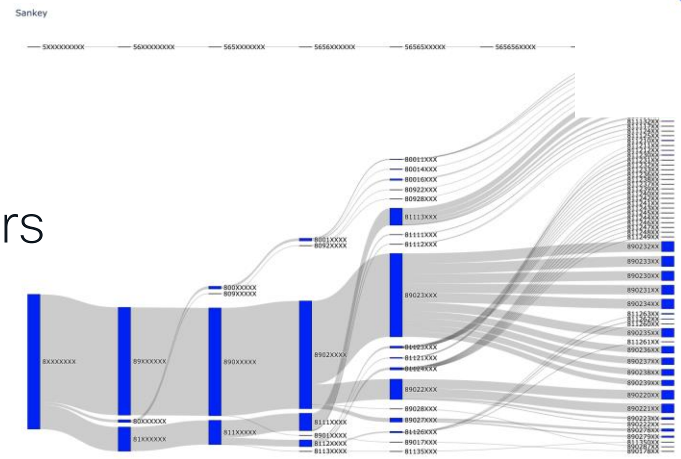
… Or something … “interesting”



Migration Batches … and other analysis

- https://github.com/jeokrohn/ucmmigration
- Uses Python 3.9
- Read and analyse Unified CM data exports
- Work in progress…
- Use as is
- … playground for developers





Discovery Options

	AXL	BAT	Config Export	Yarnlab wrangler_
Coverage	++ All feature can be read using think or thick AXL	-- Very little coverage (built for provisioning)	++ Full DB coverage	+
Ease of use	-- Programmatic approach requires some development experience	+ UCM admin GUI	+ UCM admin GUI	++
Flexibility	++	-- Limited coverage	++	++
End-to-end (incl. provisioning)	O Not integrated. Data can drive Webex Calling provisioning (manual, bulk, or API)	-- Limited coverage	O Not integrated. Data can drive Webex Calling provisioning (manual, bulk, or API)	++ Integrated mapping and provisioning

Design

Challenge: DN vs User Centric Model

- Webex Calling
  - users with extension or TN (or both) have devices
  - Concept of shared lines fundamentally different to Unified CM
  - Users ”live” in a calling location
- What about Unified CM?

Challenge: DNP vs User Centric Model

- Unified CM
  - DNPs (DN, partition) associated with devices, potentially owned by users, which might have a primary extension, ..
  - DNs can exist on an arbitrary number of devices in varying order
  - What is a user's extension?
  - What is a location?
    - Common extension range
    - common +E.164 prefix (how long, what about extension only)
    - same device pool
    - shared CoS (how to you find users sharing equivalent CoS?)
    - …

Challenge: DN vs User Centric Model

- Control Hub based migration tool has heuristics to try to address this challenge
- Admin can still override the tool
- Using the tool during the design phase (w/o actually executing a migration!) can assist in identifying characteristics of Unified CM setup

Feature Mapping

- Calling is not about features, but about business requirements
  - Different call control solutions have different sets of features
  - 1:1 feature mapping not necessarily the best option
- Different ways to address the same set of business requirements

Example: Hunt Group

- Unified CM
  - Hunt Pilot: DNP, Alerting Name
    - Call treatment: FwdNoAnswer, FwdHuntBusy, Queuing (MoH, overflow, max wait time, no agent available)
  - Hunt List: list of hunt groups
  - Line Group: distribution algorithm, RNA, hunt options (no answer, busy, unavailable), members (DNP)

- Webex Calling
  - Basics: location, name, number/extension, caller ID
  - Routing: circular, longest idle, weighted, simultaneous
  - Routing settings: advance when busy, forward after set number of rings, divert when unreachable
  - Agents

- DNP vs user based
  - Unified CM has DNPs in line groups
  - Webex Calling: user or workspaces as agents
- Gap: HG login/logout -> Webex Calling agents can set DND though
- OTOH: Webex Calling has way more options for selective call forwarding
- Gap: queueing .. But then there are Webex Calling call queues
  - Also allow agents to set their state to available/unavailable (for all queues though)

Dialing Habits

Unified CM (typical, best practice)

- Extensions (2-6 digits)
- Abbreviated inter-site (ESN)
- +E.164
- Country specific (PSTN) dialing habits
- .. and potentially countless variations based Unified CM "magic toolbox" → all bets are off

Webex Calling

- Extensions (2-6 digits)
- Abbreviated inter-site (ESN)
- +E.164
- Country specific (PSTN) dialing habits

- Unified CM dial plan flexibility: curse and blessing
  - Often used as workaround to address specific requirements
- Set of dialing habits in Webex Calling is fixed
- Case by case conversation
- Changing dialing habits → changing UX

Class of Service

Unified CM

- Based on CSSes, partitions and patterns
- Common: on-net, national, international
- Potentially: internal calling restrictions, Chinese walls, C-level fences
- Unified CM “magic toolbox” at its best→ all bets are off

Webex Calling

- outgoing permissions (internal, local, LD international…) - based on tags in national dial plan
- Executive/Executive Assistant
- Per user: selectively accept/reject/forward calls

- Different concepts
- Keep in mind: using Unified CM “dial plan magic” to address certain requirements often has been seen as a “workaround”
- If all you have is a hammer (CSS) then everything looks like a nail (set of patterns)
- Different toolbox → different solutions

Shared Lines

Unified CM

- Phone can have multiple DNPs
- Not necessarily tied to user
- DNPs can be in different sites
- Shared line appearances ring at the same time

Webex Calling

- User’s lines can be shared (35 devices max)
- Limited to a single location
- For inbound calling: hunt groups and call queues might be the better option
- Single number reach allows to ring multiple destinations
- Executive assistant feature might address some use cases (exec and assistant can be in different locations)
- Virtual lines offer a lot of flexibility

- Different concepts
- Same location limitation used to be the biggest challenge; can be addressed using virtual lines

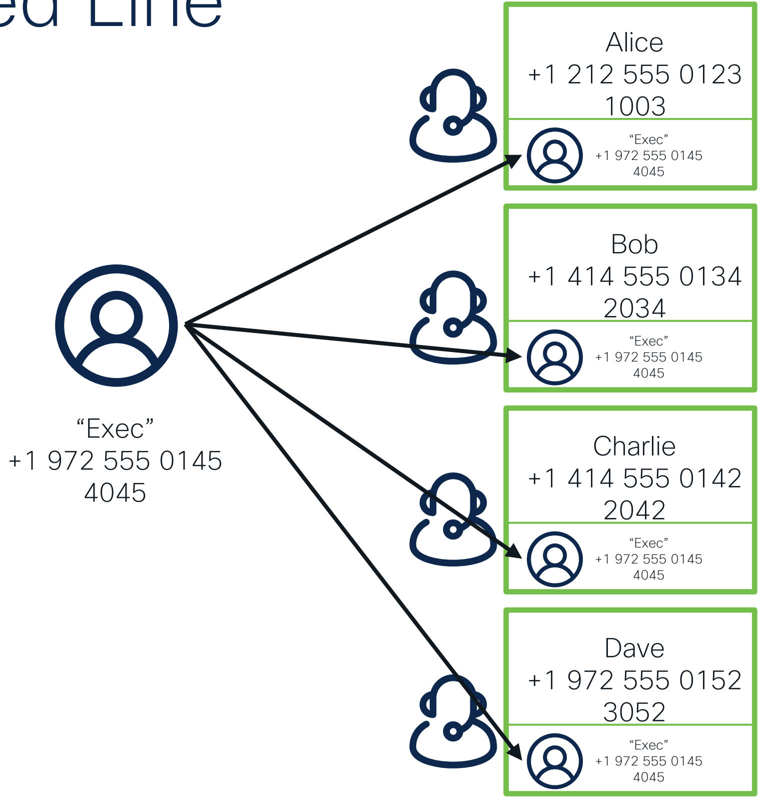
”Executive” as Flexible Shared Line

- Create ”dummy” exec

“Exec”
+1 972 555 0145
4045

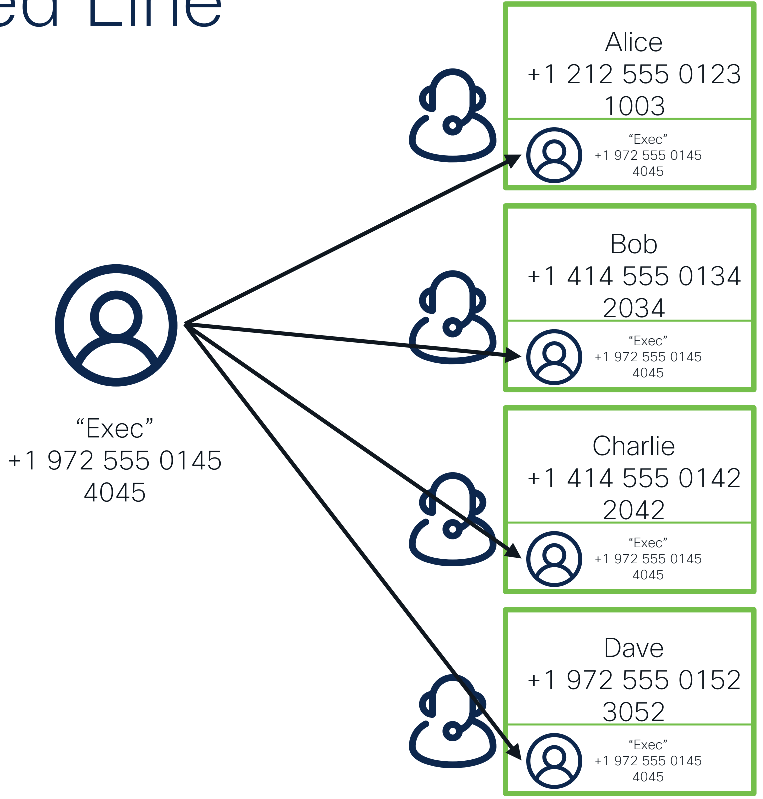
”Executive” as Flexible Shared Line

- Create ”dummy” exec
- .. and assign a bunch of assistants

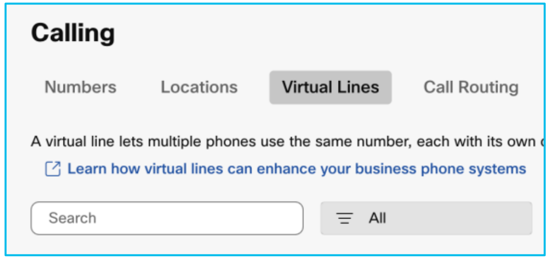


”Executive” as Flexible Shared Line

- Create ”dummy” exec
- .. and assign a bunch of assistants
- Assistants can place calls on behalf of Executive
- Assistants get notification for incoming calls to Executive → can answer calls
- Exec and assistants don’t need to be in same location



