

# CloudKit JS and Web Services

Session 710

Chris Edstrom iCloud Server Engineering

Onar Vikingstad iCloud Web Engineering

# #1

CloudKit Feature Request

# Background

Familiarity with CloudKit

---

Introducing CloudKit

WWDC14

---

Advanced CloudKit

WWDC14

---

# CloudKit Architecture

Fundamental CloudKit objects

# CloudKit Architecture

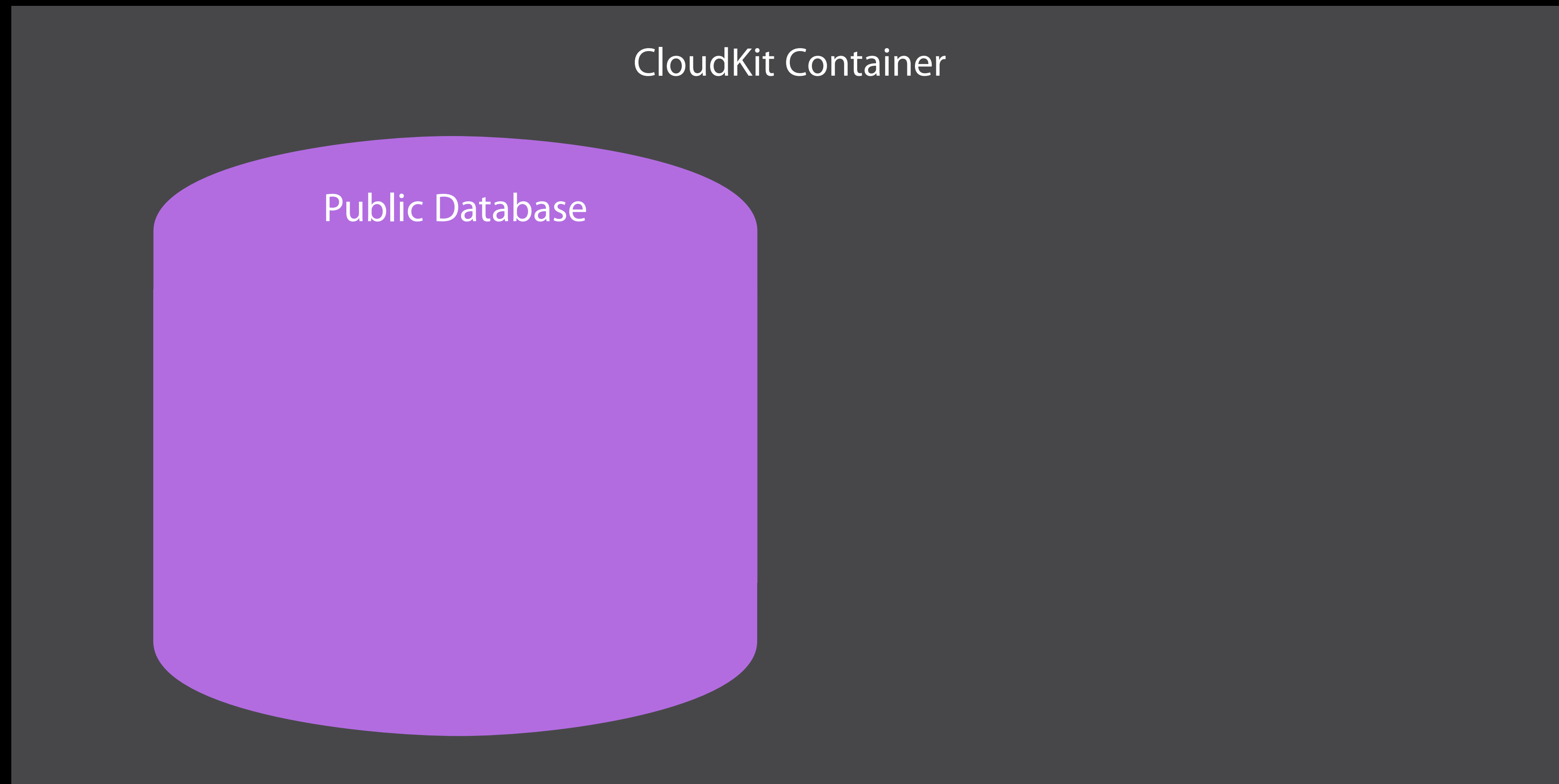
