

Lyrics Implementation Discussion with Suno

Open Questions :

There are a few critical items in regards to lyrics implementation:

- What does tag mean in latency file? Is the latency for chunk or entire lyrics?
 - tags are genre tags. Flow is song title → genre / style tags (would be separated out) → lyrics.
 - **Concerns: high latency for lyrics generation & not timestamped.**

	A	B	C	D	E	F	G
1	Item	prereq	Model	min	p50	p95	p99
2	Full Lyrics + Tags			3.1	6.3	11.8	18.6

- As per the Suno Lyrics Ingestion Proposal doc, Suno is generating song audio in 30s chunks at >2x realtime speed (so first chunk is available at ~15s). Does it mean time-marked lyrics generation for entire song would take approximately half the total audio duration. For instance, a 4-minute song would require about 2 minutes to generate complete lyrics data?
 - Polling lyrics chunks won't be feasible option for Alexa.
- We're moving forward with retrieving lyrics via URL where the provider would share the lyrics object in the initiate response. (1. Initiate Response)
- We would need Suno to provide the lyrics in industry standard format like [WEBVTT](#) and [LRC](#) and discuss how Suno will represent the lyrics when timestamp is missing (create a song request) because there is no industry standard for representing the WEBVTT and LRC without timestamp. - 2. LRC format, 3. WEBVTT format
 - **Note:** LRC doesn't have explicit end time for each line meaning a line will be displayed (highlighted) until the next timestamped line appears. If there is a gap between two lines (for instance, 10s beats between the two lines), the previous line remains visible and highlighted. On the other hand, WebVTT explicitly defines the start and end time for each line. Each line will have a fixed duration even if another line doesn't immediately follow. If there's a gap, we would be happen to render it with audio beats by adding some additional parsing rule (4. Special case when long beats (>3-5s) are present.). **Recommendation for the provider would be go with WebVTT**
 - LRC is basically used for KARAOKE whereas WebVTT is used for subtitles, captions and lyrics in video and audio.
 - **Suno agreed to use WebVTT format**
- We must also address lyrics access control (authorization and authentication) with Suno since lyrics data could be sensitive and confidential info. Few options we are currently exploring (we need to consult InfoSec team to get clarity on what is the standardized process within Alexa for such use-cases)
 - Get a pre-signed URL with X TTL.
 - IAM based access control.
 - ACCESS token can be used (the same token when used during the lambda call).
- We also need to consult InfoSec team on whether do we want the provider to encrypt lyrics file data? If yes, how will Alexa service decrypt the lyrics data? Is there any standardized approach?
 - **Not feasible as of now since data is stored at multiple places within Suno.**

Meeting Notes

Meeting Date : 12th March, 2025

ATTENDEES: PRAGYA SINGHANIA, SANTHOSH ADARI, ROHIT ROONGTA AND CHI FENG (SUNO)

Meeting Notes:

- **Latency Concern:** As per the shared latency file, the P95 (11.8) and P99 (18.6) latency for lyrics generation are higher than the normal timeout of 10 seconds. This is a concern because the lyrics are retrieved immediately after the Initiate call to display the now-playing screen on the device. To address this issue, the Suno team will investigate the feasibility of reducing latency and ensuring that the lyrics (without timestamps for create requests) are ready and available before the Initiate response. Additionally, it's important to note that polling the lyrics in chunks won't be feasible for Alexa at this time due to the additional effort required for discussions, alignments with relevant teams, architectural changes, and implementation.
- **Lyrics File Format:** Suno has agreed to adhere to the WebVTT file format for both play and create requests. In cases where the timestamp is missing, the WebVTT file will be missing the timestamp section.
- **Authentication and Authorization:** For dev E2E and integration testing, we have a pre-signed URL, but we may need to transition to a different approach based on internal InfoSec recommendations and Alexa's standardized approach for data retrieval. I'll take care of this action item and keep everyone in loop.
- **Create a song CX:** Regarding the customer experience (CX), when the timestamp data is missing, Alexa will disable auto-scrolling and auto-highlighting for the lyrics and implement constant-speed scrolling.