Virtual Lines



Calling

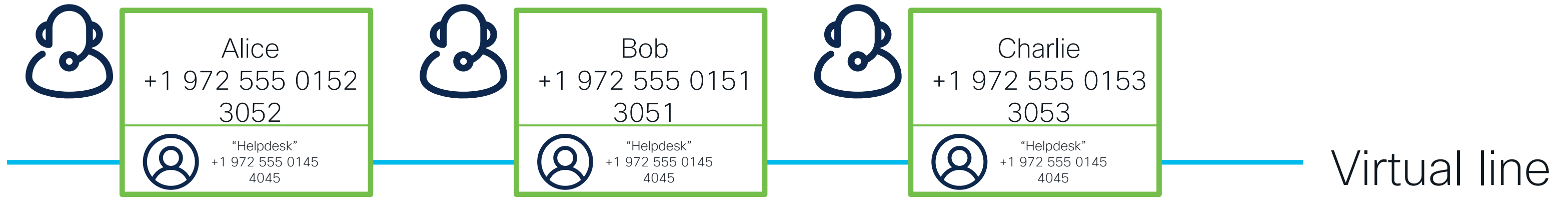
A virtual line lets multiple phones use the same number, each with its own

Learn how virtual lines can enhance your business phone systems

- Virtual Lines act like “dummy” users
  - Attributes: first/last/display name, TN and/or extension
  - Live in a location
  - Settings: caller ID, ECBN, incoming/outgoing permissions, intercept, barge, ...
  - ... like calling settings of a user
- Virtual lines can be added to MPPs and app instances as additional lines
- Can be used for incoming/outgoing calls
- No location restriction: MPP and app can have lines from multiple locations

Virtial Lines vs Shared Lines

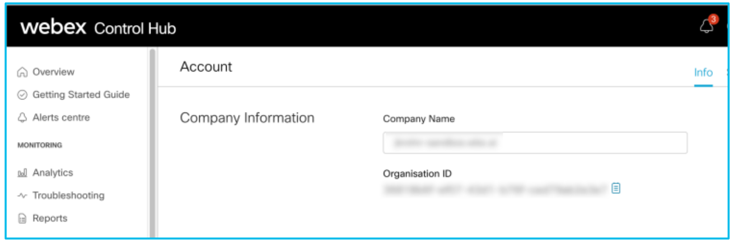
- Virtual lines don't remove the user centric concept of Webex Calling
- … but remove a lot of the original restrictions
- For example: sharing a helpdesk line across a group of users now is really easy



Deploy

Webex Org

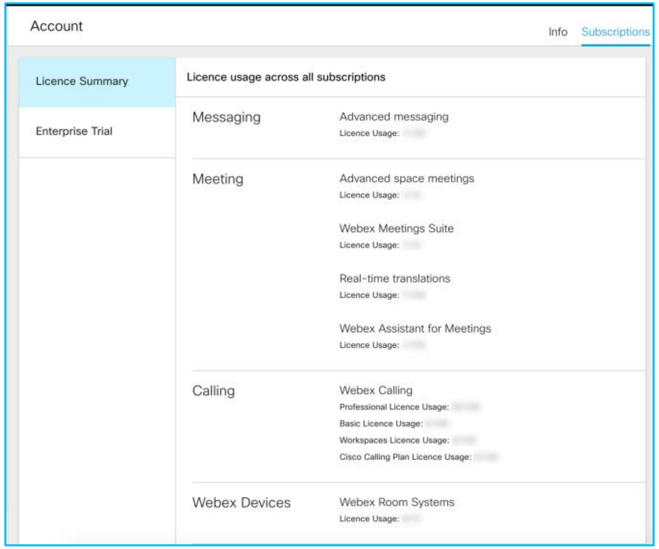
- Create production Webex Org (or verify)
- Check licenses (add if required)
  - Make sure that required licenses are available
  - Requires prior assessment … and potentially re-assessment during discovery phase
  - Check expiration (if starting as trial)



Company Information

Company Name: [redacted]

Organisation ID: [redacted]



Licence usage across all subscriptions

Messaging	Advanced messaging Licence Usage: [redacted]
Meeting	Advanced space meetings Licence Usage: [redacted] Webex Meetings Suite Licence Usage: [redacted] Real-time translations Licence Usage: [redacted] Webex Assistant for Meetings Licence Usage: [redacted]
Calling	Webex Calling Professional Licence Usage: [redacted] Basic Licence Usage: [redacted] Workspaces Licence Usage: [redacted] Cisco Calling Plan Licence Usage: [redacted]
Webex Devices	Webex Room Systems Licence Usage: [redacted]

Setup prior to Calling Migration

- Domain verification/claim
- License templates
- User provisioning
  - Directory Connector, SCIM, CCUC, CSV, manual, API
- SSO

User Migration or Provisioning

User Migration/Provisioning Options

- Manual or CSV bulk operation
  - Not really scalable
  - Risk of inconsistencies
- Cloud Connected UC
  - Migration of batches
- APIs
  - Foundation for custom integrations
- Directory synchronization
  - Okta, Azure, AD
  - Foundation for SSO

Best practice

Foundation: Identity

- Concept of “Common Identity”: same identity within the enterprise and for cloud services
- Synchronization of enterprise and cloud identity
- Benefits:
  - User Experience: users can use same identity (and credentials w/ SSO) for authentication
  - Operational Efficiency: minimized management overhead
- Foundation for all Webex services

User Provisioning for Cloud Services

- Enterprises typically maintain user information in an enterprise directory
- Cisco Webex maintains common identity storage for user information for all cloud services
- Requirements:
  - Consistent user information in enterprise directory and cloud identity storage
  - avoid additional maintenance effort for system administrator
  - Automatic create, update, and delete of users
- Solution: directory sync

User Provisioning Options

	AD sync	Sync from Okta or Azure AD	Manual provisioning	Bulk Provisioning (CSV)	People API
Moves, Adds, Changes	++ automatic	++ automatic	-- manual	- CSV prep	o / + / ++ Depends on level of integration
Ease of use	+ Initial setup required	+ Initial setup required	++ No setup, public documentation	o Process setup (data source, data format, ..)	- Steep learning curve, development required for integration
Infrastructure requirements	Directory connector	None	None	None	Hosting if using web app
Flexibility	o some customization possible (groups, attribute mapping, ..)*	o some customization possible (groups, attribute mapping, ..)*	++	+	++

*Some user attributes (e.g. mobile number, department, manager, title) can only be set via directory sync

https://help.webex.com/en-us/article/nj34yk2

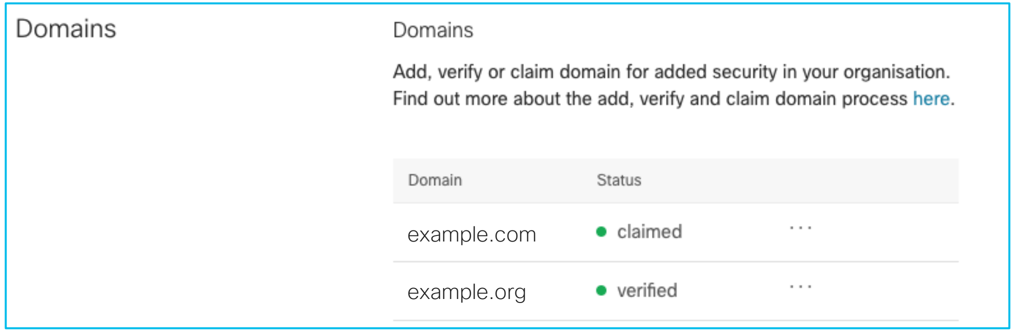
Converting Users

- Admin can convert users belonging to other orgs (including free org) to org users
- Based on email address domain
  - Requires email domain verification (or claim)
- Immediate or delayed claim
- Claim only possible if directory sync is not enabled
- License assignments checked/updated as part of the conversion process

https://help.webex.com/en-us/article/nceb8tm/Claim-users-to-your-organization-("convert"-users)
https://help.webex.com/en-us/article/e4ektc/Disable-Delayed-Claim-for-Your-Organization

“Claimed” and “Verified” Domains

- Control hub allows to “claim” and “verify” domains
- Domains can be verified via Control Hub (DNS validation via TXT record)



Domains

Domains

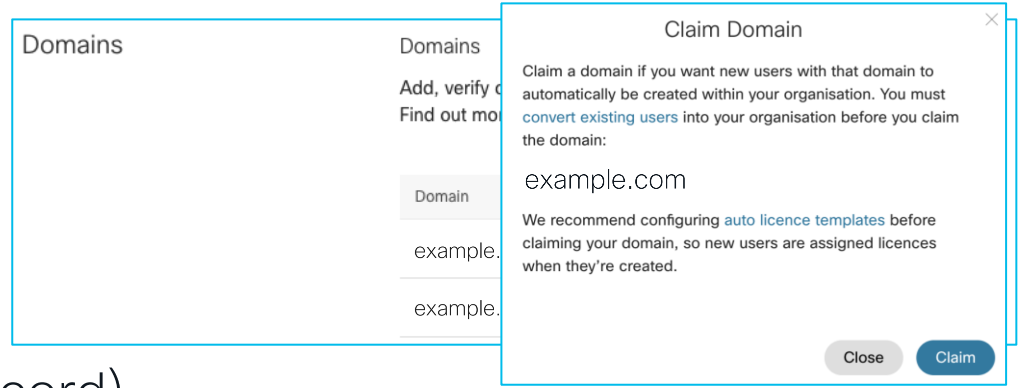
Add, verify or claim domain for added security in your organisation.
Find out more about the add, verify and claim domain process here.

Domain	Status
example.com	● claimed
example.org	● verified

https://help.webex.com/en-us/article/cd6d84/Manage-your-domains

“Claimed” and “Verified” Domains

- Control hub allows to “claim” and “verify” domains
- Domains can be verified via Control Hub (DNS validation via TXT record)
- Domain claim requires that domain has been verified before


Domains

Domains

Add, verify c
Find out mo

Domain
example.
example.