Fundamental CloudKit objects



CloudKit Container

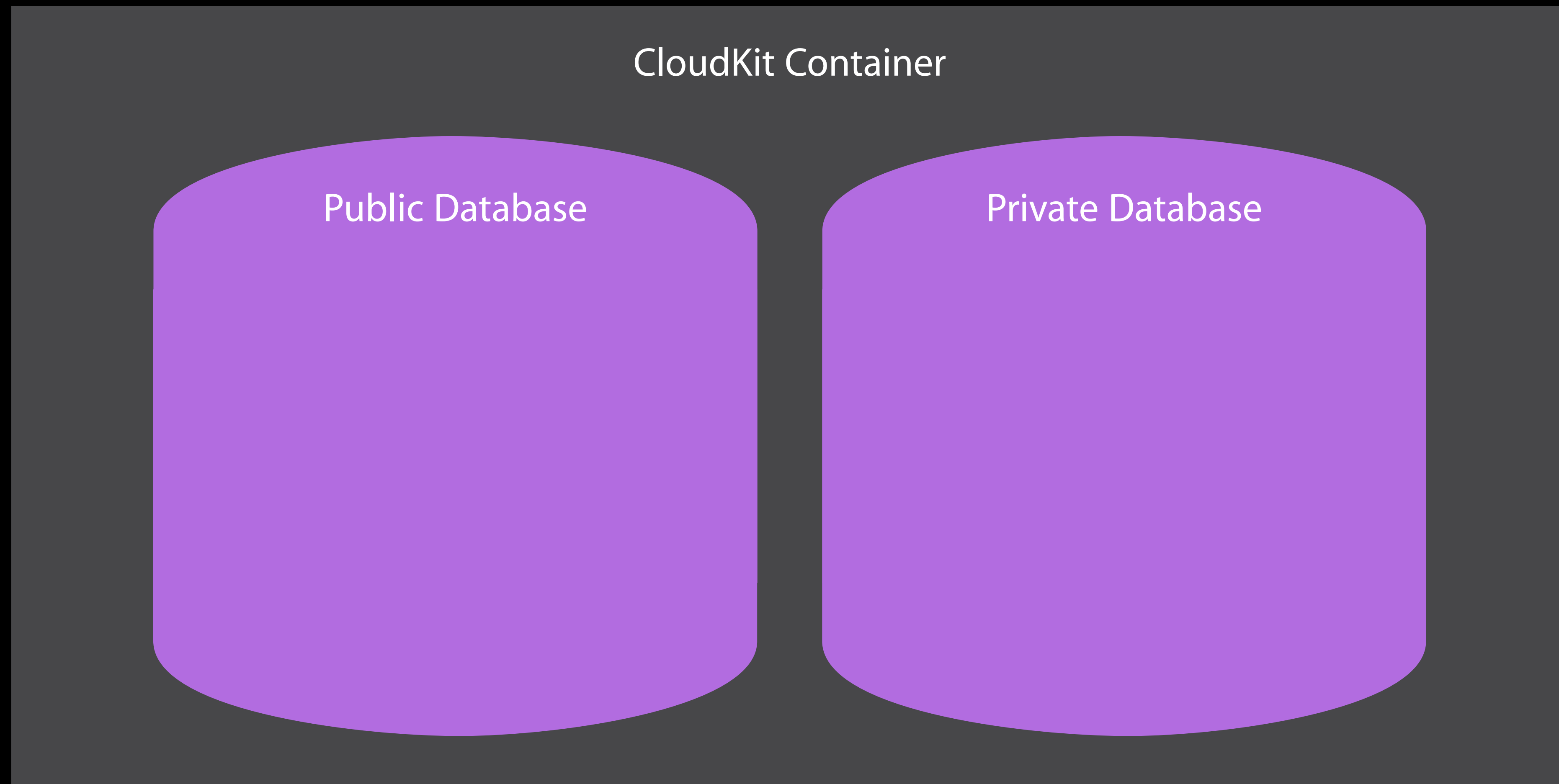
# CloudKit Architecture

## Fundamental CloudKit objects



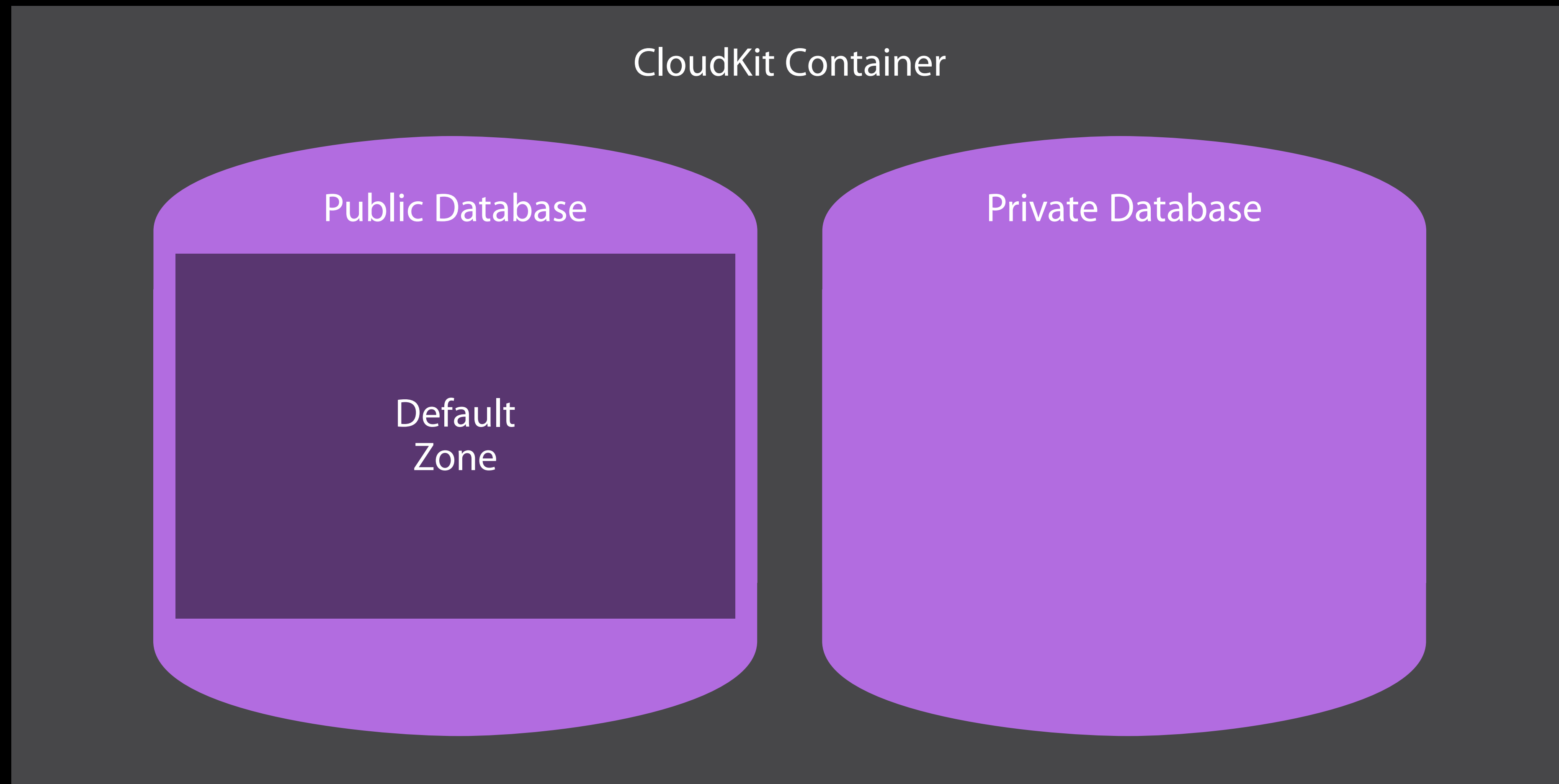
# CloudKit Architecture

## Fundamental CloudKit objects



# CloudKit Architecture

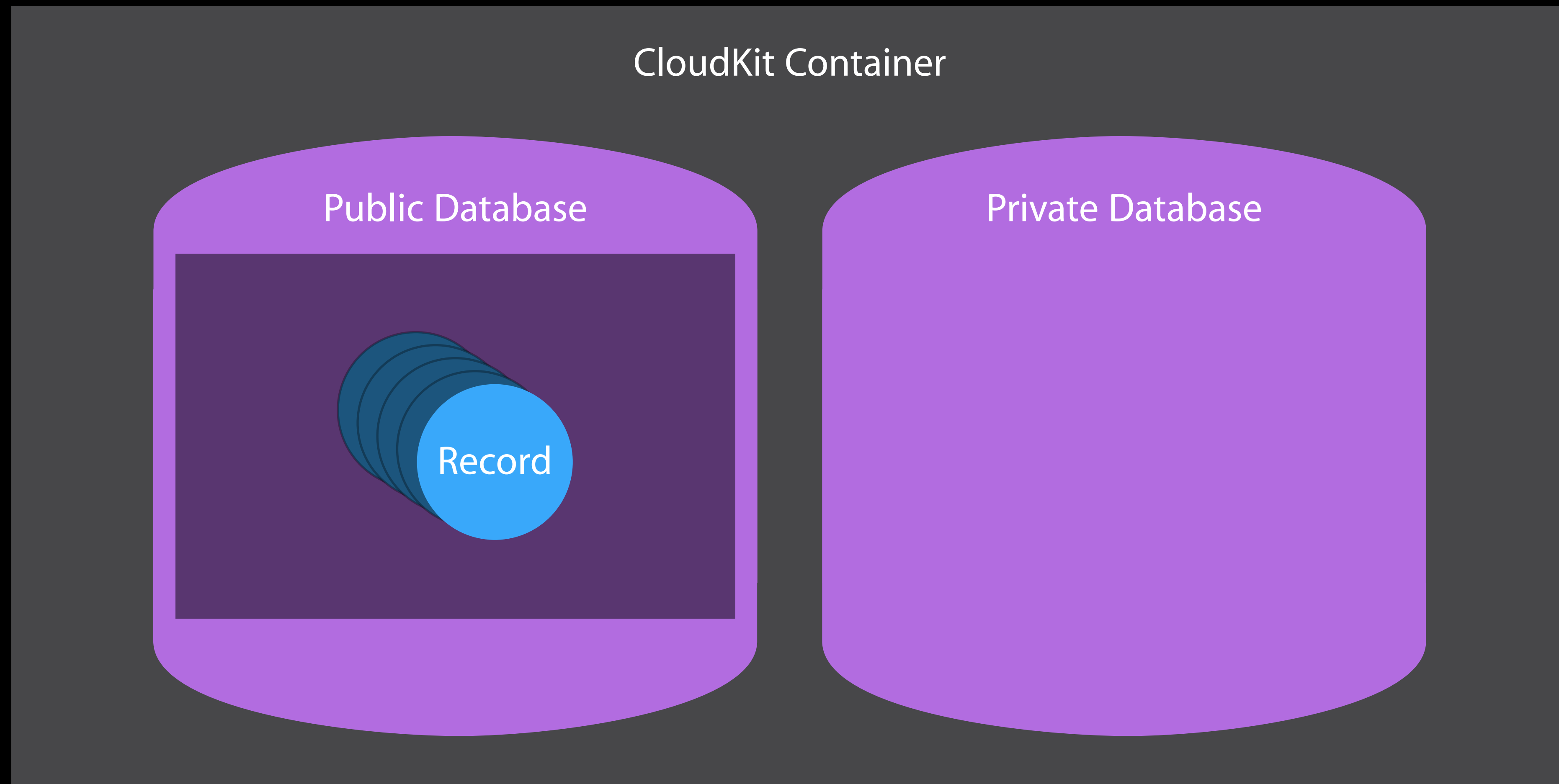