Here are a few additional questions:

- Could Suno provide a few WebVTT data file samples for both timestamped and non-timestamped data?
- As mentioned earlier, we'll implement constant scrolling for lyrics when the timestamp data is missing. Could Suno provide a rough estimate for the total song duration?
- (Just a thought for future development and discussion) Is it possible for Suno to include an ICY/ID4 tag within the audio streaming URL or publish an event to indicate that the lyrics (either new chunks or timestamped data) are now available for fetching?

Reference

1. Initiate Response

We would make modification to SPI "Item" object to include the lyrics object as Optional field.

1.1 Lyrics Structure

	Attribute	Description	Required	Type
1	id	An identifier for the lyrics that is opaque to Alexa.	yes	string
2	uri	URI of the lyric file. This URI is sent to the Alexa Audio service to display lyrics along with the track. URIs must be HTTPS.	yes	string
3	format	The lyric file format. Supported formats: WEBVTT , LRC	yes	enum

1.2 Example

```

"lyrics": {
  "id": "LYRICID_92_2382736",
  "uri": "https://cdn.example.com/api/1/a2f318467fbf2829996adc0880e0abd03d03b1ba6ac.vt
  "type": "WEBVTT"
}

"lyrics": {
  "id": "LYRICID_92_2382736",
  "uri": "https://cdn.example.com/api/1/a2f318467fbf2829996adc0880e0abd03d03b1ba6ac.lr
  "type": "LRC"
}

```

1.3 Initiate Response Example (as a whole)

```

{
  "Output":
  {
    "__type": "com.amazon.alexamusicsskillskit.service.operations#InitiateResponse",
    "playbackMethod":
    {
      "__type": "com.amazon.alexamusicsskillskit.service.types#AudioPlayerQueue",
      "additionalMetadata":
      {
        "TIME_ZONE": "America/Atikokan",
        "LOCALE": "en-US",
        "ELF_ENABLED": "false",
        "SKILL_DEFINITION": "{\\"serviceName\\":null,\\"stage\\":\\"development\\",\\"
        "COUNTRY_CODE": "US"
      },
      "controls": [
        {
          "__type": "ToggleControl",
          "enabled": false,
          "name": "SHUFFLE",
          "selected": false
        },
        {
          "__type": "ToggleControl",
          "enabled": false,
          "name": "LOOP",
          "selected": false
        },
        {
          "__type": "CommandControl",
          "enabled": false,
          "name": "NEXT"
        },
        {
          "__type": "CommandControl",
          "enabled": false,
          "name": "PREVIOUS"
        }
      ]
    }
  }
}

```

```

        "__type": "AdjustControl",
        "enabled": true,
        "name": "SEEK_POSITION"
    ]],
    "firstItem":
    {
        "controls": [
            {
                "__type": "CommandControl",
                "enabled": true,
                "name": "NEXT"
            },
            {
                "__type": "CommandControl",
                "enabled": true,
                "name": "PREVIOUS"
            },
            {
                "__type": "AdjustControl",
                "enabled": true,
                "name": "SEEK_POSITION"
            }
        ],
        "durationInMilliseconds": 224000,
        "feedback":
        {
            "type": "PREFERENCE",
            "value": "POSITIVE"
        },
        "id": "track-song.7595",
        "metadata":
        {
            "__type": "TrackMetadata",
            "art":
            {
                "sources": [
                    {
                        "heightPixels": 600,
                        "size": "X_LARGE",
                        "url": "tracks/twenty04/album.875/54373_Ian-Post-Odyssey-/",
                        "widthPixels": 600
                    },
                    {
                        "heightPixels": 256,
                        "size": "LARGE",
                        "url": "tracks/twenty04/album.875/54373_Ian-Post-Odyssey-/",
                        "widthPixels": 256
                    },
                    {
                        "heightPixels": 110,
                        "size": "MEDIUM",
                        "url": "tracks/twenty04/album.875/54373_Ian-Post-Odyssey-/",
                        "widthPixels": 110
                    },
                    {
                        "heightPixels": 60,