Claim Domain

Claim a domain if you want new users with that domain to automatically be created within your organisation. You must convert existing users into your organisation before you claim the domain:

example.com

We recommend configuring auto licence templates before claiming your domain, so new users are assigned licences when they're created.

https://help.webex.com/en-us/article/cd6d84/Manage-your-domains

”Claimed” and “Verified” Domains

- Verified domain
  - Users w/ email addresses from verified domains can be converted to licensed users from consumer organization
  - To avoid “pending” users domain (at least) needs to be verified
- Claimed domain
  - New users with email addresses w/ that domain can only be added to organization for which the domain has been claimed
  - Users existing before claim are not affected
  - Make sure to convert* existing users
  - Automatic user activation requires claimed domain (and SSO)

*https://help.webex.com/en-us/article/nceb8tm/Claim-users-to-your-organization-(%22convert%22-users)

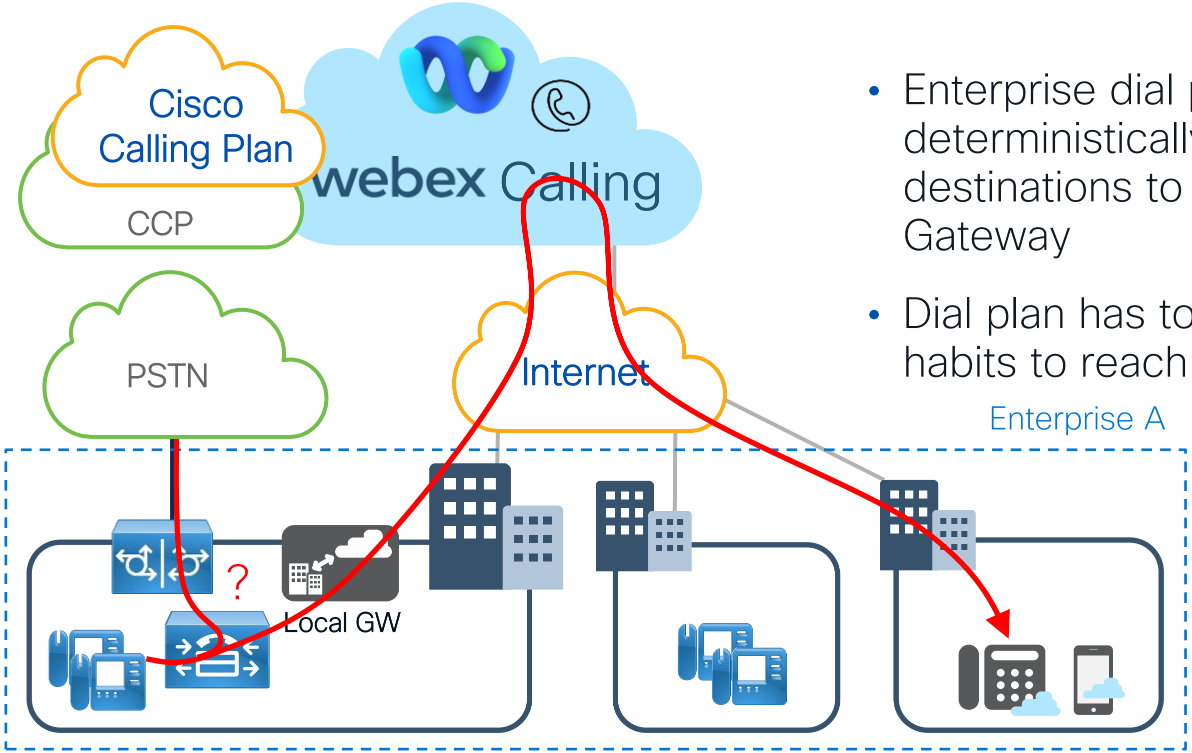
“Claimed” vs “Verified”

	Verified Domain	Claimed Domain
Process	Control Hub, DNS based validation (TXT record)	Verify 1st, then claim
Exclusive	Domain users can exist in and can be added to other organizations	Domain users can not be added to other organizations Domain users existing prior to claim are not affected
Sideboarding	Domain users can be sideboarded into consumer organization	into customer organization, can be disabled* w/ Directory Connector: no sideboarding!
Conversion	Domain users can be converted from consumer organization Delayed conversion if email domain is not verified nor claimed	
Directory Connector	Can add users from domain	Can add users from domain

*https://help.webex.com/en-us/article/nfiu0ed/Prevent-Users-from-Self-Registering-with-your-Domain

Interworking Unified CM / Webex Calling

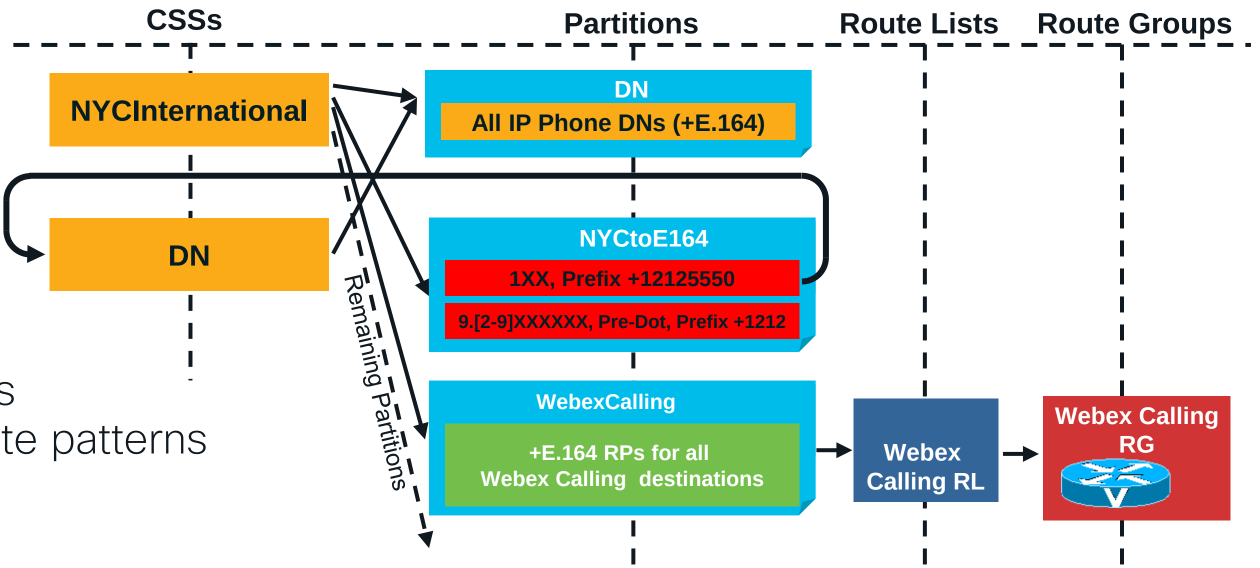
Local Gateway Dial Plan Integration



- Enterprise dial plan on Unified CM needs to deterministically send Webex Calling destinations to Webex Calling via Local Gateway
- Dial plan has to support "typical" dialling habits to reach Webex Calling destinations
- Webex Calling destinations need to be regularly updated during transition period as users move to Webex Calling

LGW Dial Plan Integration

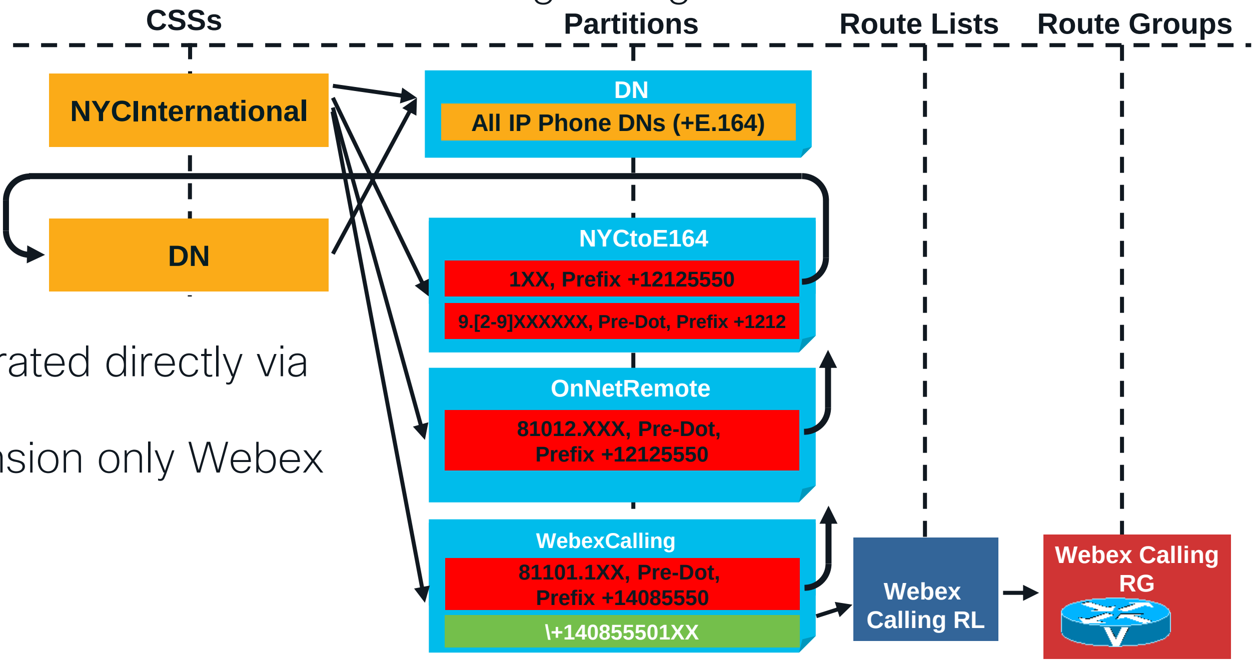
- With a single Egress LGW selection not site specific → no LRG based LGW selection required*
- Multiple LGWs in multiple locations: RL and LRG
- Can can use multiple LGWs for scale and redundancy
- Webex Calling destinations provisioned as +E.164 route patterns



*For extension dialling w/ non-unique extensions site specific trunks are required so that Webex Calling can establish proper dialling context. LRG can be used for site specific egress trunk selection

Abbreviated Inter-site Dialing to Webex Calling

- Any dialing habit can be transformed to +E.164 using dialing normalization translations
- Same partition as used for Webex Calling routes
- Webex calling dialing habits can also be integrated directly via route patterns
  → also works with extension only Webex Calling devices



Dial Plan Maintenance

- Depending on the number of Webex Calling TN ranges maintaining the set of Webex Calling RPs can be complex
- Alternative: use GDPR imported catalog with set of Webex Calling TN ranges
- SIP route pattern for catalog's route string needs to be provisioned in the `WebexCalling` partition
- BAT File format (example):

```
PatternType,PSTNFailover,Pattern
pattern,2:+0,+1408555012X@example.com
pattern,2:+0,+1212555013X@example.com
pattern,0:+0,811011XX@example.com
```

- GDPR PSTN failover needs to be suppressed (illegal numbers) if PSTN access for Webex Calling destinations is via UCM; else: loops!