## Fundamental CloudKit objects





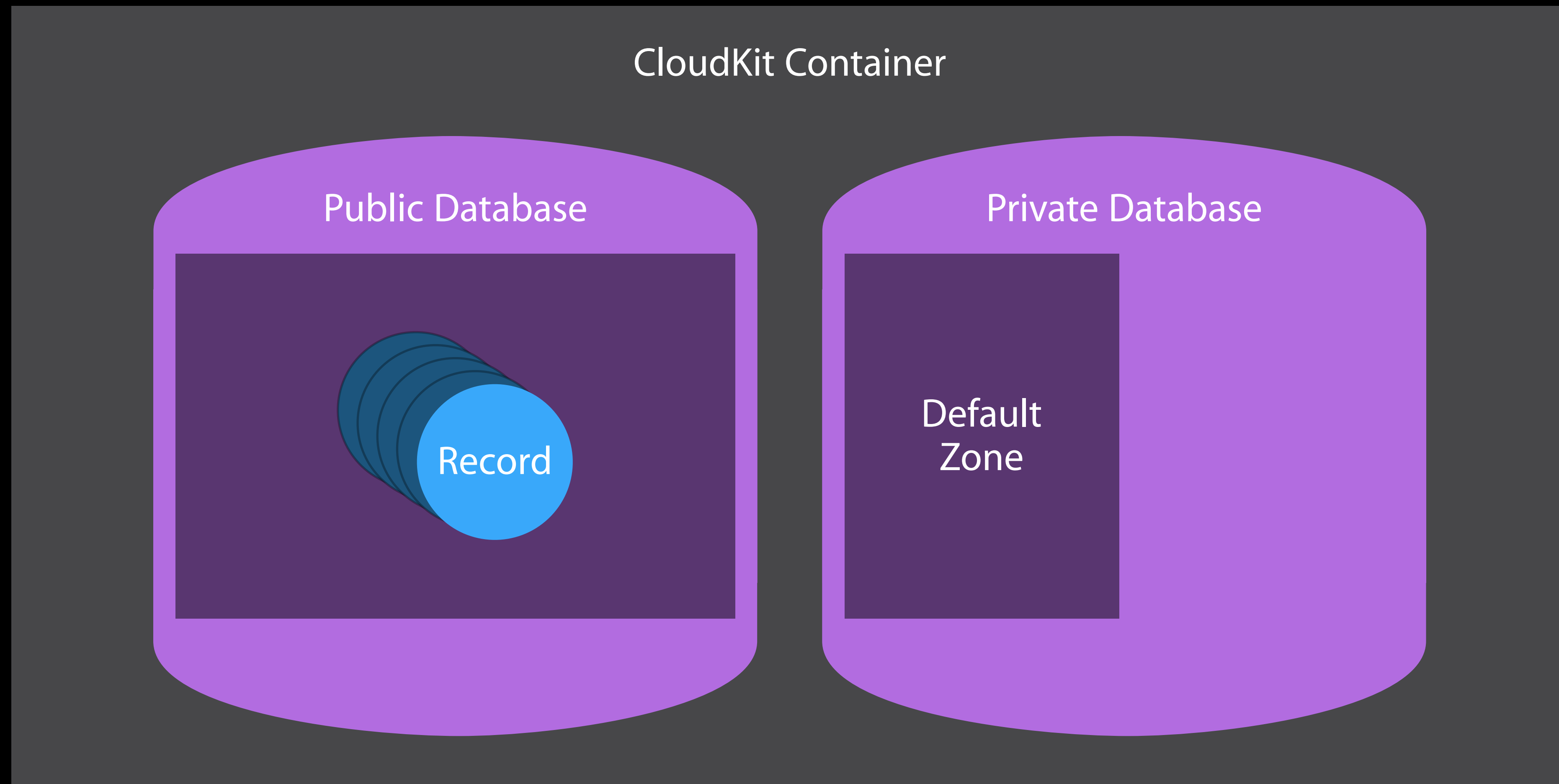
# CloudKit Architecture

## Fundamental CloudKit objects



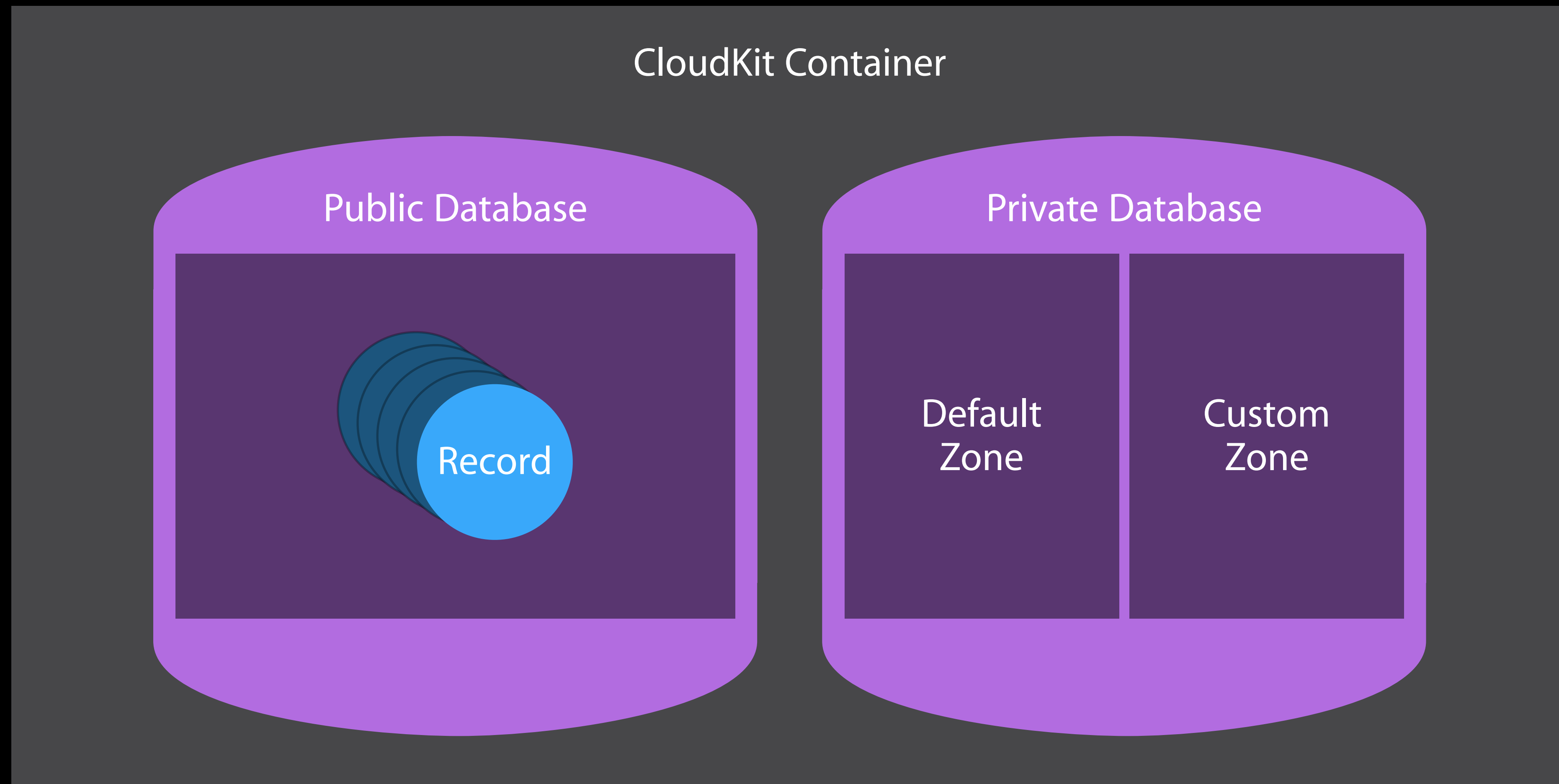
# CloudKit Architecture

## Fundamental CloudKit objects



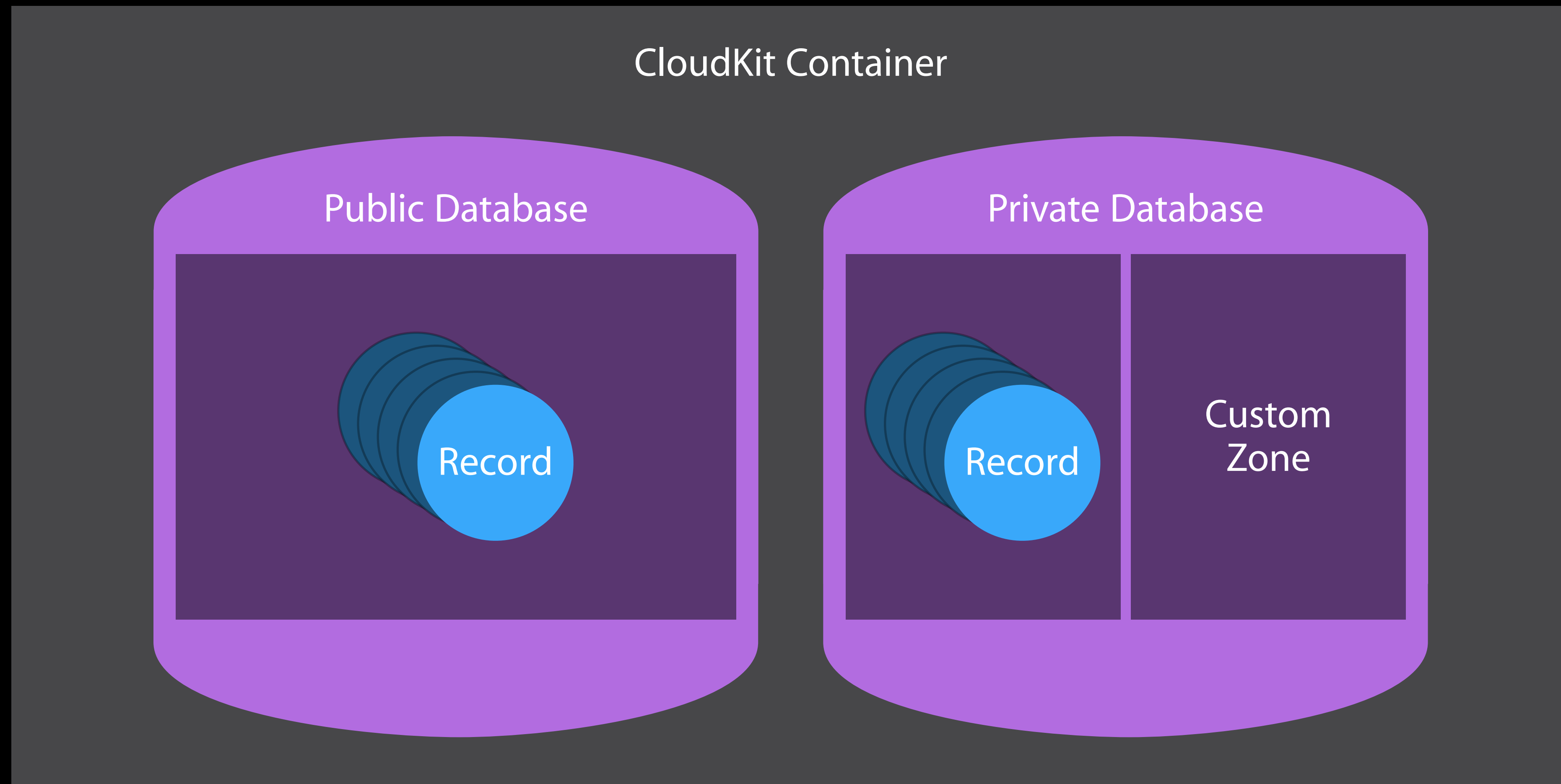
# CloudKit Architecture

## Fundamental CloudKit objects



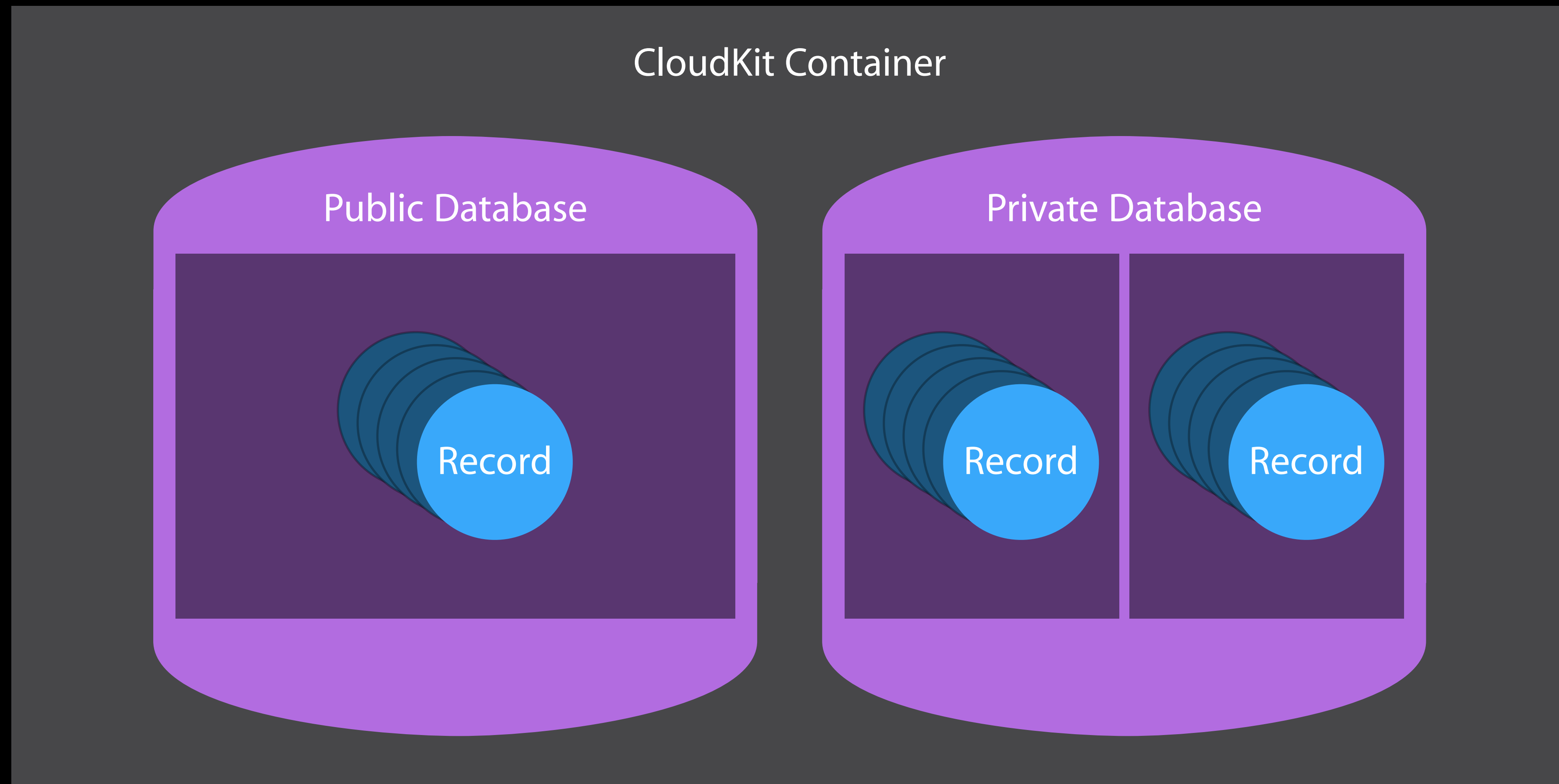