```

```

        "size": "SMALL",
        "url": "tracks/twenty04/album.875/54373_Ian-Post-Odyssey-/",
        "widthPixels": 60
    },
    {
        "heightPixels": 48,
        "size": "X_SMALL",
        "url": "tracks/twenty04/album.875/54373_Ian-Post-Odyssey-/",
        "widthPixels": 48
    }
  ]
},
"authors": [],
"name": {
  "display": "In Orbit",
  "speech": {
    {
      "__type": "PlainTextSpeechInfo",
      "text": "In Orbit"
    }
  },
  "type": "TRACK"
},
"playbackInfo": {
  "type": "DEFAULT"
},
"rules": {
  "feedbackEnabled": false
},
"stream": {
  "id": "stream-track-song.7595",
  "offsetInMilliseconds": 0,
  "requestHeaders": [],
  "uri": "tracks/twenty04/album.875/InOrbit by IanPost Artlist.mp3"
},
"lyrics":
{
  "id": "LYRICID 92 2382736",
  "uri": "https://cdn.example.com/api/1/a2f318467fbf2829996adc0880ec",
  "type": "LRC"
},
"queueId": "queue-FHJPow5ICbsjaUVJ0sAuGzINx",
"queueIdsToDeactivate": [],
"rules": {
  "feedback": {
    "enabled": true,
    "type": "PREFERENCE"
  }
}

```

```
    }  
  },  
  "Version": "1.0"  
}
```

2. LRC format

LRC is short form of lyrics. It is a computer file format that synchronizes song lyrics with an audio file. Example : https://helpguide.sony.net/gbmig/42708101/v1/eng/contents/03/03/11/11.html#l3_notes%20on%20creating%20lrc%20files

Simple LRC format example

```
[ti:Song Title]  
[ar:Artist Name]  
[al:Album Name]  
  
[00:12.00]This is the first line of lyrics  
[00:17.20]Here's the second line  
[00:21.10]And the third one, with repeats  
[00:45.10]For the chorus  
[01:10.00]etc. etc. etc.
```

LRC without timestamp example

```
[ti:Song Title]  
[ar:Artist Name]  
[al:Album Name]  
  
This is the first line of lyrics  
Here's the second line  
And the third one, with repeats  
For the chorus  
etc. etc. etc.
```

3. WEBVTT format

<https://www.w3.org/TR/webvtt1/>

WEBVTT example

```
WEBVTT  
  
00:00:00.000 --> 00:00:05.000  
This is the first line of text.  
  
00:00:05.000 --> 00:00:10.000
```

This is the second line.

