

8

@yodaos-project

a new operating system
for voice user interface
based on the Web

TOC

Google Assistant Integrated YodaOS

What's the new at 8

YodaOS Community

Integrating YodaOS

With Google Assistant



```
net_manager[1647]: [INFO] 07-26 22:09:30.016716 [ipc/flora_upload.c] 21] network_flora_send_msg: { "network": { "state": "connected" } }  
net_manager[1647]: [INFO] 07-26 22:09:32.053140 [ipc/flora_upload.c]
```

离开时记得关闭
显示器电源哦！
ROKID 欢愉

What's the **new** at 8

New architecture for the better **multi-layered** system.

New **application framework** for Web developers.

Integrate any **Voice Service** like Google Assistant, Alexa and others.

YodaOS **Development Toolkit**.

The Arch of 7

Rokid Voice Service
(asr, nlp, cloud skills)

Local Rokid Skills

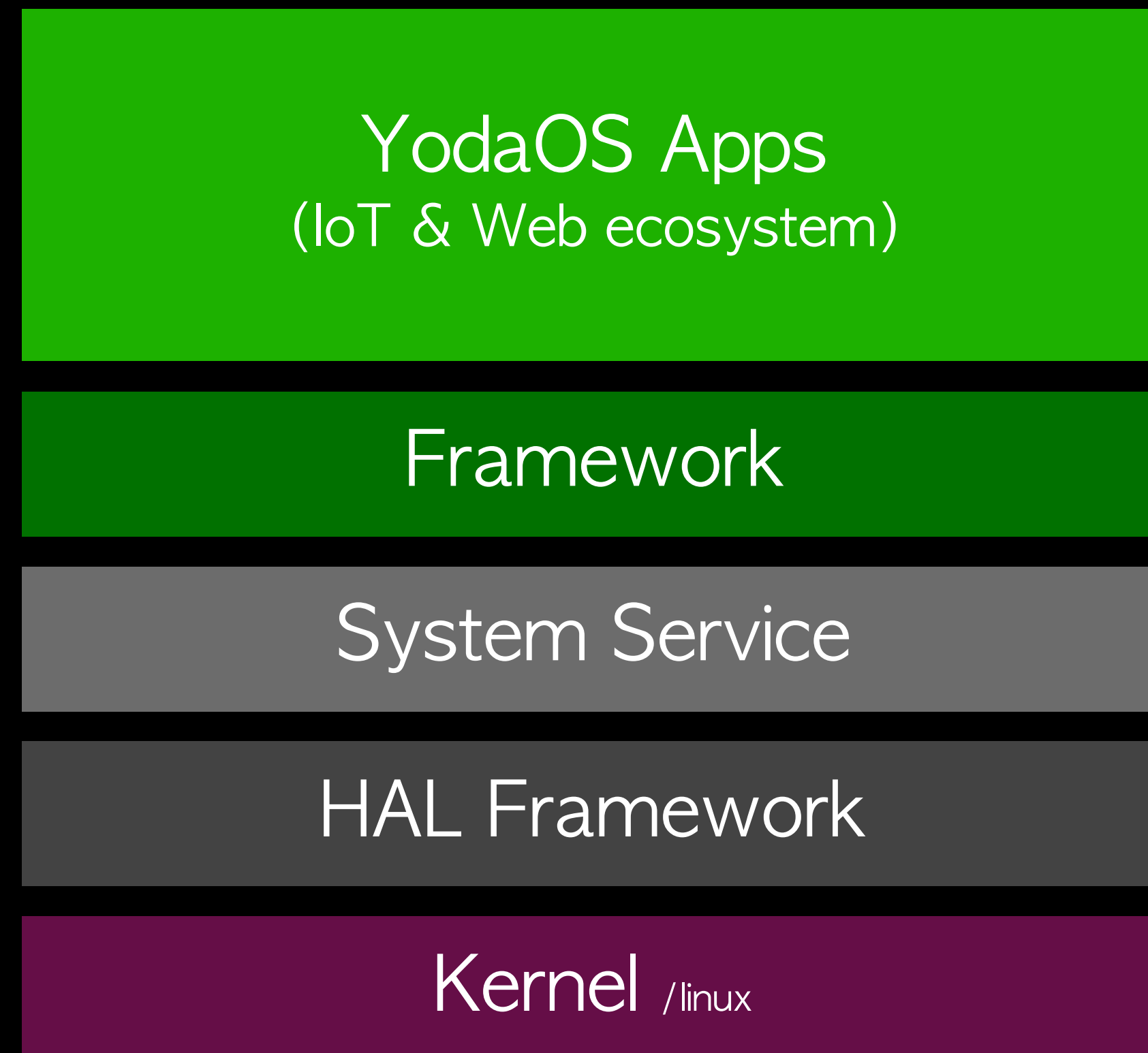
Framework

System Service

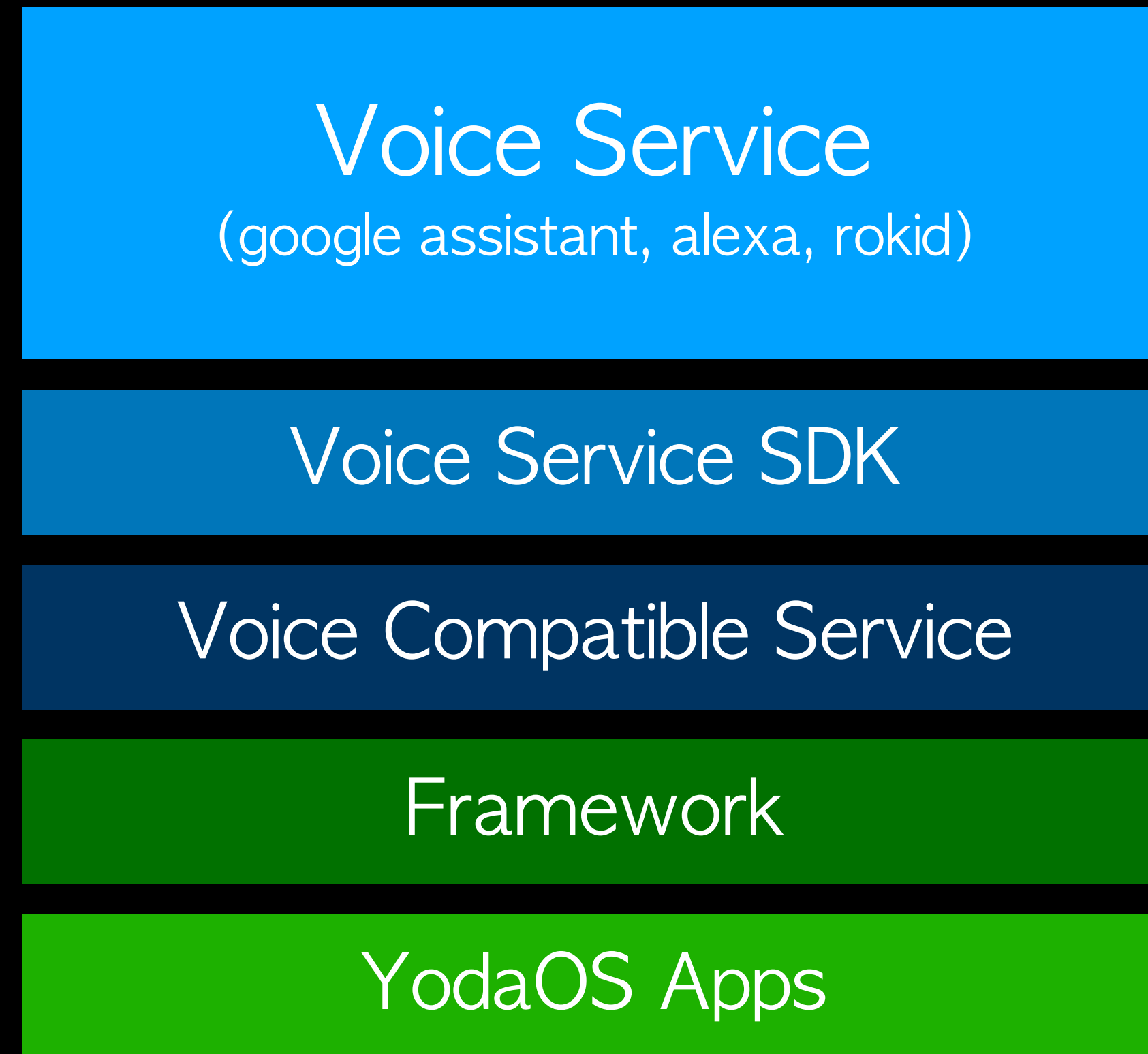
HAL Framework

Kernel /linux

The Arch of 8



The Arch of 8



YodaOS Messaging

Multiplex Messaging

YodaOS URL

YodaOS Event

YodaOS Function

YodaOS URL

yoda-app://setup/init

yoda-app://cloud-player/play

yoda-app://bluetooth/open?discovery=yes

yoda-app://volume/set?value=90


```
Application({  