# CloudKit Architecture

## Fundamental CloudKit objects



# CloudKit Architecture

## Fundamental CloudKit objects

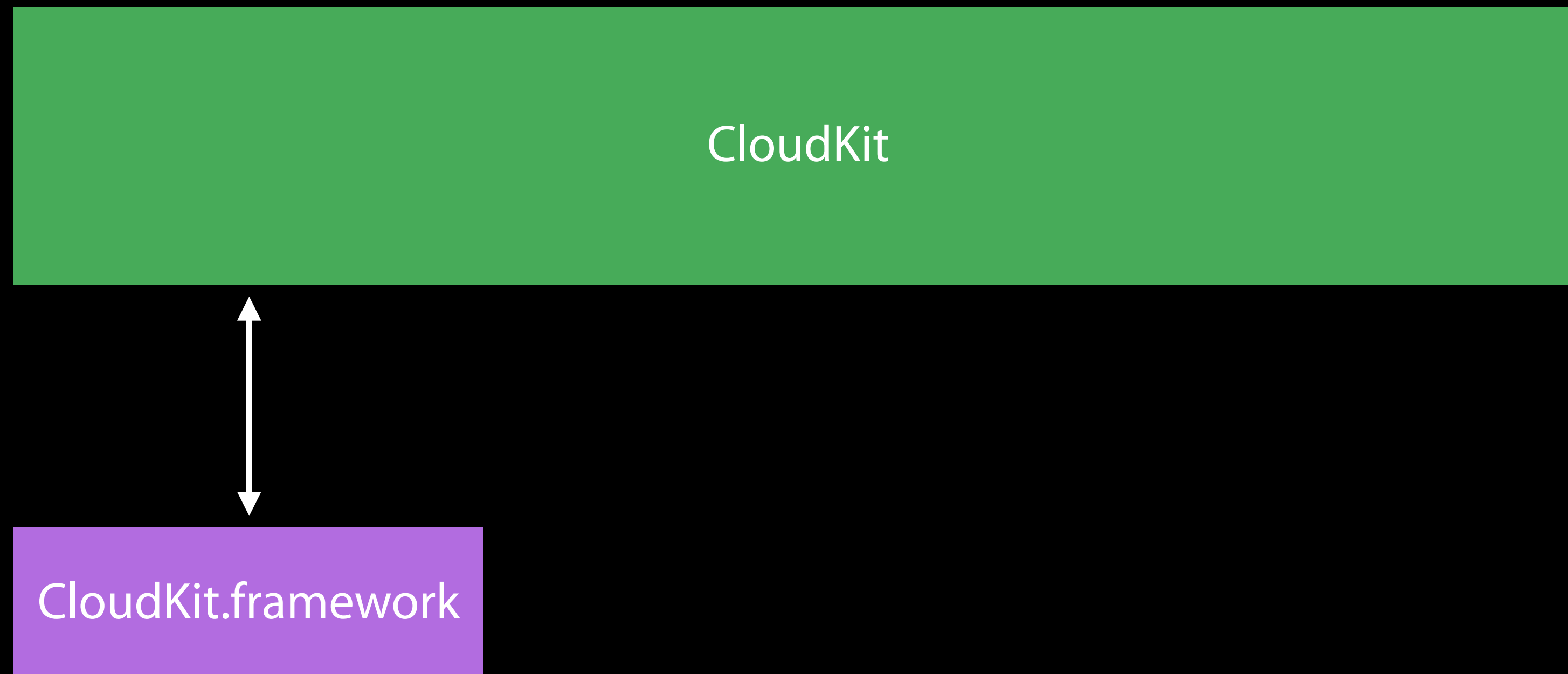


# CloudKit Architecture

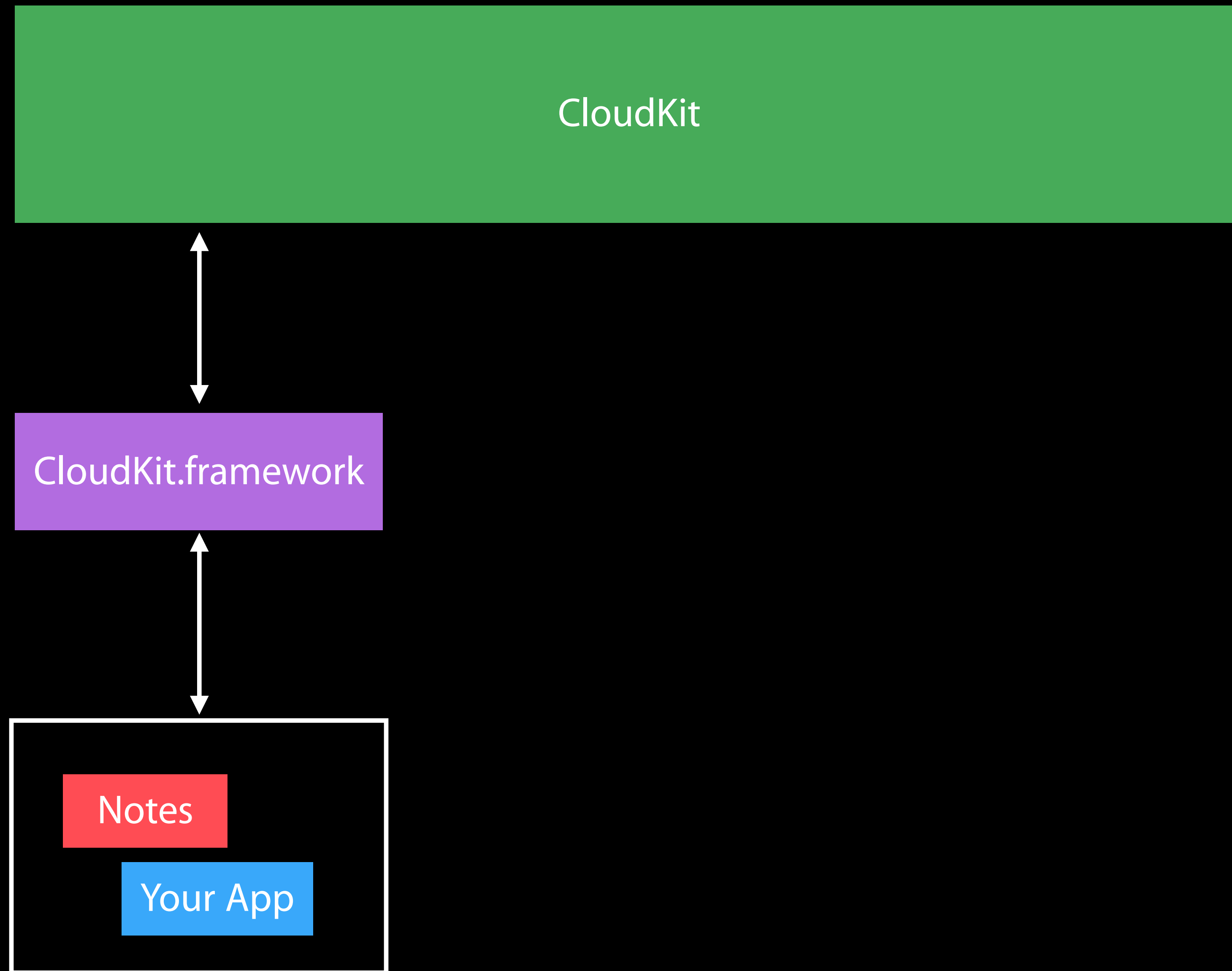


CloudKit

# CloudKit Architecture

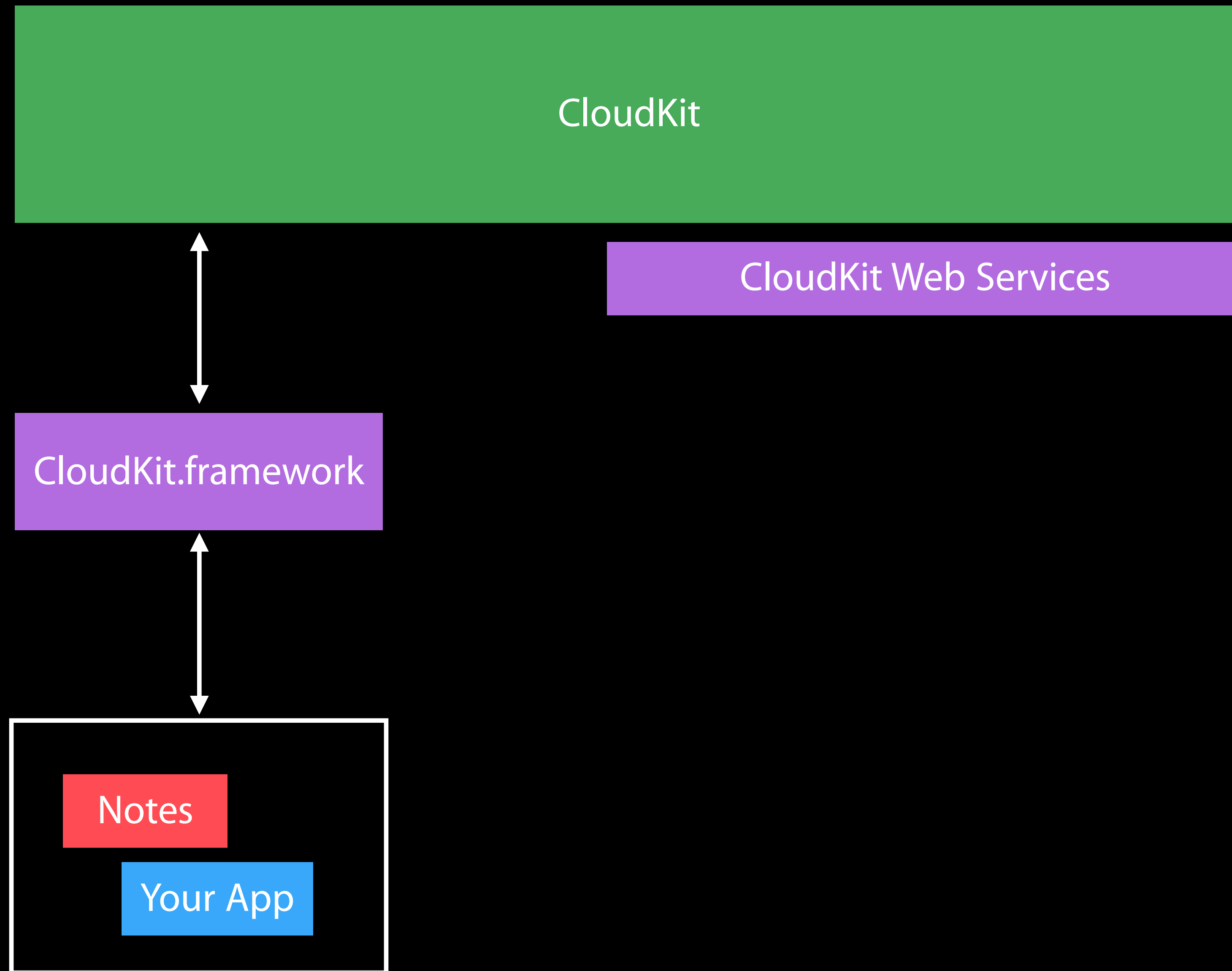


# CloudKit Architecture

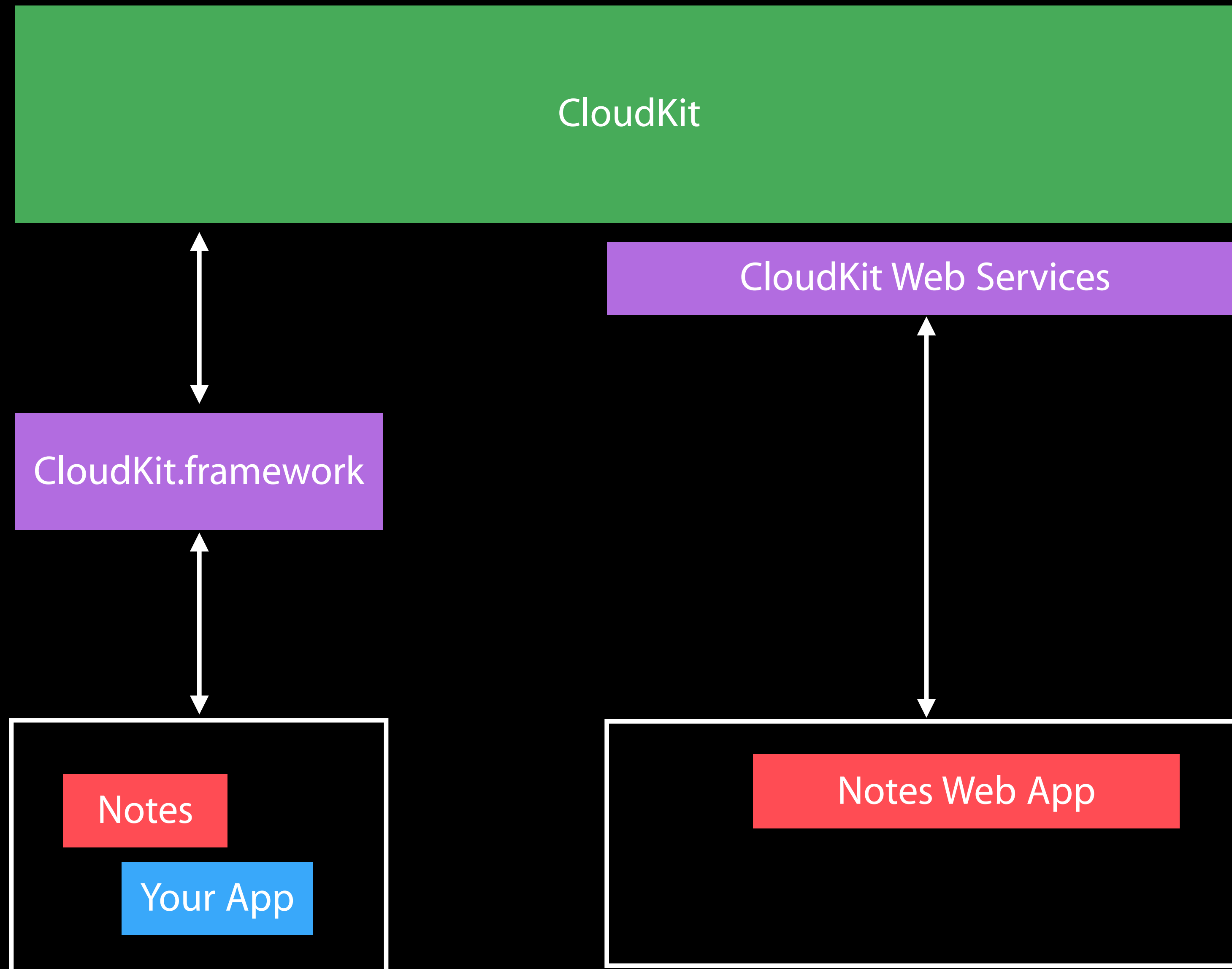