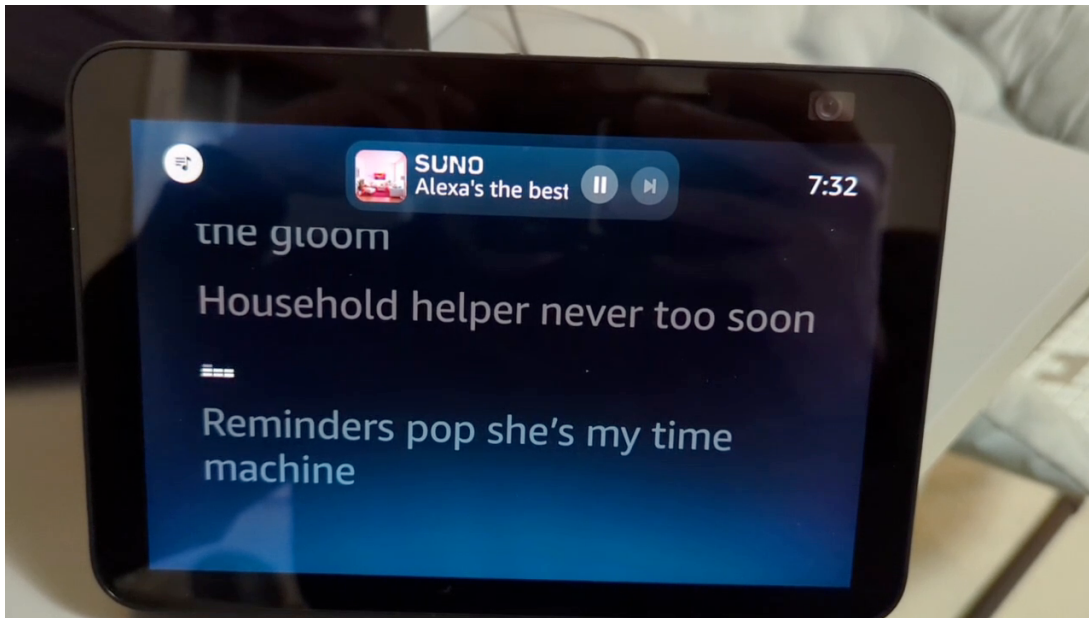
WEBVTT without timestamp example

WEBVTT

This is the first line of text
This is the second line of text

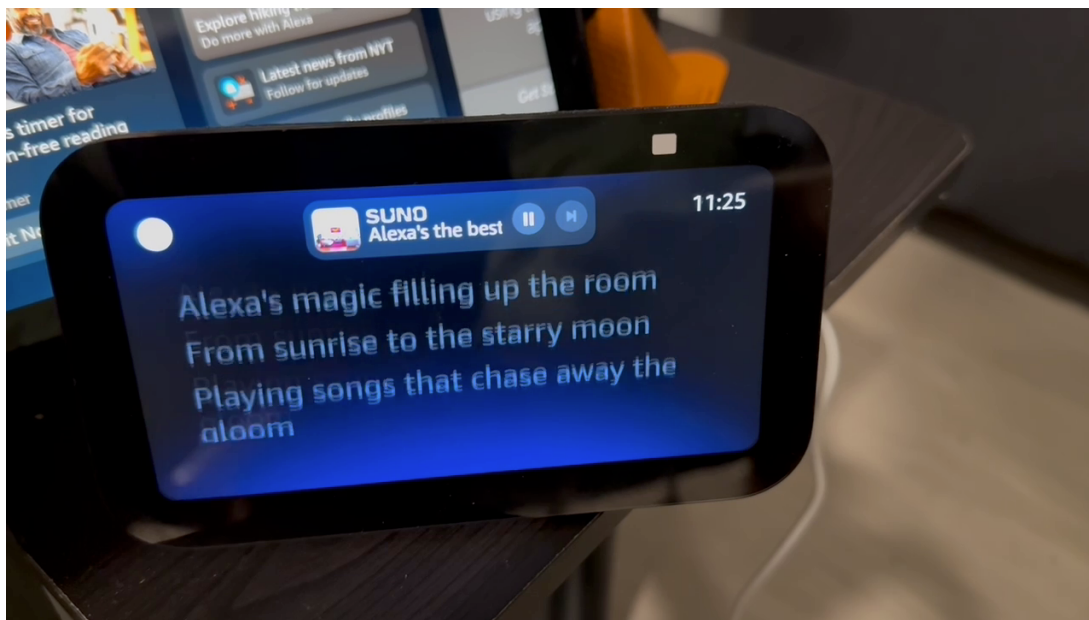
4. Special case when long beats (>3-5s) are present.

To mask beats with an emoji when playing



Video: [Beat.mp4](#)

When audio beats are not masked the experience would look this



Video: [Not emoji.mov](#)