

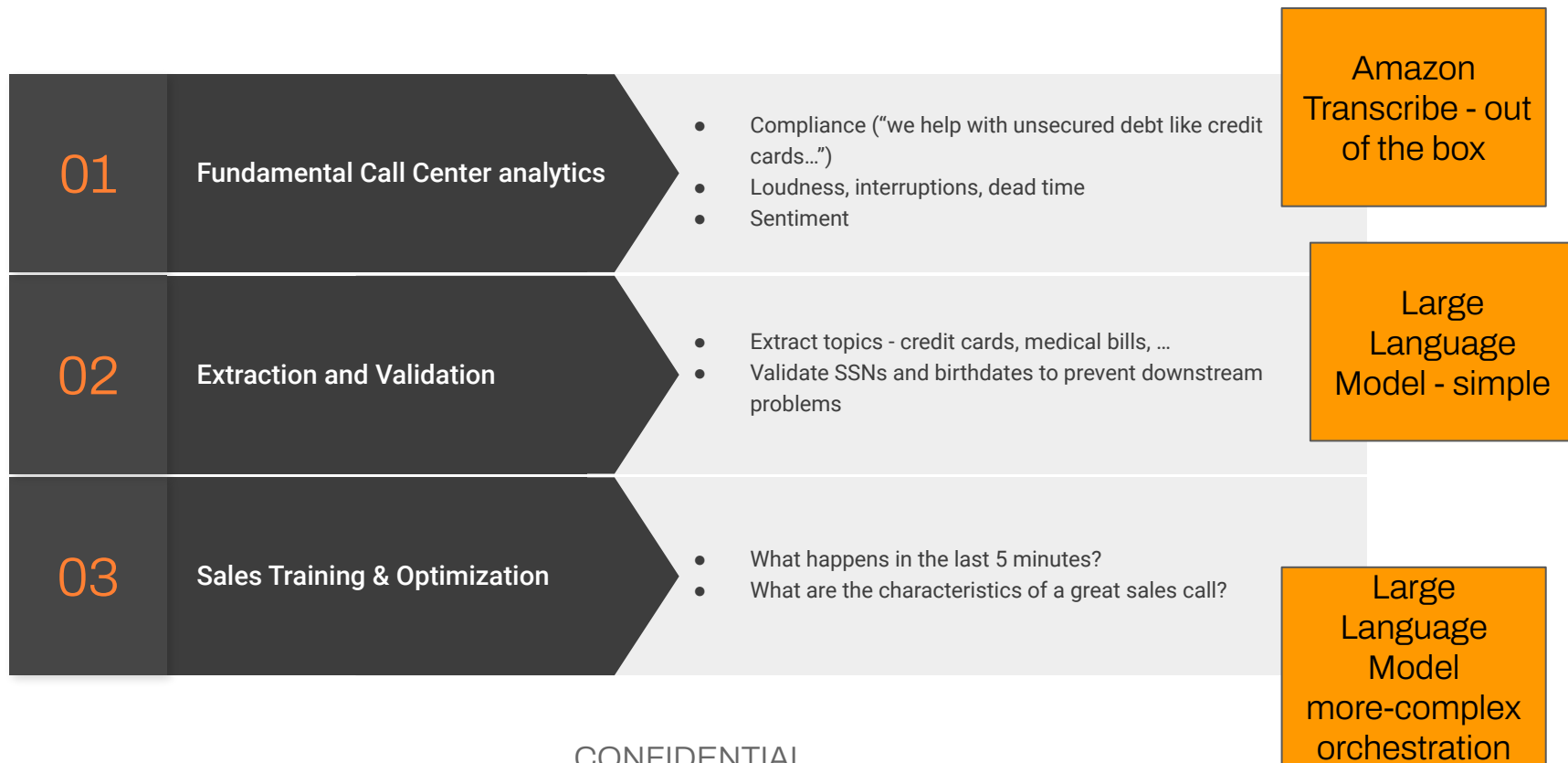


1200 calls/day  
20% longer than 20min  
= 80 hours of unused recordings/day  
→ *1600 hours / month*