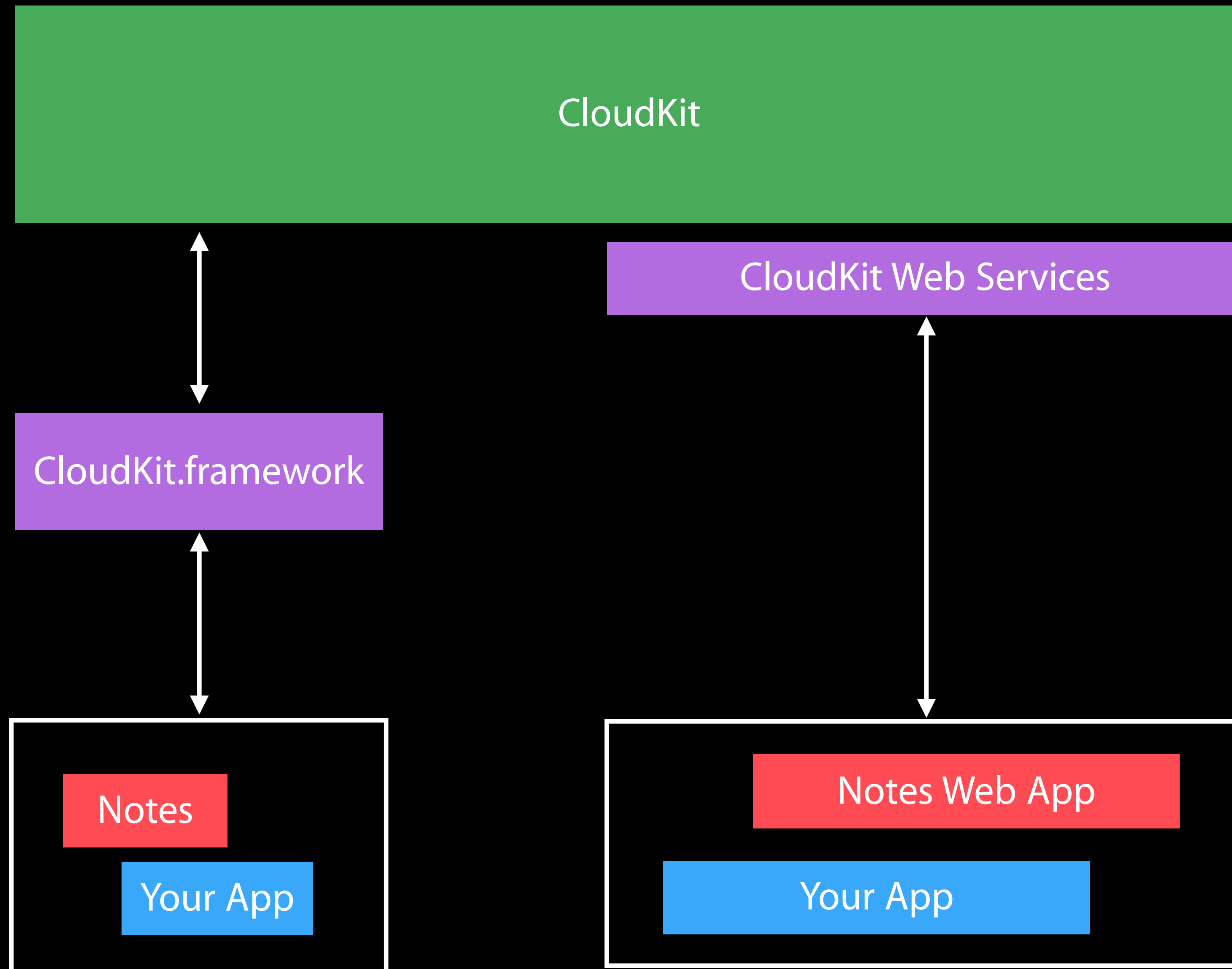
# CloudKit Architecture



# CloudKit Architecture



# CloudKit Architecture



# What Is It?

A JSON/HTTPS interface to CloudKit

Web sign-in with Apple ID

A JavaScript library

# Features

# Features

Public/private database access

# Features

Public/private database access

Create/read/update/delete

# Features

Public/private database access

Create/read/update/delete

Assets



# Features

Public/private database access

Create/read/update/delete

Assets

Query

# Features

Public/private database access

Create/read/update/delete

Assets

Query

Subscriptions and notifications

# Features

Public/private database access

Create/read/update/delete

Assets

Query

Subscriptions and notifications

User discoverability

# Features

Public/private database access

Create/read/update/delete

Assets

Query

Subscriptions and notifications

User discoverability

Sync

# Features

Public/private database access

Create/read/update/delete

Assets

Query

Subscriptions and notifications

User discoverability

Sync

Authentication

Feature Parity!

