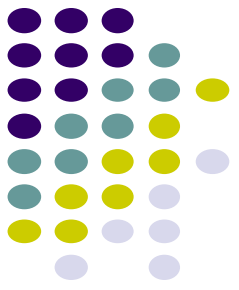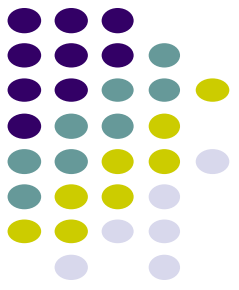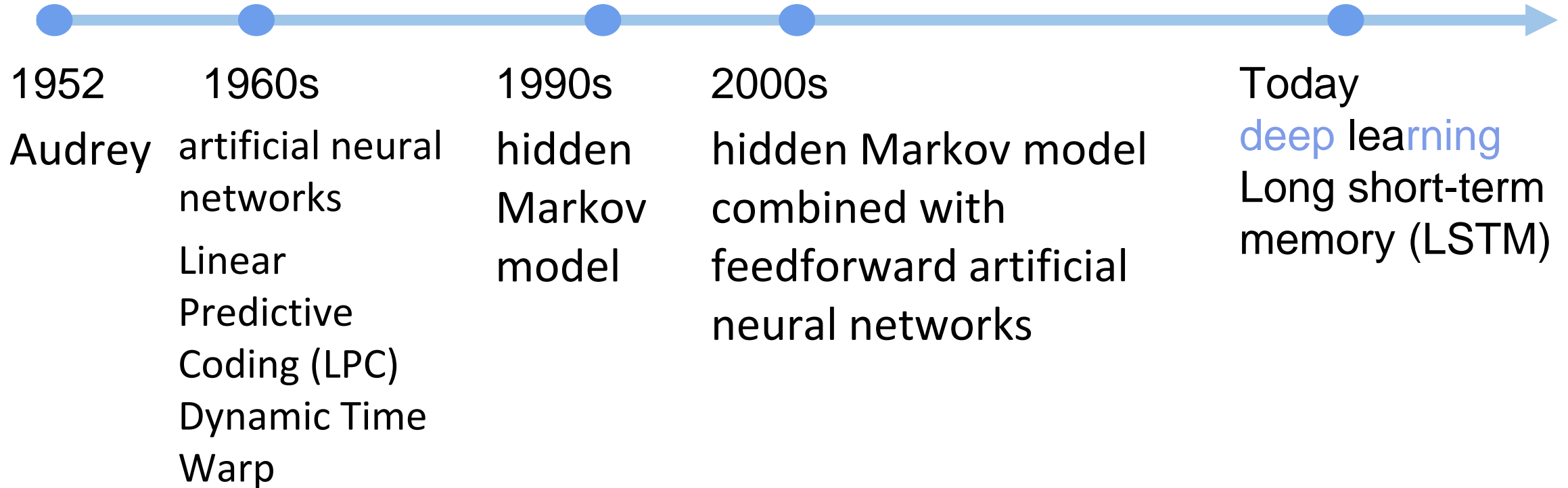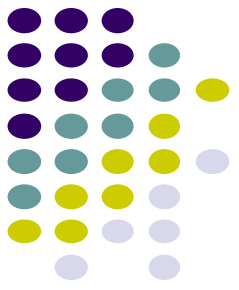# Speaking to Android

- Speech Recognition
  - Enable the recognition and translation of spoken language into text by computers
  - Automatic speech recognition (ASR) / speech to text (STT)
- Voice Actions
  - Let users quickly complete tasks in app using voice commands
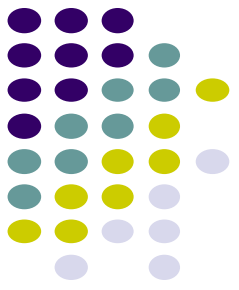
History of Speech Recognition

- In 1952, a system called 'Audrey' was developed by AT&T Bell Lab, which is an automatic digit recognizer for single-speaker digit recognition (around 10 words)

- In the 1960s, artificial neural networks were introduced into speech recognition. Linear Predictive Coding (LPC) and Dynamic Time Warp were two major breakthroughs

- In 1990s, the most significant breakthrough in speech recognition technology is the application of the hidden Markov model. Xuedong Huang developed the Sphinx-II system at CMU. The Sphinx-II system was the first to do speaker-independent, large vocabulary, continuous speech recognition and it had the best performance in DARPA's 1992 evaluation.