Dial Plan Maintenance

- Depending on the number of Webex Calling TN ranges maintaining the set of Webex Calling RPs can be complex
- Alternative: use GDPR imported catalog with set of Webex Calling TN ranges
- SIP route pattern for catalog's route string needs to be provisioned in the `WebexCalling` partition
- BAT File format (example):

Forcing illegal number for GDPR PSTN failover (strip/prefix). Make sure that dial plan blocks \+0! or use prefix not covered by PSTN route patterns*

```
PatternType,PSTNFailover,Pattern
pattern,2:+0,+1408555012X@example.com
pattern,2:+0,+1212555013X@example.com
pattern,0:+0,811011XX@example.com
```

+E.164 destination in Webex Calling

Abbreviated inter-site dialing to Webex Calling

- GDPR PSTN failover needs to be suppressed (illegal numbers) if PSTN access for Webex Calling destinations is via UCM; else: loops!

*If Webex Calling is not using premises PSTN then GDPR PSTN failover actually is an alternative to using the Local Gateway trunk)

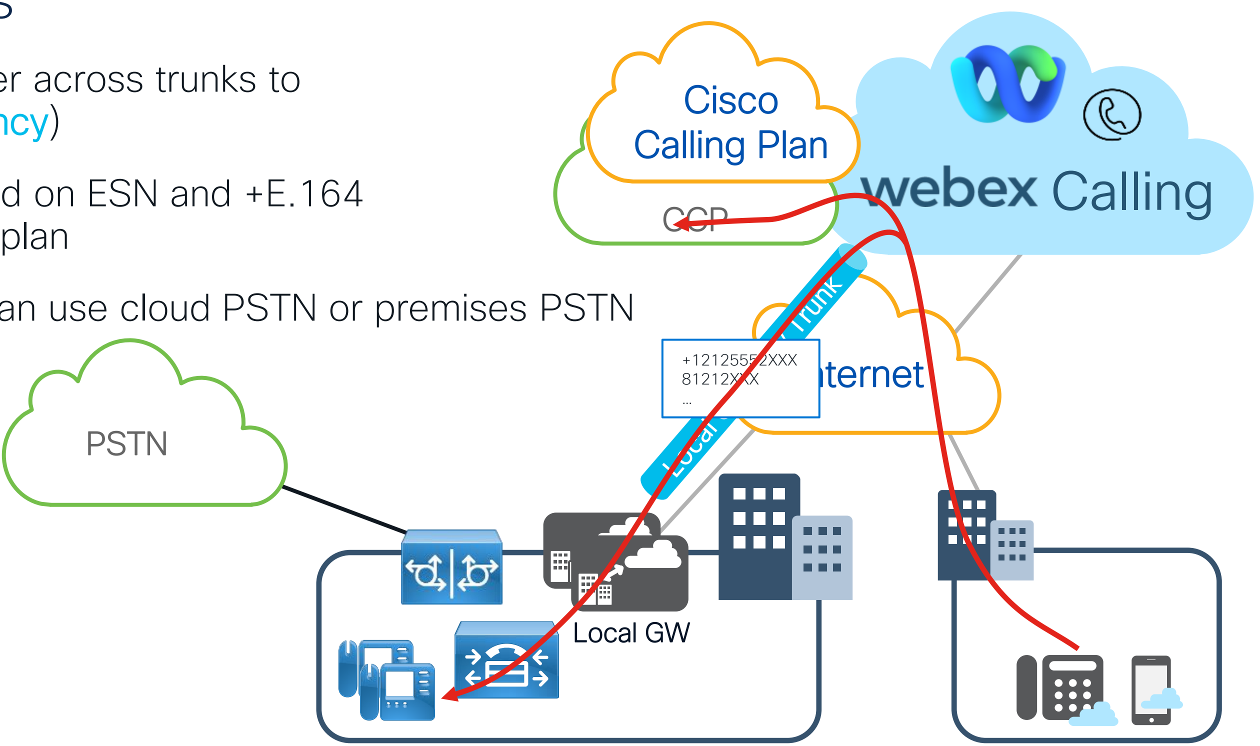
GDPR Imported Catalog Considerations

- Allows to share Webex Calling destinations between clusters: for example for SME deployments with centralized LGW
- Calls coming into UCM from Webex Calling need access to destinations learned from ILS/GDPR (access to remote on-net sites)
- Imported +E.164 and ESN patterns end up in the same partition (for example `OnNetRemote`)
- Breaking the loop: SIP route pattern for catalog’s route string is in partition the trunk from Webex Calling does not have access to

Routing from Webex Calling to Unified CM

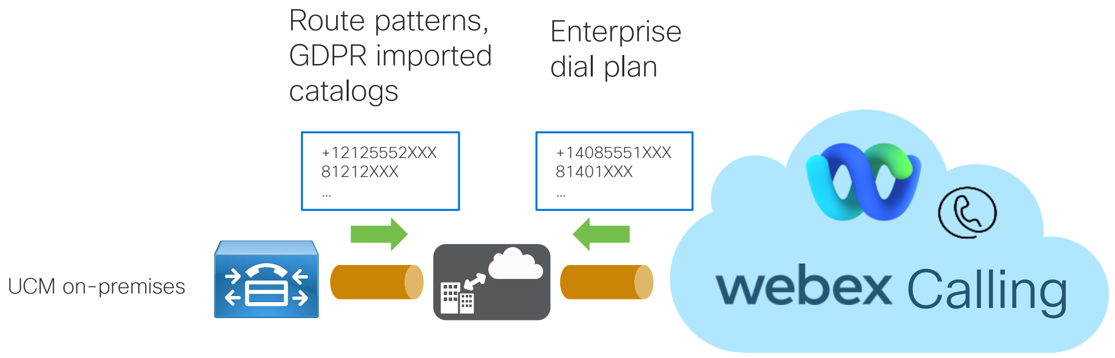
Enterprise Dial Plans

- Load balancing and failover across trunks to premises (scale, redundancy)
- Deterministic routing based on ESN and +E.164 patterns in enterprise dial plan
- Webex Calling locations can use cloud PSTN or premises PSTN
- Porting numbers from premises PSTN to cloud can happen as users move or at the end of the migration



Interworking: Webex Calling and Unified CM

- Interworking between Webex Calling and Unified CM requires
  - Trunk, Local Gateway
  - Dial plan configuration
  - .. Both on Unified CM and on Webex Calling



Cisco UCM and Webex Calling coexistence

- Proper dial plan design (see Enterprise PA* for details) enables seamless transition of DNs from UCM to Webex Calling
  - All dialing habits are possible: ESN, DN and +E.164
- Detailed information in the “Transitioning from Cisco UCM to Webex Calling Deployment Guide”: https://www.cisco.com/c/dam/en/us/td/docs/solutions/PA/mcp/DEPLOYMENT_CALLING_Unified_CM_to_Webex_Calling.pdf

*https://www.cisco.com/c/en/us/td/docs/solutions/CVD/Collaboration/enterprise/12x/120/collbcvd.html

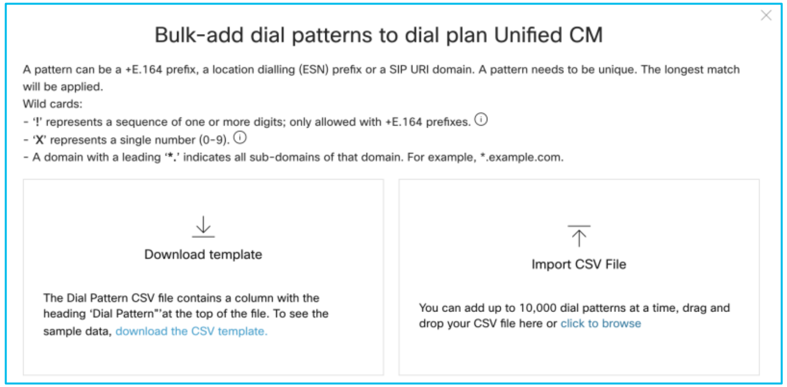
GDPR Export to Populate Dial Plans

Use GDPR Information for Webex Calling Dial Plans

- Unified CM uses ILS/GDPR to exchange catalogs of routing information
- This information is stored in the `remoteroutingpattern` table in Unified CM
- We can export the learned patterns and re-use them for dial plan provisioning
- Can only be used in multi-cluster deployments

Problems to Solve

- Export from Unified CM
  - Use thin AXL to read database directly
  - Need to read from multiple UCMs
- Transform
  - Only wildcard in Webex Calling dial plan patterns is “X” at the end
- Import into Webex Calling
  - Webex Calling dial plans patterns can be provisioned using CSV



Bulk-add dial patterns to dial plan Unified CM

A pattern can be a +E.164 prefix, a location dialling (ESN) prefix or a SIP URI domain. A pattern needs to be unique. The longest match will be applied.
Wild cards:
- '!' represents a sequence of one or more digits; only allowed with +E.164 prefixes.
- 'X' represents a single number (0-9).
- A domain with a leading '*.' indicates all sub-domains of that domain. For example, *.example.com.