  url: url => {  
    switch (url.pathname) {  
      case '/set':  
        AudioManager.setVolume(url.query.value)  
        break  
      case '/mute':  
        AudioManager.mute()  
        break  
      Default:  
        speechSynthesis.speak('beep beep')  
        break  
    }  
  }  
})
```

```
yoda-app://volume/set?value=100  
yoda-app://volume/mute
```

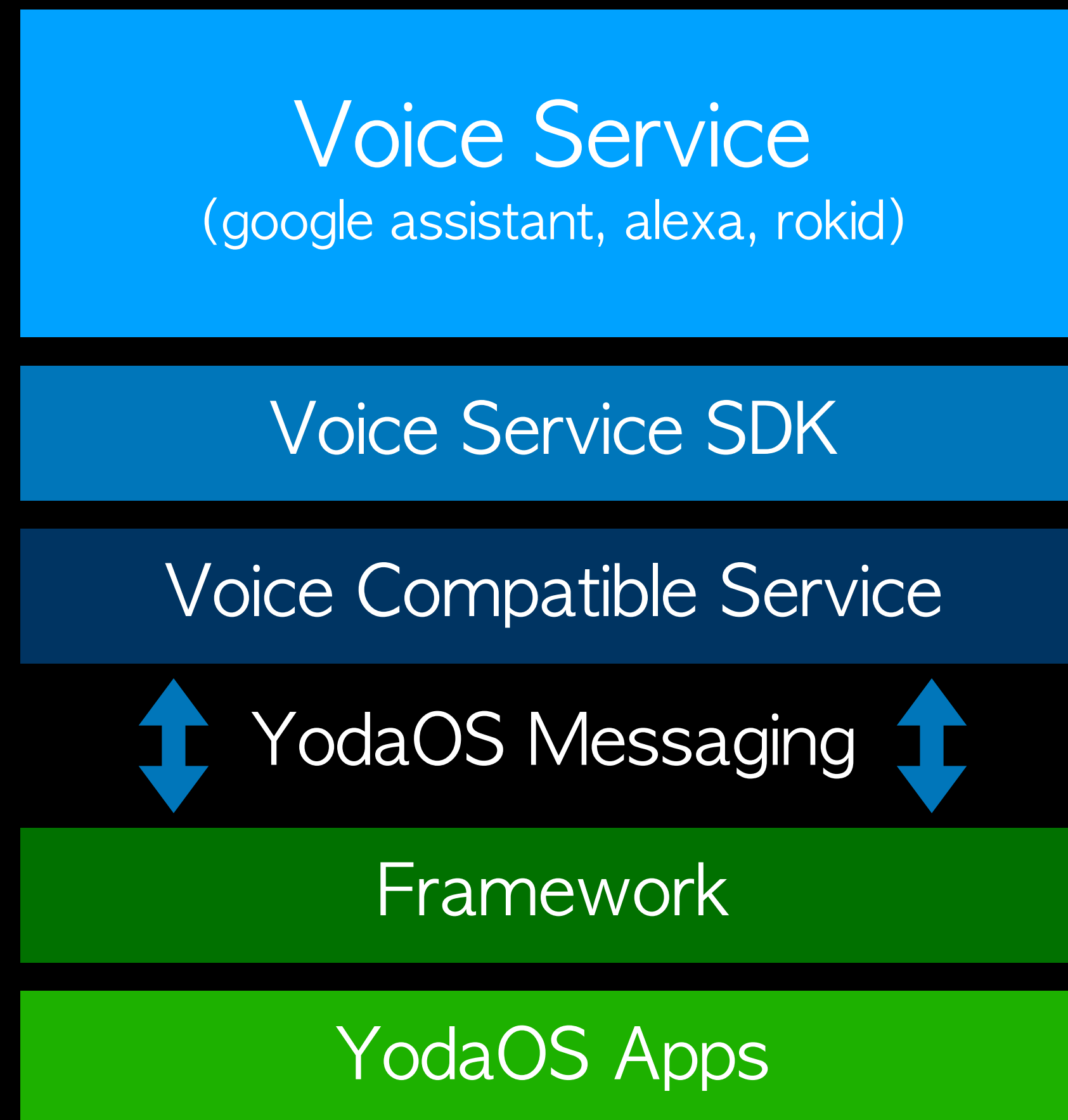
YodaOS Event

```
var target = new EventTarget(  
    'yodaos.apps.cloud-player')  
  
target.on('tts.status', code => {  
    console.log('handling the status code')  
})
```