Development of Speech Recognition

1952

Audrey

1960s

artificial neural networks

Linear Predictive Coding (LPC)

Dynamic Time Warp

1990s

hidden Markov model

2000s

hidden Markov model combined with feedforward artificial neural networks

Today

deep learning

Long short-term memory (LSTM)

Specific Problem

- Convenience: Free hands and make it faster, e.g. people who deal with multitask (they can search using speech while typing another document)
- To help people who cannot listen to the voice in some situation (like meeting)
- To help people who may not use mobile phone normally (finger/hands disability)
- Let products such as smart appliances can complete specific instructions by interacting with people

Use Case (Example)
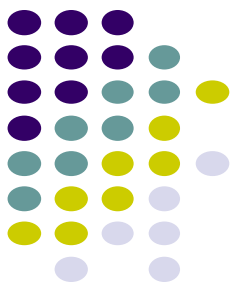
- AI app (Artificial Intelligence)
  - when a user launches a music app by saying "play some music", the app may want to ask the user "what genre?"

- IoT (Internet of Things)
  - when a home automation app hears the user say "OK Google, turn on the lights", it might want to ask "which room?" The Voice Interaction API lets apps ask follow-up questions like these.

- Robot (e.g. Echo Dot)
  - when a user try to communicate with a robot, the robot can know what the user wants to represent and complete the requirements

# Real world examples of where it is being used
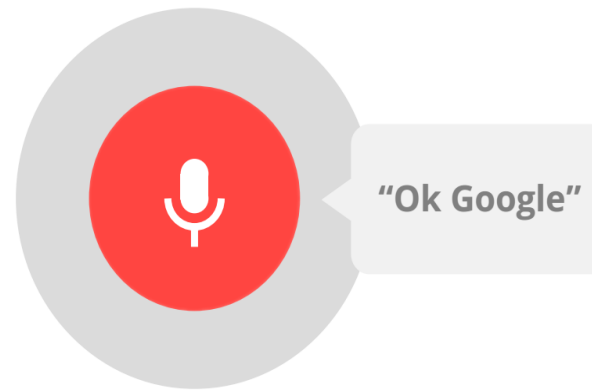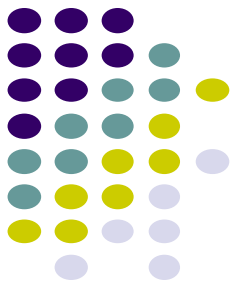
Code snippet(Speech to Text)
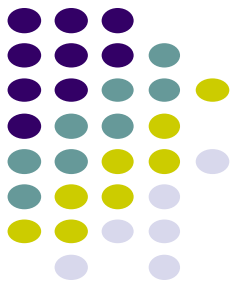
```java
// Showing google speech input dialog

private void askSpeechInput() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
            value: "Team3 demo for tech talk, speak something!");
    try {
        startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
    } catch (ActivityNotFoundException a) {


    }
}
```
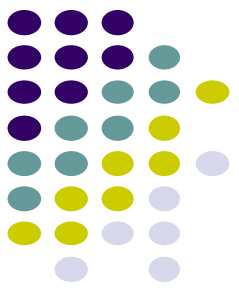
Code snippet(Speech to Text)

```java
// Receiving speech input

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case REQ_CODE_SPEECH_INPUT: {
            if (resultCode == RESULT_OK && null != data) {

                ArrayList<String> result = data
                        .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                voiceInput.setText(result.get(0));
            }
            break;
        }

    }
}
```
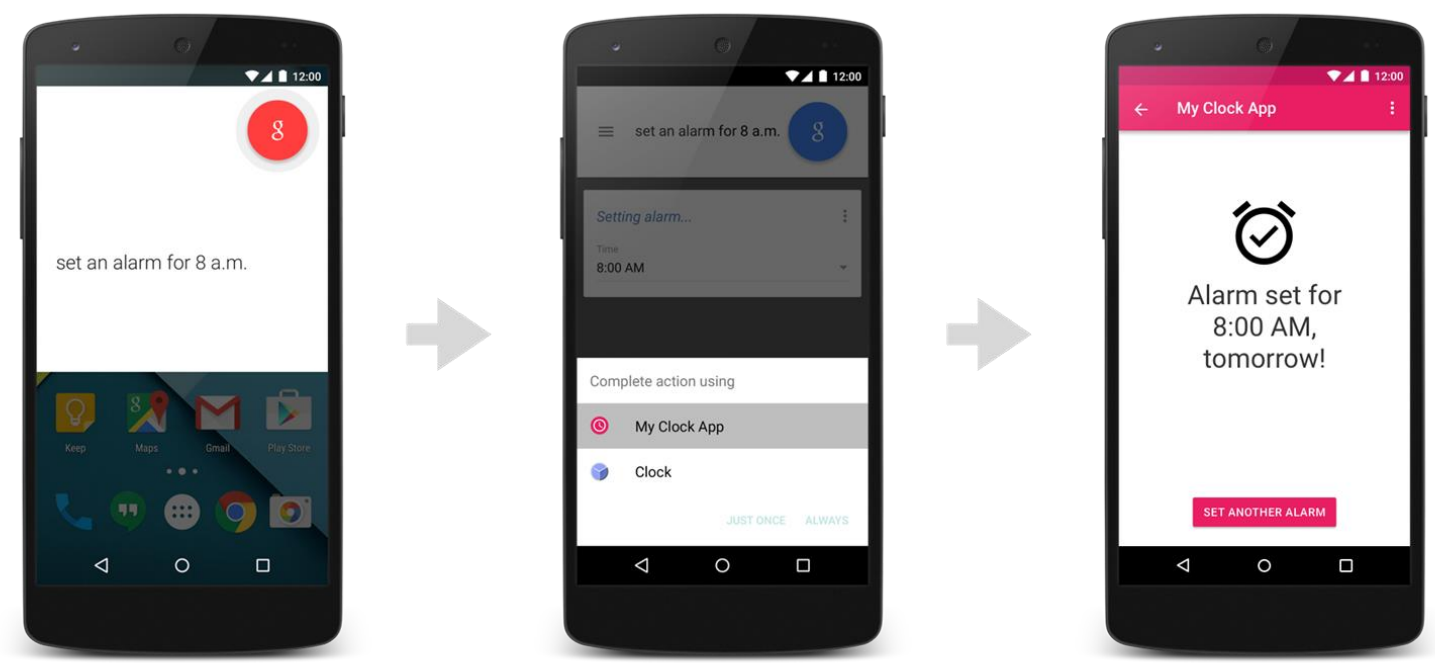
Overview of how it works?