Download template

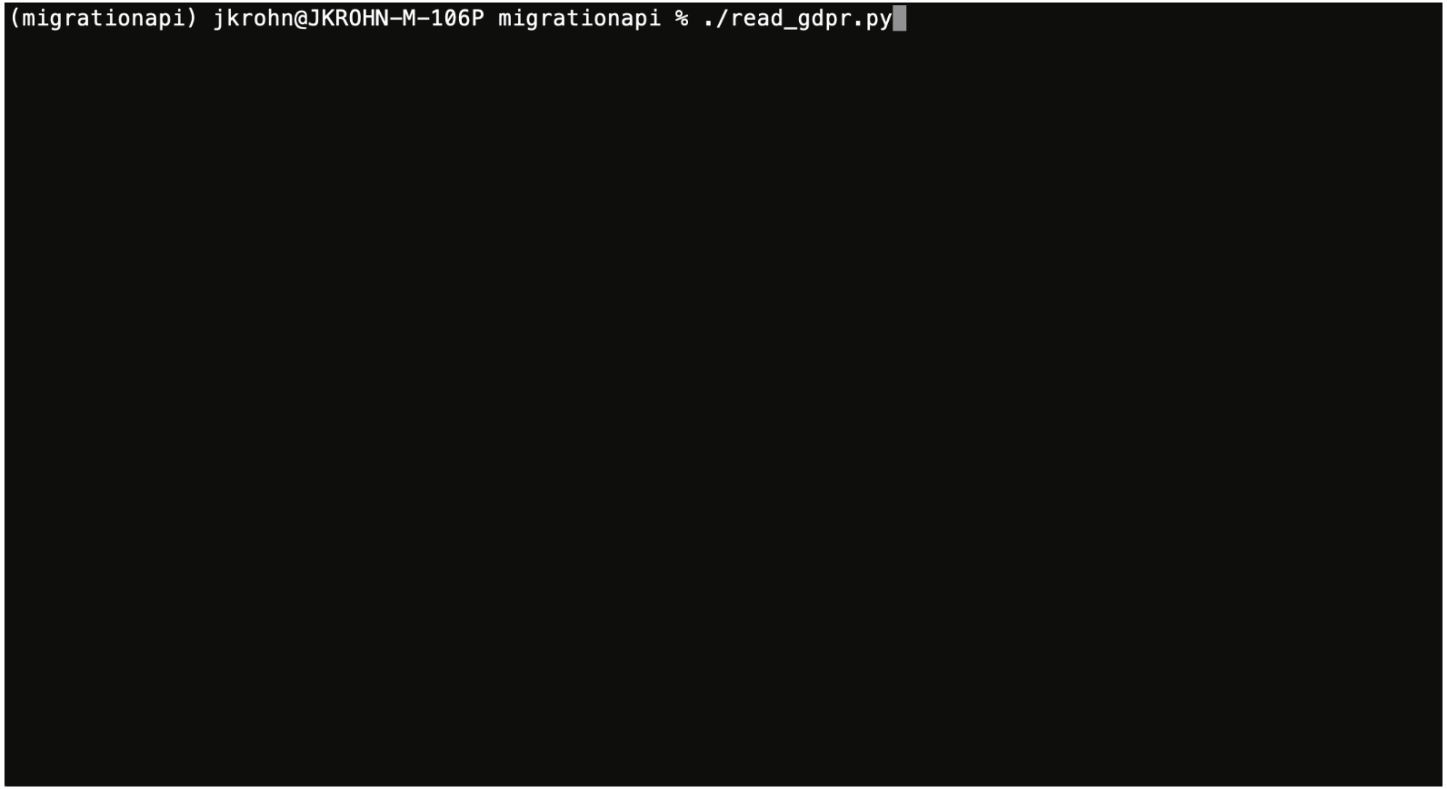
The Dial Pattern CSV file contains a column with the heading 'Dial Pattern"'at the top of the file. To see the sample data, download the CSV template.

Import CSV File

You can add up to 10,000 dial patterns at a time, drag and drop your CSV file here or click to browse

https://github.com/jeokrohn/migrationapi
File: read_gdpr.py

Demo: GDPR Export



```
(migrationapi) jkrohn@JKROHN-M-106P migrationapi % ./read_gdpr.py
```

Migrate

Migrate UCM to Webex Calling: Tools

CCUC-powered

I want to explore Webex…

User + Org Contact Migration

Provision End Users and Org-level Contacts in Webex Identity and Contacts Service.

- UCM BAT CSV to Control Hub (enduser data)
- Control Hub – new Bulk Admin for Org Contacts
- (beta) CCUC agent sync to Webex backend

CCUC-powered

I'm ready to use the Webex App…

Jabber to WxApp Migration

Stay on-premises UCM, migrate Jabber's messaging, meetings to the cloud.

- Control Hub - basic Jabber deployment insights
- UCM BAT CSV to Control Hub (Jabber config)

I want to personalize my Webex app…

Personal Contact Migration

Make personal contacts from Jabber available in Webex App.

- IMP BAT CSV to Control Hub (custom contacts)

I'm ready to move to Webex Calling!

UCM to WxCalling Migration

Entitle users, devices, locations and phone numbers from UCM in Webex Calling.

- UCM BAT CSV to Control Hub (UCM config)

I want to use my Cisco device for Webex Calling…

Phone Firmware Migration

Migrate Cisco IP phone enterprise firmware to the Webex Calling-ready MPP firmware.

- Simplified experience via Control Hub

I want to configure detailed Webex Calling services…

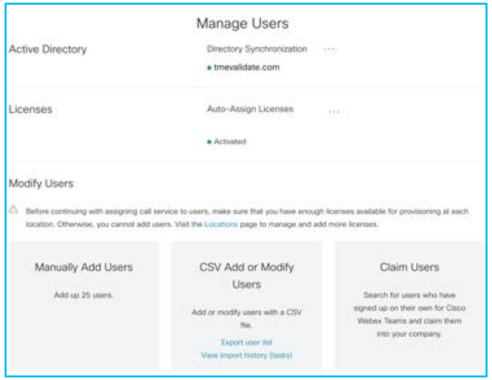
APIs, CSV, Bulk Admin

Take advantage of multiple tools available to customize your Webex experience.

- Webex Calling – new Provisioning APIs
- Control Hub – new Bulk Admin CSV options

Tools Supporting Migrations

- Control Hub migration tools: users, numbers, devices
- Webex API support
- Batch operations



Manage Users

Active Directory — Directory Synchronization ··· ● tmevalidate.com

Licenses — Auto-Assign Licenses ··· ● Activated

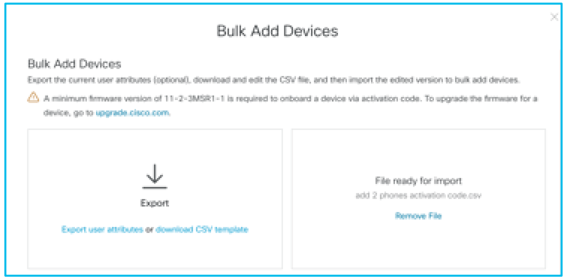
Modify Users

⚠ Before continuing with assigning call service to users, make sure that you have enough licenses available for provisioning at each location. Otherwise, you cannot add users. Visit the Locations page to manage and add more licenses.

Manually Add Users
Add up 25 users.

CSV Add or Modify Users
Add or modify users with a CSV file.
Export user list
View import history (tasks)

Claim Users
Search for users who have signed up on their own for Cisco Webex Teams and claim them into your company.



Bulk Add Devices

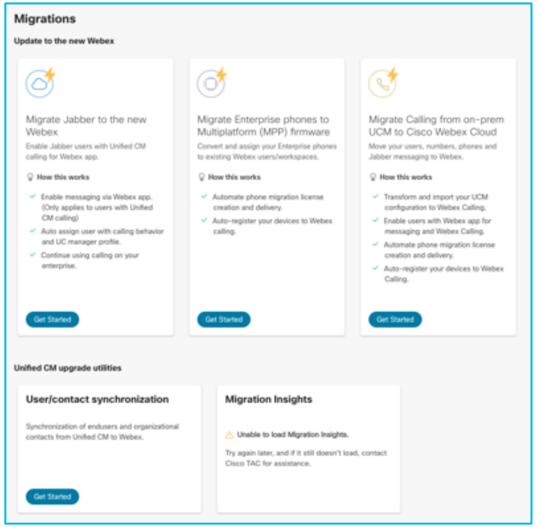
Bulk Add Devices

Export the current user attributes (optional), download and edit the CSV file, and then import the edited version to bulk add devices.

⚠ A minimum firmware version of 11-2-3MSR1-1 is required to onboard a device via activation code. To upgrade the firmware for a device, go to upgrade.cisco.com.

Export
Export user attributes or download CSV template

File ready for import
add 2 phones activation code.csv
Remove File



Migrations

Update to the new Webex

Migrate Jabber to the new Webex
Enable Jabber users with Unified CM calling for Webex app.
How this works
- ✓ Enable messaging via Webex app. (Only applies to users with Unified CM calling)
- ✓ Auto assign user with calling behavior and UC manager profile.
- ✓ Continue using calling on your enterprise.

Get Started

Migrate Enterprise phones to Multiplatform (MPP) firmware
Convert and assign your Enterprise phones to existing Webex users/workspaces.
How this works
- ✓ Automate phone migration license creation and delivery.
- ✓ Auto-register your devices to Webex calling.

Get Started

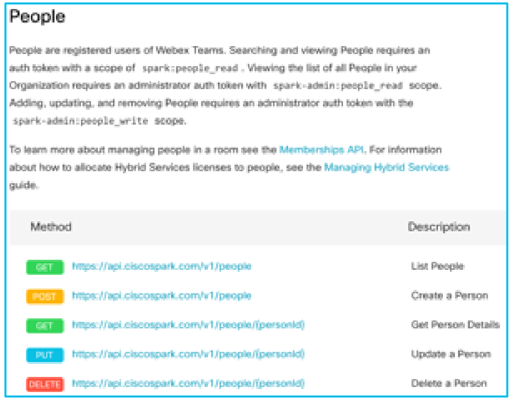
Migrate Calling from on-prem UCM to Cisco Webex Cloud
Move your users, numbers, phones and Jabber messaging to Webex.
How this works
- ✓ Transform and import your UCM configuration to Webex Calling.
- ✓ Enable users with Webex app for messaging and Webex Calling.
- ✓ Automate phone migration license creation and delivery.
- ✓ Auto-register your devices to Webex Calling.

Get Started

Unified CM upgrade utilities

User/contact synchronization
Synchronization of endusers and organizational contacts from Unified CM to Webex.
Get Started

Migration Insights
⚠ Unable to load Migration Insights.
Try again later, and if it still doesn't load, contact Cisco TAC for assistance.