YodaOS Function

```
var player = new FunctionTarget(  
    'yodaos.apps.cloud-player')  
  
var chan = player.getStreamChannel()  
player.inspect().map(info => {  
    console.log(info.duration, info.position)  
})
```

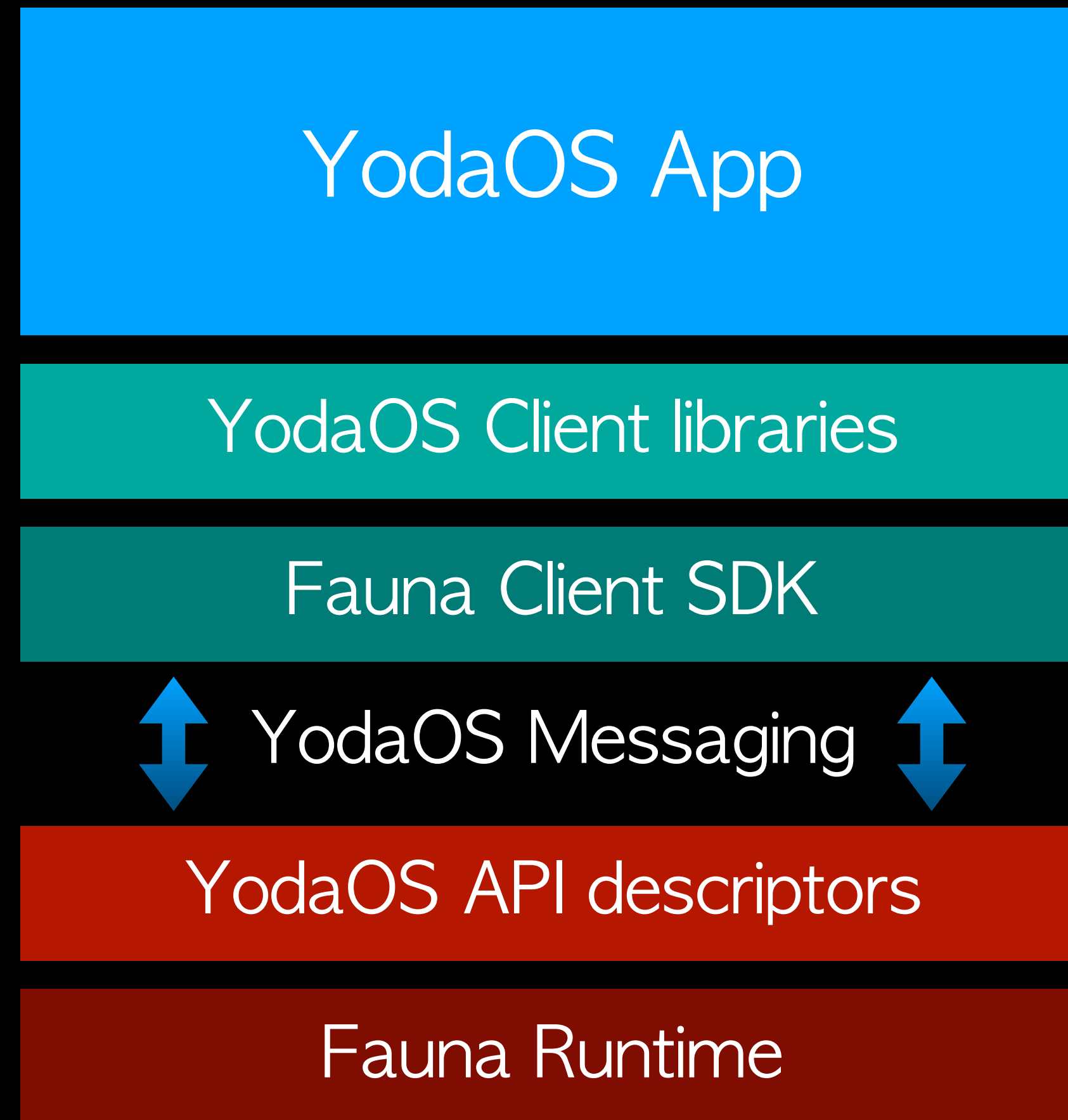
The Arch of 8



Fauna

the self-discovering **API framework** for **YodaOS**

Fauna Framework



Fauna Client

JavaScript: self-discovering YodaOS API at runtime.

C/C++: generating application SDK for applications.

Others? Build a fauna client SDK for your favorite language is welcome.

Fauna Runtime

The open **protocol and docs** to get involved easily

Insides **hive** with improvements of process booting

Supports **in-process & independent** modes

YodaOS App

YodaOS Libraries

@yodaos/application

@yodaos/speech-synthesis

@yodaos/voice-interface

@yodaos/application

Application

AppService

AudioFocus

App Manifest

```
{
  "name": "bluetooth",
  "version": "1.0.1",
  "main": "app.js",
  "manifest": {
    "services": [
      "bluetooth-service"
    ],
    "hosts": [
      "bluetooth"
    ]
  },
  "author": "Morrowind Xie <morrowindxie@gmail.com>",
  "license": "Apache 2.0"
}
```

@yodaos/application AppService

```
var app = Application({
  created: () => {
    logger.debug('app created')
  },
  destroyed: () => {
    logger.debug('app destroyed')
  },
  url: location => {
    this.startService('bluetooth-service')
  }
})
```

@yodaos/application **AudioFocus**

```
var focus = new AudioFocus()
focus.onGain = () => {
    // able to play audio cooperatively
}
focus.onLoss = (transient) => {
    if (transient === true) {