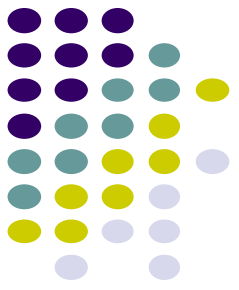## 1.Define an intent filter

```
<activity ...>
    <intent-filter>
        <action android:name="android.intent.action.SET_ALARM"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

Alarm Actions

Communication Actions

Fitness Actions

Media actions

....

Overview of how it works?

2.Handle the intent in your app

When your app receives the intent, it should perform the action.

```
// Get the intent
Intent intent = getIntent();
if (AlarmClock.ACTION_SET_ALARM.equals(intent.getAction())) {
    if (intent.hasExtra(AlarmClock.EXTRA_HOUR)) {
        // Step 2: get the rest of the intent extras and set an alarm
        ...
    }
}
```

Overview of how it works?
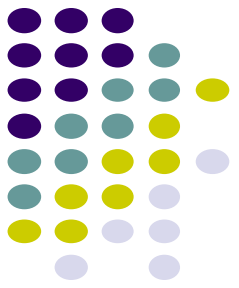
3.Update your app completion status

```
Thing alarm = new Thing.Builder()
        .setName("Alarm for 4:00 PM")
        .setDescription("Alarm set for 4:00 PM, with the 'Argon' ringtone"
                        + " and vibrate turned on.")
        .setUrl(APP_URI)
        .build();


Action setAlarmAction = new Action.Builder(Action.TYPE_ADD)
        .setObject(alarm)
        .setActionStatus(Action.STATUS_TYPE_COMPLETED)
        .build();


AppIndex.AppIndexApi.end(mClient, setAlarmAction);
```