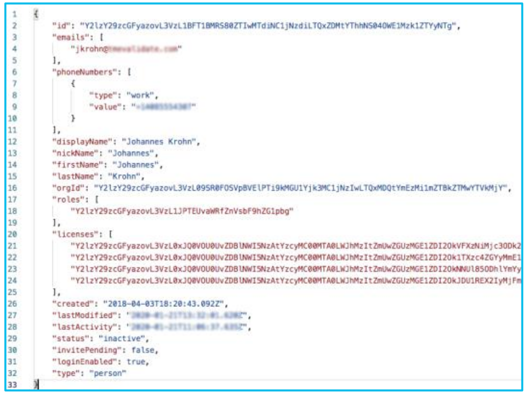


People

People are registered users of Webex Teams. Searching and viewing People requires an auth token with a scope of `spark:people_read`. Viewing the list of all People in your Organization requires an administrator auth token with `spark-admin:people_read` scope. Adding, updating, and removing People requires an administrator auth token with the `spark-admin:people_write` scope.

To learn more about managing people in a room see the Memberships API. For information about how to allocate Hybrid Services licenses to people, see the Managing Hybrid Services guide.

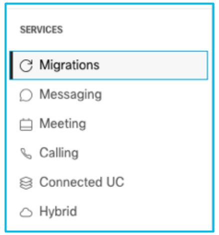
Method		Description
GET	https://api.ciscospark.com/v1/people	List People
POST	https://api.ciscospark.com/v1/people	Create a Person
GET	https://api.ciscospark.com/v1/people/{personId}	Get Person Details
PUT	https://api.ciscospark.com/v1/people/{personId}	Update a Person
DELETE	https://api.ciscospark.com/v1/people/{personId}	Delete a Person

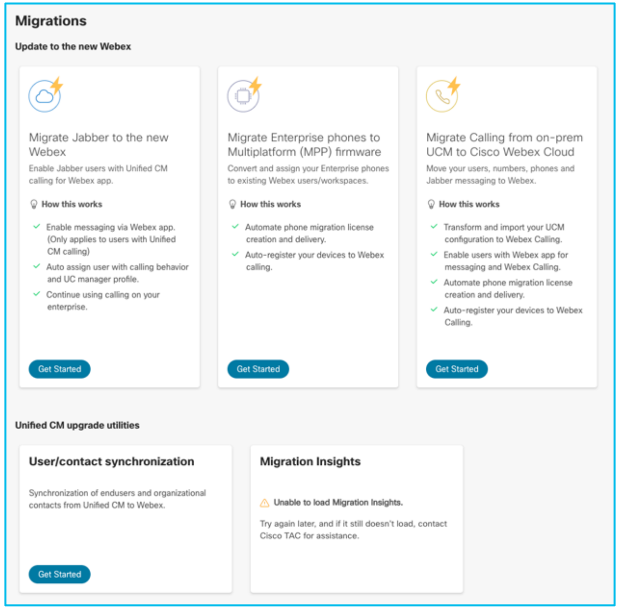


```
{
    "id": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS80ZTIwMTdiNC1jNzdiLTQxZDMtYThhNS04OWE1Mzk1ZTYyNTg",
    "emails": [
        "jkrohn@[redacted]"
    ],
    "phoneNumbers": [
        {
            "type": "work",
            "value": "[redacted]"
        }
    ],
    "displayName": "Johannes Krohn",
    "nickName": "Johannes",
    "firstName": "Johannes",
    "lastName": "Krohn",
    "orgId": "Y2lzY29zcGFyazovL3VzL09SR0FOSVpBVElPTi9kMGU1Yjk3MC1jNzIwLTQxMDQtYmEzMi1mZTBkZTMwYTVkMjY",
    "roles": [
        "Y2lzY29zcGFyazovL3VzL1JPTEUvaWRfZnVsbF9hZG1pbg"
    ],
    "licenses": [
        "Y2lzY29zcGFyazovL3VzL0xJQ0VOU0UvZDBlNWI5NzAtYzcyMC00MTA0LWJhMzItZmUwZGUzMGE1ZDI2OkVFXzNiMjc3ODk2
        "Y2lzY29zcGFyazovL3VzL0xJQ0VOU0UvZDBlNWI5NzAtYzcyMC00MTA0LWJhMzItZmUwZGUzMGE1ZDI2Ok1TXzc4ZGYyMmE1
        "Y2lzY29zcGFyazovL3VzL0xJQ0VOU0UvZDBlNWI5NzAtYzcyMC00MTA0LWJhMzItZmUwZGUzMGE1ZDI2OkNNUl85ODhlYmYy
        "Y2lzY29zcGFyazovL3VzL0xJQ0VOU0UvZDBlNWI5NzAtYzcyMC00MTA0LWJhMzItZmUwZGUzMGE1ZDI2OkJDU1REX2IyMjFm
    ],
    "created": "2018-04-03T18:20:43.092Z",
    "lastModified": "[redacted]",
    "lastActivity": "[redacted]",
    "status": "inactive",
    "invitePending": false,
    "loginEnabled": true,
    "type": "person"
}
```

Calling Migration Tool

- Launched from Control Hub
- Use Unified CM configuration export (TAR)
- Data validation
- Extract/validate numbers, users, device associations
- Identify compatible devices
- Migration in batches
- Direct provisioning; not based on Webex Calling batch provisioning




Migrations

Update to the new Webex

Migrate Jabber to the new Webex
Enable Jabber users with Unified CM calling for Webex app.
How this works
- ✓ Enable messaging via Webex app. (Only applies to users with Unified CM calling)
- ✓ Auto assign user with calling behavior and UC manager profile.
- ✓ Continue using calling on your enterprise.

Get Started

Migrate Enterprise phones to Multiplatform (MPP) firmware
Convert and assign your Enterprise phones to existing Webex users/workspaces.
How this works
- ✓ Automate phone migration license creation and delivery.
- ✓ Auto-register your devices to Webex calling.

Get Started

Migrate Calling from on-prem UCM to Cisco Webex Cloud
Move your users, numbers, phones and Jabber messaging to Webex.
How this works
- ✓ Transform and import your UCM configuration to Webex Calling.
- ✓ Enable users with Webex app for messaging and Webex Calling.
- ✓ Automate phone migration license creation and delivery.
- ✓ Auto-register your devices to Webex Calling.

Get Started

Unified CM upgrade utilities

User/contact synchronization
Synchronization of endusers and organizational contacts from Unified CM to Webex.

Get Started

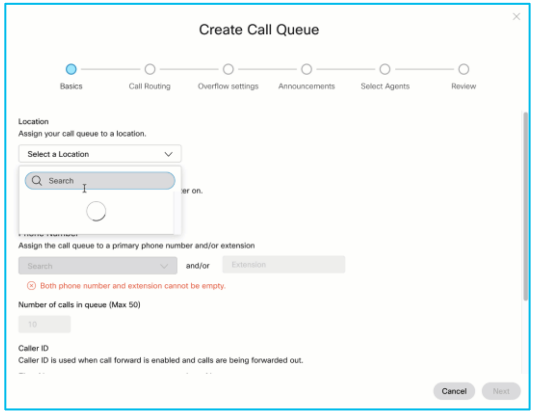
Migration Insights
⚠ Unable to load Migration Insights.
Try again later, and if it still doesn't load, contact Cisco TAC for assistance.

Bulk Operations

- API 1st strategy: build API support 1st, CSV bulk operations follow
- CSV based bulk administration for:
  - Users
  - Devices
  - Call pickups
  - Call queues
  - Hunt groups
- Provisioning of call pickups, call queues, and hunt groups w/o bulk operation can account for a significant portion of time in migration projects
  - # of instances
  - # of parameters

Bulk Provisioning Time Savings

- Each call pickup, call queue, hunt group instance requires populating a wizard with many parameters
- Example: call queue wizard with six pages and dozens of parameters
- Estimate: up to 10 min to create a single hunt group
- Causes redundant work if many instances need to be created with same/similar settings
- Repetitive tasks are likely to cause errors
- CSV support reduces effort for deployment and migrations


Create Call Queue

Location
Assign your call queue to a location.
Select a Location

er on.

Assign the call queue to a primary phone number and/or extension

Both phone number and extension cannot be empty.

Number of calls in queue (Max 50)
10

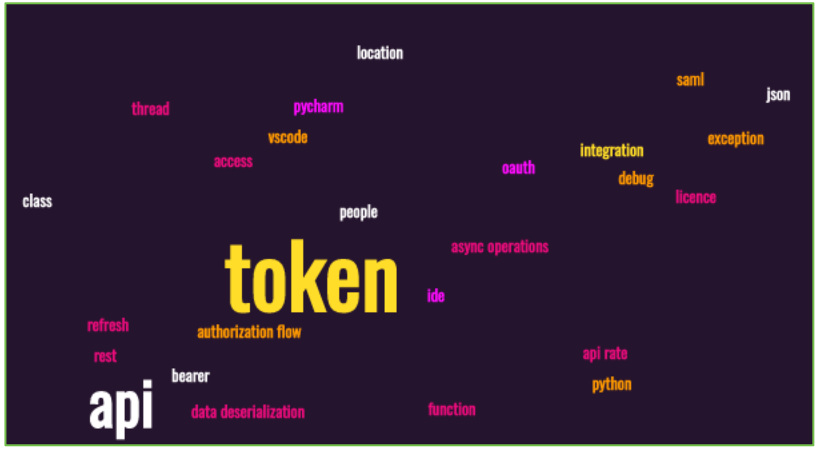
Caller ID
Caller ID is used when call forward is enabled and calls are being forwarded out.