        // should pause your playback because the requesting
        request is transient.
    } else {
        // should stop your playback.
    }
}

// just do others
focus.request()
```


YodaOS Libraries

@yodaos/application

@yodaos/speech-synthesis

@yodaos/voice-interface

Testing YodaOS App

bootstrap **test** environment

mock anything

Testing YodaOS App

```
var mm = require('@yodaos/mm')

var test = mm.test
test = mm.beforeEach(test, t => {
  t.suite = mm.bootstrap()
  t.end()
})
test = mm.afterEach(test, t => {
  t.suite.teardown()
  t.end()
})
```

Testing YodaOS App

```
var AudioFocus = require('@yodaos/application').AudioFocus

test('should speak an audio', t => {
  t.plan(2)

  t.suite.openUrl('yoda-app://system/shutdown')
  t.suite.audioFocus
    .on('gain', focus => {
      t.strictEqual(focus.type, AudioFocus.Type.TRANSIENT)
    })
    .on('loss', focus => {
      t.strictEqual(focus.type, AudioFocus.Type.TRANSIENT)
      t.end()
    })
  })
})
```

Testing YodaOS App

```
var mock = require('@yodaos/mm/mock')

mock.proxyMethod(require('@yoda/manifest'),
  'get', {
    after: (ret, self, args) => {
      if (args[0] === 'capabilities.battery') {
        return false
      }
      return ret
    }
  })
```

YodaOS Development Toolkit

yoda-am application manager toolkit

yoda-pm package manager toolkit

yoda-debug debugger

YodaOS

Community edition

- **Raspberry** with ReSpeaker supported.
- **Kamino 18** Devkit for full-integrated for Voice Interface.
- x86 runtime on QEMU/VirtualBox.

Google Assistant

Inside

Alexa

Inside

Rokid

Inside

Community
for Business?

Community
is pure

