Implementing a Smart Home Control Center with a GUI, RFID entry, door, window, and light control, air conditioner simulation, gas leak detection, fire detection, voice commands, and integration with Google Assistant and IFTTT can be a complex project. To help you get started, I'll provide a step-by-step guide:

Step 1: Hardware Setup

- Raspberry Pi 3B+: Set up your Raspberry Pi by installing the operating system (Raspberry Pi OS) and connecting it to the internet.

- RFID Reader: Connect the RFID reader module to your Raspberry Pi and configure it according to the manufacturer's instructions.

- Servo Motors: Connect the servo motors to the appropriate GPIO pins on your Raspberry Pi.

Step 2: Install Dependencies

- Update the system:

```bash

sudo apt update

sudo apt upgrade

```

- Install required packages:

```bash

sudo apt install python3-dev python3-pip

sudo pip3 install flask flask\_ask pyserial

```

Step 3: GUI Control Center

- Create a Python script to build the GUI Control Center using a library like Tkinter.

- Design the interface to include buttons or switches for controlling doors, windows, lights, and air conditioner.

- Implement the logic to handle the user interactions and control the corresponding devices.

Step 4: Mobile Application

- Develop a mobile application using frameworks like Flask .

- Implement the necessary functionalities to communicate with the Control Center over a network (I used IFTTT (google assistant and wehook with pythonanywhere)).

- Add options for controlling doors, windows, lights, and air conditioner.

Step 5: Air Conditioner Simulation

- Write code to control the servo motors connected to your Raspberry Pi.

- Define the appropriate angles for different temperature settings.

- Map the desired temperature values to servo motor positions.

- Integrate the air conditioner simulation code into the Control Center and mobile application.

Step 6: Gas Leak Detection

- Connect a gas sensor module to your Raspberry Pi.

- Write code to read data from the gas sensor and detect gas leaks.

- Implement a warning mechanism to alert the user via the Control Center and mobile application.

Step 7: Fire Detection

- Use flame sensors or other fire detection modules compatible with Raspberry Pi.

- Write code to detect fire based on the sensor readings.

- Implement an alarm or notification system to alert the user.

Step 8: Voice Commands and Integration

- Set up a Google Assistant developer account and project.

- Use Flask and Flask-Ask to create a webhook for handling voice commands from Google Assistant.

- Define voice commands to control various aspects of the Smart Home.

- Integrate the Control Center and mobile application with the voice command system.

Step 9: IFTTT Integration

- Set up an IFTTT (If This Then That) account and create the necessary applets.

- Connect IFTTT to your Google Assistant and implement triggers and actions.

- Define rules and automation scenarios using IFTTT to interact with the Smart Home.

Throughout the implementation, make sure to refer to the documentation and examples provided by the libraries, modules, and platforms you are using. Adapt and modify the steps as necessary based on your specific requirements and hardware setup.