Webex APIs

- Coverage
  - Users (incl. calling entitlements), locations (r/o), call pickups, call queues, hunt groups, auto attendant, call parks, schedules, voice messaging settings, …
  - person settings: barge, call forwarding, call intercept, call recording, caller ID, voicemail settings, …
- Currently new API endpoints added on a monthly basis
- Foundation for flexible automation .. not only during migrations
- Reference: https://developer.webex.com

Using Webex APIs

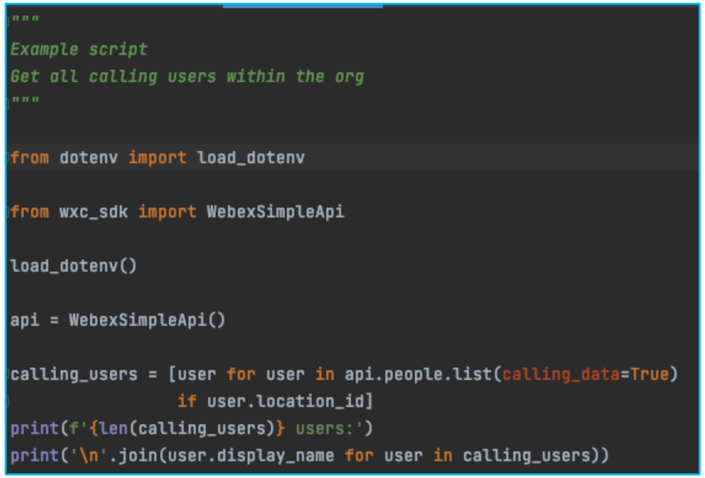
- Documentation at: https://developer.webex.com/
- But: Steep learning curve
- A lot of concepts to master
- SDK helps to abstract from the "dirty details"



- https://pypi.org/project/webexteamssdk/: great framework, but no support for Webex Calling specific provisioning

wxc_sdk: SDK for Webex Calling APIs

- PyPi: https://pypi.org/project/wxc-sdk/
- Documentation: https://wxc-sdk.readthedocs.io/en/latest/
- Simple SDK to work with Webex APIs
  - Focus on Webex Calling specific endpoints
- Takes care of all the "ugly" stuff
  - JSON (de-)serialization, authentication, 429 retries,
  - Pagination, …
- Python objects for all API objects
  - Tab completion → efficient coding
- Actively maintained
  - New API endpoints will be added
- Foundation for your migration/provisioning automation and other projects around Webex Calling

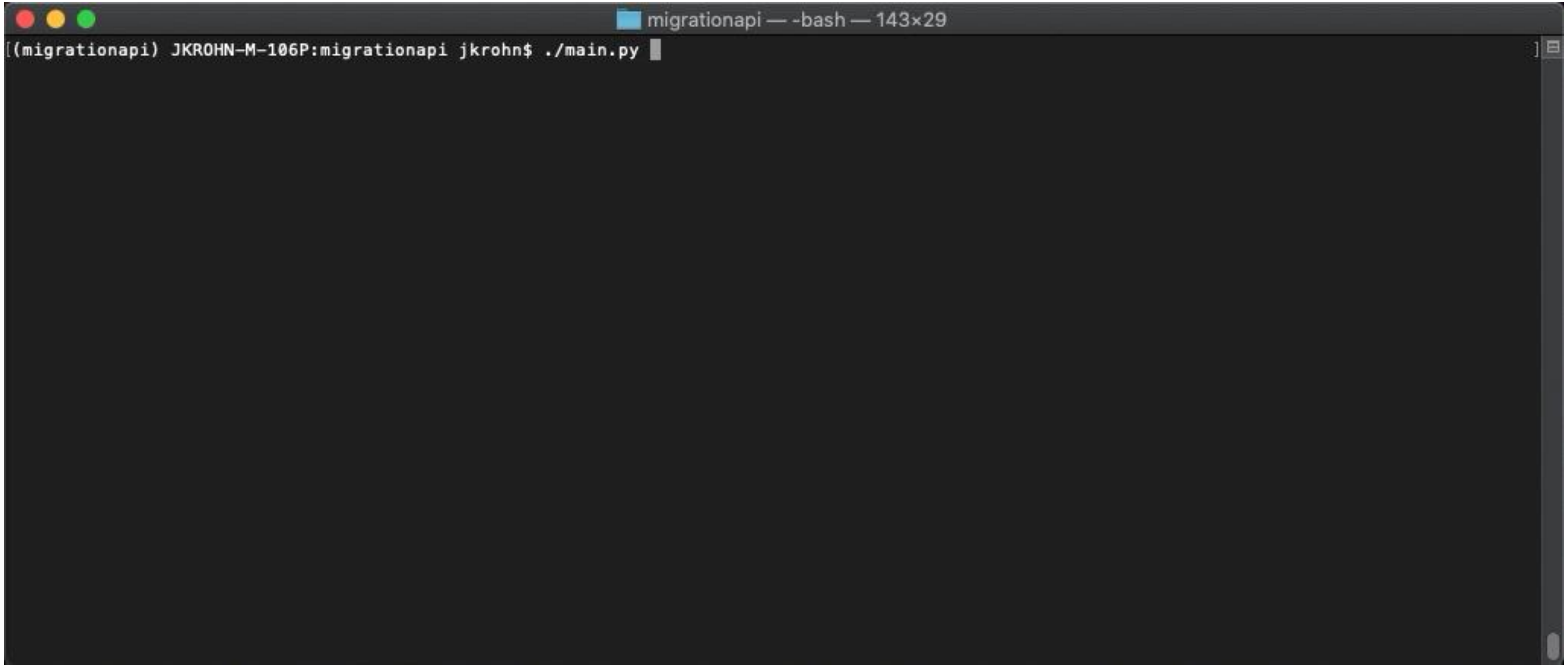


```
"""
Example script
Get all calling users within the org
"""

from dotenv import load_dotenv

from wxc_sdk import WebexSimpleApi

load_dotenv()

api = WebexSimpleApi()

calling_users = [user for user in api.people.list(calling_data=True)
                 if user.location_id]
print(f'{len(calling_users)} users:')
print('\n'.join(user.display_name for user in calling_users))
```

Demo Framework

- https://github.com/jeokrohn/migrationapi
- Read users from Unified CM via AXL
- Select users with phone numbers in a specific range
- Provision these users for Webex Calling and assign their extension
  - Async calls b/c Webex Calling provisioning calls are slow
  - Async code allows concurrent execution of multiple REST API calls
- Access Token for Webex API has to be obtained from developer.cisco.com

Demo



migrationapi — -bash — 143×29

```
[(migrationapi) JKROHN-M-106P:migrationapi jkrohn$ ./main.py
```

Observations

- Each Webex Calling provisioning request takes multiple seconds to complete
- Concurrent execution of requests helps to speed up the provisioning

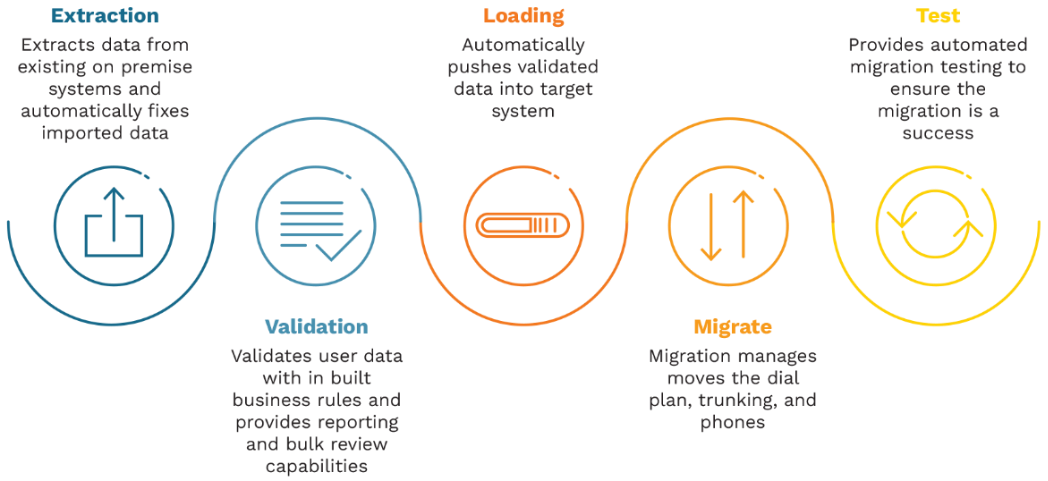
wrangler_

yarnlab

Feature Summary

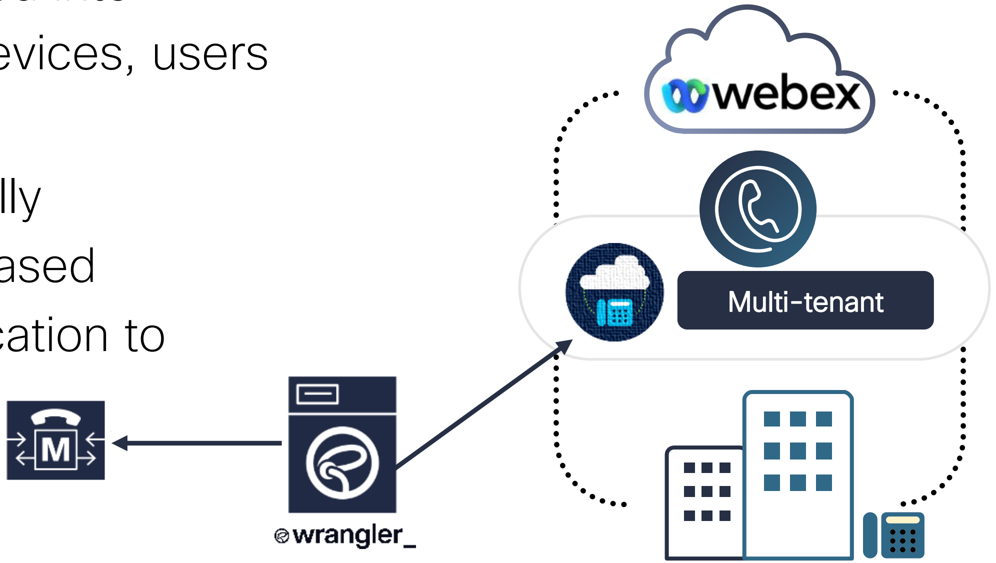
- Data extraction v10.5+ from CUCM/UCXN
- Data Validation
  - Normalization
  - Transformation
  - Correction
- Configuration mapping to WbC-MT
- Validation issues for configuration changes
- Data load via APIs
- Multiple source clusters
- Phased Migration in batches
  - Inter site dependency reporting
- Dial plan analysis and connection

Migration Process - Automated Webex Calling Migration


Extraction
Extracts data from existing on premise systems and automatically fixes imported data
Validation
Validates user data with in built business rules and provides reporting and bulk review capabilities
Loading
Automatically pushes validated data into target system
Migrate
Migration manages moves the dial plan, trunking, and phones
Test
Provides automated migration testing to ensure the migration is a success