Massive unstructured data holding the key to a higher close rate



# What's possible



CONFIDENTIAL



*Demo*



# Ingesting audio files

<https://github.com/big-cloud-country/audio-analytics>

This is how my they  
stored audio files

tracking_gui	Billable Flag	Recording	Buyer Reco	Transcript	Inbound Ver	Start Date Ti	Tracking
01632d96-ca82-460a-8338-cae2f		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=01632d96-ca82-460a-8338-cae2f">https://admins.callerready.com/Recordings/Recording?AccountSid=01632d96-ca82-460a-8338-cae2f</a>				04/12/2024 4:47	
07b74e61-9eaa-4822-a5b1-b41f44436bf0						04/13/2024 6:08	
0e223789-426d-4a45-9c4e-f056c		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=0e223789-426d-4a45-9c4e-f056c">https://admins.callerready.com/Recordings/Recording?AccountSid=0e223789-426d-4a45-9c4e-f056c</a>				04/12/2024 4:46	
10397b57-b148-4892-8b0c-c87c		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=10397b57-b148-4892-8b0c-c87c">https://admins.callerready.com/Recordings/Recording?AccountSid=10397b57-b148-4892-8b0c-c87c</a>				04/12/2024 4:44	
1201d2b9-acbf-42c4-bab8-b0b42		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=1201d2b9-acbf-42c4-bab8-b0b42">https://admins.callerready.com/Recordings/Recording?AccountSid=1201d2b9-acbf-42c4-bab8-b0b42</a>				04/12/2024 4:50	
237cb6a5-dba1-4493-8751-8672f		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=237cb6a5-dba1-4493-8751-8672f">https://admins.callerready.com/Recordings/Recording?AccountSid=237cb6a5-dba1-4493-8751-8672f</a>				04/12/2024 4:45	
274ac510-3652-486c-a6fd-96c64		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=274ac510-3652-486c-a6fd-96c64">https://admins.callerready.com/Recordings/Recording?AccountSid=274ac510-3652-486c-a6fd-96c64</a>				04/12/2024 4:57	
3a448a90-b3ad-488e-80cd-b4b5f		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=3a448a90-b3ad-488e-80cd-b4b5f">https://admins.callerready.com/Recordings/Recording?AccountSid=3a448a90-b3ad-488e-80cd-b4b5f</a>				04/12/2024 4:55	
3aff0f62-62a1-4944-a397-2b95af		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=3aff0f62-62a1-4944-a397-2b95af">https://admins.callerready.com/Recordings/Recording?AccountSid=3aff0f62-62a1-4944-a397-2b95af</a>				04/12/2024 4:44	
3f4d73f7-1b09-4010-85a8-7baee		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=3f4d73f7-1b09-4010-85a8-7baee">https://admins.callerready.com/Recordings/Recording?AccountSid=3f4d73f7-1b09-4010-85a8-7baee</a>				04/12/2024 4:52	
3fb32b50-e568-4760-84c1-c8c96		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=3fb32b50-e568-4760-84c1-c8c96">https://admins.callerready.com/Recordings/Recording?AccountSid=3fb32b50-e568-4760-84c1-c8c96</a>				04/12/2024 4:57	
42629c32-2fa8-470b-9870-4e0c8		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=42629c32-2fa8-470b-9870-4e0c8">https://admins.callerready.com/Recordings/Recording?AccountSid=42629c32-2fa8-470b-9870-4e0c8</a>				04/12/2024 4:48	
4f0b3186-8c11-4bc8-b3c9-dcb9e		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=4f0b3186-8c11-4bc8-b3c9-dcb9e">https://admins.callerready.com/Recordings/Recording?AccountSid=4f0b3186-8c11-4bc8-b3c9-dcb9e</a>				04/12/2024 4:49	
561504e1-0243-4b56-92af-0ae6e		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=561504e1-0243-4b56-92af-0ae6e">https://admins.callerready.com/Recordings/Recording?AccountSid=561504e1-0243-4b56-92af-0ae6e</a>				04/12/2024 4:45	
5a14ccf8-6c1e-45e5-abe7-b3f00b92abec						04/12/2024 4:54	
5b6b469b-d91c-46bf-aed3-42388		<a href="https://admins.callerready.com/Recordings/Recording?AccountSid=5b6b469b-d91c-46bf-aed3-42388">https://admins.callerready.com/Recordings/Recording?AccountSid=5b6b469b-d91c-46bf-aed3-42388</a>				04/12/2024 4:43	

CONFIDENTIAL



# Transcribe the audio

app.py run() method

```
files = os.listdir('mp3-downloads')
s3 = boto3.client('s3', region_name='us-east-1')
for file_path in files:
    f = f'mp3-downloads/{file_path}'
    resp = s3.upload_file(f, mp3_upload_bucket, file_path)
    if resp is not None:
        print(f"Failed to upload {file_path} to {call_log_bucket}")
    else:
        print(f"Uploaded {file_path} to {mp3_upload_bucket}")
        # start transcription jobs
        job_name = file_path.split('/')[-1].split('.')[0]
        job_name = f"{job_name}-{batch_id}"
        media_s3_uri = f"s3://{mp3_upload_bucket}/{file_path}"
        output_bucket = mp3_upload_bucket
        output_key = f"transcription-outputs/{batchid}-{batch_id}/{job_name}.json"
        transcription_job_response = start_transcription_job(job_name, media_s3_uri, output_bucket, output_key)
        # call_analytics_job_response = start_call_analytics_job(job_name, media_s3_uri, output_bucket)
        transcription_job_ids.append(transcription_job_response['TranscriptionJob']['TranscriptionJobName'])
        # call_analytics_job_ids.append(call_analytics_job_response['CallAnalyticsJob']['CallAnalyticsJobName'])
completed_transcription_jobs = set()
```



Make an “objection library” → *what's an objection?*

```
{
  "id": "misunderstanding_offer",
  "example_objection": "I thought it was a loan program",
  "example_response": "This is not a loan program, it's a debt relief program"
},
{
  "id": "minimum_debt_amount",
  "example_objection": "I don't have enough debt to qualify",
  "example_response": "Our minimum debt amount to qualify is 7500."
},
{
  "id": "unsure_about_debt_amount",
  "example_objection": "I'm not really sure about the amount of debt I have",
  "example_response": "That's okay, we can help you gather that information and determine if you meet the mi
```



Give an example for how to score (ProcessingMethods.py)

```
scoring_example=[{  
    "id": "legitimacy",  
    "example_objection": "I'm not sure if this is real or not",  
    "actual_objection": "Is this real?",  
    "example_response": "We've been in business for 15 years and have been accredited by the better business  
    "agent_response": "We've been around for a while and have a good reputation",  
    "score": 5  
}]
```



## Scoring prompt (ProcessingMethods.py)

```
def create_objection_scoring_prompt(library, transcript, scoring_example): 2 usages  ⤴ Robert Whelan
    prompt = f"""
{transcript}

---

objection library:
{json.dumps(library)}

---

above is a phone conversation transcript, followed by an objection library. The agent who received
the call is "spk_0". The customer is "spk_1". The company taking the call offers structured debt services for
with debts they cannot pay off, such as credit card debt or medical bills. This is a sales call
where the agent seeks to get the customer to commit to the service.

Find all the objections the customer discusses in this call, such as concerns about whether
the offer is legitimate, and whether the credit check will hurt them. Compare them to the list of
accepted objections and responses in the library. For each objection, state how the agent handles responds.
then, give the agent a score based on how well they handled the objection. If the agent's response is
close to the accepted response, give them a score of 10. If the agent's response is the opposite of
the accepted response, give a 0. You can give numbers in between too. Return an array of objects like this:
{json.dumps(scoring_example)}

JSON:
"""

    return prompt
```





## Score each transcript against each objection using Mistral

```
def invoke_model(prompt): 4 usages (1 dynamic) ⓘ Robert Whelan
    bedrock_runtime = boto3.client('bedrock-runtime', region_name='us-east-1')
    enclosed_prompt = "<s>[INST]" + prompt + "[/INST]"
    body = {
        "prompt": enclosed_prompt,
        "max_tokens": 1000,
    }
    response = bedrock_runtime.invoke_model(
        body=json.dumps(body),
        contentType='application/json',
        accept='application/json',
        modelId='mistral.mistral-7b-instruct-v0:2'
    )
    response_body = json.loads(response["body"].read())
    return response_body
```



## Score json output:

```
{
  "tracking_guid": "RE140d7767d0b97cf299c70567499846ae-batch-1",
  "responses": [
    {
      "id": "unfamiliar_creditors",
      "example_objection": "I don't know which creditors I have and which debts qualify",
      "actual_objection": "I'm not sure which creditors you're referring to",
      "example_response": "I can pull up all the debts on your credit report for you",
      "agent_response": "We reached out to you about a specific hardship program for unsecured debts like c",
      "score": 8
    },
    {
      "id": "credit_check_concerns",
      "example_objection": "I'm worried about the credit check hurting my score",
      "actual_objection": "I'm concerned about the impact on my credit score",
      "example_response": "It's a soft credit pull, which doesn't affect your credit score negatively",
      "agent_response": "They'll ask you for a temporary lift on the freeze on the credit report, and it's",
      "score": 10
    },
  ]
}
```



# Aggregate and profit

average score and frequency