Data Extraction

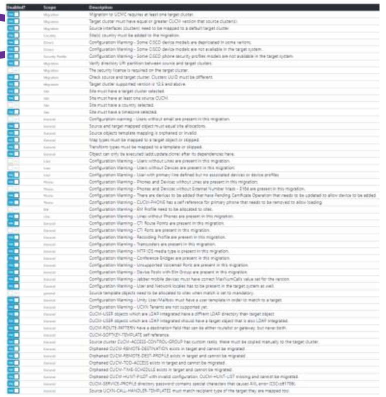
- Configuration is imported from the source and target organisation and imported into wrangler_ including all lines, devices, users and dial plan
- Configuration can be periodically rediscovered if required for phased migrations using change notification to minimize data freezes



Allocation Validation and Mapping

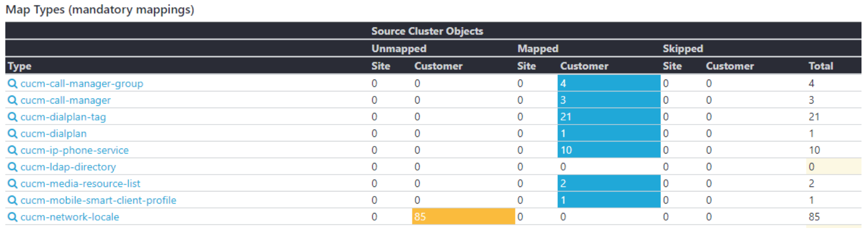
- Objects are automatically allocated to sites and sites are used to define migration batches
- Data is validated using validation rules that detect problems with configuration and raise issues for resolution
- Key issues are raised for Webex Calling migration such as cross location restrictions
- Target mappings are set to map sites, call barring levels, etc. between UCM and Webex Calling

Validation Rules



Enabled?	Scope	Description
	Migration	Migration to UCMC requires at least one target cluster.
	Migration	Target cluster must have equal or greater CUCM version that source cluster(s).
	Migration	Source interfaces (clusters) need to be mapped to a default target cluster.
	Country	Site(s) country must be added to the migration.
	Device	Configuration Warning - Some CISCO device models are depricated in some versions.
	Device	Configuration Warning - Some CISCO device models are not available in the target system.
	Security Profile	Configuration Warning - Some CISCO phone security profiles models are not available in the target system.
	Migration	Verify directory URI partition between source and target clusters.
	Migration	The security license is required on the target cluster.
	Migration	Check source and target cluster. Clusters UUID must be different.
	Migration	Target cluster supported version is 12.5 and above.
	Site	Site must have a target cluster selected.
	Site	Site must have at least one source CUCM.
	Site	Site must have a country selected.
	Site	Site must have a timezone selected.
	General	Configuration warning - Users without email are present in this migration.
	General	Source and target mapped object must equal site allocations.
	General	Source objects template mapping is orphaned or invalid.
	General	Map types must be mapped to a target object or skipped.
	General	Transform types must be mapped to a template or skipped.
	General	Object can only be executed (add,update,clone) after its dependencies have.
	User	Configuration Warning - Users without Lines are present in this migration.
	User	Configuration Warning - Users without Devices are present in this migration.
	User	Configuration Warning - User with primary line defined but no associated devices or device profiles.
[illegible]	[illegible]	[illegible]

Target Mappings

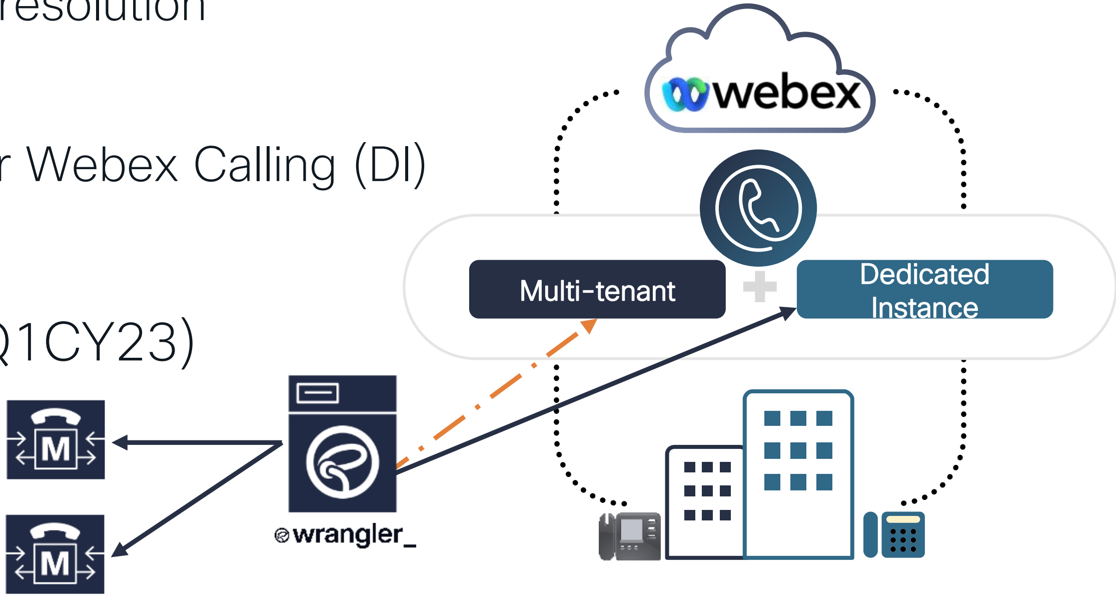


Map Types (mandatory mappings)

	Source Cluster Objects						
	Unmapped		Mapped		Skipped		
Type	Site	Customer	Site	Customer	Site	Customer	Total
cucm-call-manager-group	0	0	0	4	0	0	4
cucm-call-manager	0	0	0	3	0	0	3
cucm-dialplan-tag	0	0	0	21	0	0	21
cucm-dialplan	0	0	0	1	0	0	1
cucm-ip-phone-service	0	0	0	10	0	0	10
cucm-ldap-directory	0	0	0	0	0	0	0
cucm-media-resource-list	0	0	0	2	0	0	2
cucm-mobile-smart-client-profile	0	0	0	1	0	0	1
cucm-network-locale	0	85	0	0	0	0	85

Hybrid Migrations
Dedicated Instance and Multi-Tenant

- Multi Cluster consolidation
  - Duplicates identification and resolution
- Cluster separation
  - Megacluster redistribution for Webex Calling (DI)
- Phased migration
- Webex Calling (MT) (Late Q1CY23)
  - CUCM -> WxC-MT
  - CUCM -> WxC-DI + MT



Migration Options - Comparison

	Manual Provisioning	Control Hub Tool	Bulk Provisioning (CSV)	API	wrangler_
Ease of use	o initial training	++	- Process setup (data source, data format, ..)	-- Steep learning curve, development required for integration	+ Needs some training due to breadth of coverage
Flexibility	++	o limited to devices, users, numbers	+ limited coverage	++ increasing API coverage	++ uses Webex APIs
Integration into business processes	None	None	Possibly limited integration via customized data export/import	Tight integration possible	Primarily intended as standalone tool Other tools available supporting day to day operations
Speed	--	+	+	++	++

Closing

Summary

- Migration Process
- Discovery
- Design
- Deployment
- Migration

References

- Analyze Unified CM config exports: https://github.com/jeokrohn/ucmmigration
- API supported migration from Unified CM to Webex Calling, GDPR export, CSV export: https://github.com/jeokrohn/migrationapi
- Python SDK for Webex Calling provisioning: https://pypi.org/project/wxc-sdk/
- Yarnlab: https://www.yarnlab.io/

Key Takeaways

- User batches based on dependencies between users
- Unified CM and Webex Calling are different ☺
- Focus on business requirements instead of 1:1 feature mapping
- User provisioning: foundation for all Webex services
- Interworking between Unified CM and Webex Calling during transition period
  - Dial plan maintenance is key
- Migration options: tools, bulk operations, APIs
- APIs provide greatest flexibility and allow for tight integration in business logic
- 3rd party migration tools available to support easy migrations

Collaboration

Cloud Calling and Phones

Learn about cloud and hybrid calling design and troubleshooting, calling endpoints for those seeking to use cloud calling or migrating from an existing on-premise environment.

START

Feb 6 | 08:45
TECCOL-2191
Troubleshooting Cisco Webex Calling

Feb 6 | 14:15
TECCOL-2010
News in Webex Cloud Collaboration Security

START

Feb 6 | 14:15
TECCOL-2180
Webex Collaboration Interoperability – Video and Calling Integrations

Feb 7 | 11:30
BRKCOL-3818
Troubleshooting UCM Calling in the Webex App

Feb 7 | 17:00
BRKCOL-2787
Planning and Designing Successful Cloud Calling Deployments with Webex Calling

Feb 8 | 08:30
BRKCOL-2198
Deploying the Webex App to your Organization

Feb 8 | 10:30
BRKCOL-2312
High Capacity Premises-based PSTN Option for Webex Calling

Feb 8 | 11:45
IBOCOL-2420
Calling Migrations: an Interactive Session to Share Experiences, Ideas, Solutions, and Best Practices

Feb 9 | 08:30
BRKCOL-2812
Troubleshooting Webex Calling Premises-based PSTN

Feb 9 | 10:45
BRKCOL-2481
Successful Migrations from Unified CM to Webex Calling

Feb 9 | 12:00
BRKCOL-2993
Enabling Site Survivability for Webex Calling

Feb 9 | 12:30
BRKCOL-2066
Top Ten Tips for Deploying Cisco Phones in the Cloud

FINISH

Feb 9 | 13:45
BRKCOL-2990
Webex platform infrastructure: Where, How and Why we do it like this?

Learning maps online : https://www.ciscolive.com/emea/learn/technical-education/learning-maps.html

If you are unable to attend a live session, you can watch it On Demand after the event

CISCO Live!

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (open from Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at https://www.ciscolive.com/emea/learn/sessions/session-catalog.html

Continue Your Education

- Visit the Cisco Showcase for related demos.
- Book your one-on-one Meet the Engineer meeting.
- Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.
- Visit the On-Demand Library for more sessions at ciscolive.com/on-demand.

The bridge to possible

Thank you

cisco Live!

ALL IN