# Feature Parity

All operations are available in the JSON API

Field names

Completions via JS promises

Code ends up looking similar

# Getting Started

Create a container

Create a schema

Container can be created from either Portal or Xcode

Schema can be created from Dashboard or on-demand by your app



# Enable Web Access

Generate a web service API token

Set your login callback mechanism

Set domain restrictions

CloudKitCatalog

SCHEMA

Record Types

Security Roles

Subscription Types

PUBLIC DATA

User Records

Default Zone

UsageBeta

PRIVATE DATA

Default Zone

ADMIN

API TokensBeta

Deployment

Environment: DEVELOPMENT

API Tokens

Sort by Name

Code Tutorial

Created: May 12 2015 11:12 AM

Code Tutorial

Created:May 12 2015 11:12 AM

Modified:May 29 2015 2:59 PM

Token: 1c6d0ac40cf685931a021359dd217680182692e958d5c95248e36ee44696cd5d

Login Callback

postMessage

https://Custom URL

Allowed Origins

Any domain

Specific domains:

https://Add Domain...

Notes

Created:

May 12 2015 11:12 AM

Modified:

May 29 2015 2:59 PM

Token: 1c6d0ac40cf685931a021359dd217680182692e958d5c95248e36ee44696cd5d

Login Callback

☒ postMessage

☐ https:// | v Custom URL

Allowed Origins

☒ Any domain

☐ Specific domains:

https:// Add Domain...

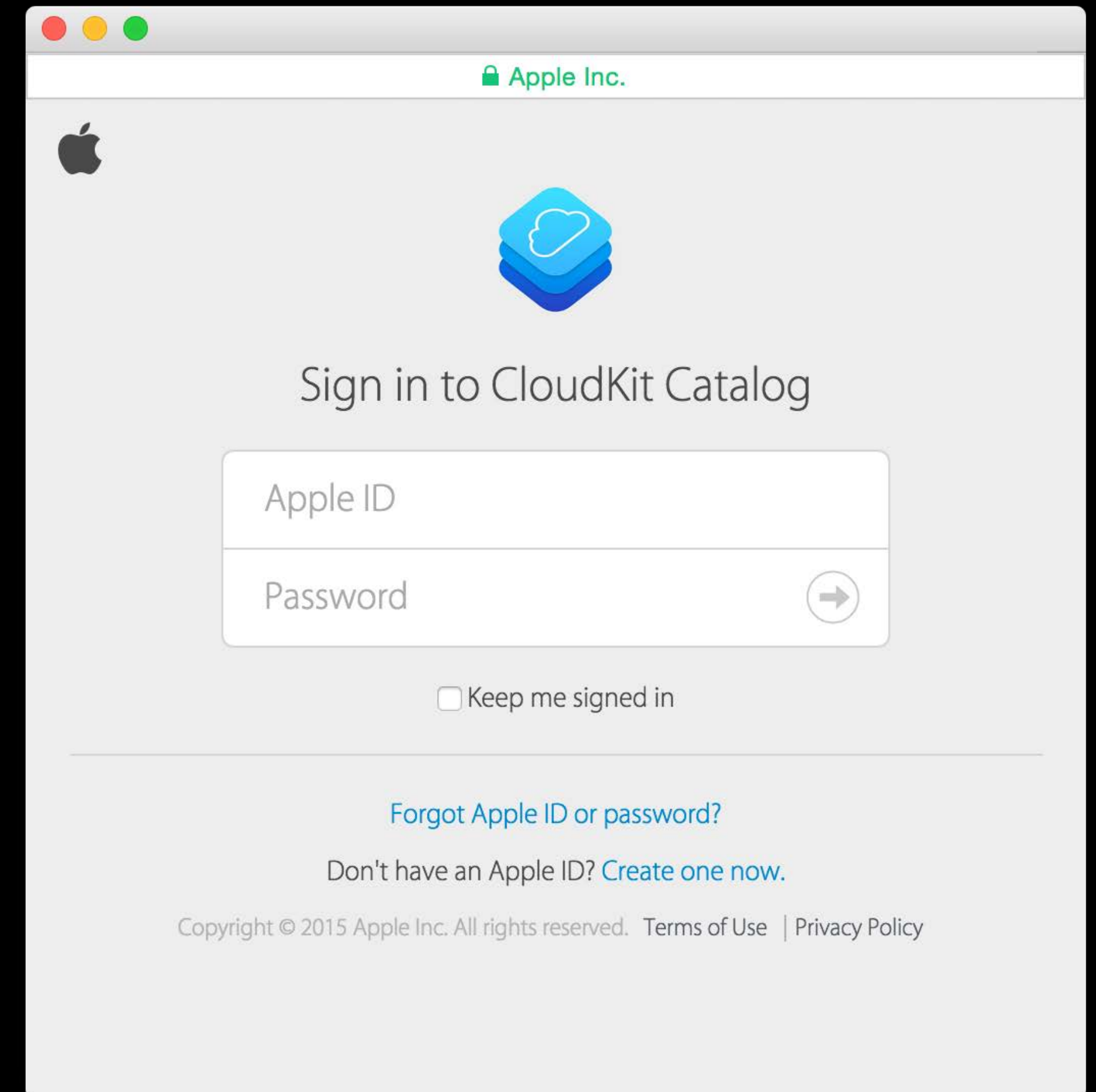
Notes

# Authentication

Redirect URL provided when authentication is needed

Users authenticate with an Apple ID

Username and password are not seen by your app



# JSON API Example

Container name is *com.example.tasks*

Private database use case

```
Tasks: {  
  name: STRING,  
  priority: INT64  
}
```

# JSON API Example

Container name is *com.example.tasks*

Private database use case

```
Tasks: {  
  name: STRING,  
  priority: INT64  
}
```

# JSON API Example

## Request

POST <https://api.apple-cloudkit.com/database/1/com.example.tasks/-production/private/records/modify>

```
{
  "operations" : [{
    "operationType" : "create",
    "record" : {
      "recordType" : "Tasks",
      "fields" : { "name": { "value" : "Buy Milk" }},
      "recordName" : "task-1"
    }
  }]
}
```

# JSON API Example

## Request

```
POST https://api.apple-cloudkit.com/database/1/com.example.tasks/-  
production/private/records/modify
```

```
{  
  "operations" : [{  
    "operationType" : "create",  
    "record" : {  
      "recordType" : "Tasks",  
      "fields" : { "name": { "value" : "Buy Milk" }},  
      "recordName" : "task-1"  
    }  
  }]  
}
```



# JSON API Example

## Request

POST <https://api.apple-cloudkit.com/database/1/com.example.tasks/-production/private/records/modify>

```
{
  "operations" : [{
    "operationType" : "create",
    "record" : {
      "recordType" : "Tasks",
      "fields" : { "name": { "value" : "Buy Milk" }},
      "recordName" : "task-1"
    }
  }]
}
```

# JSON API Example

Unauthorized

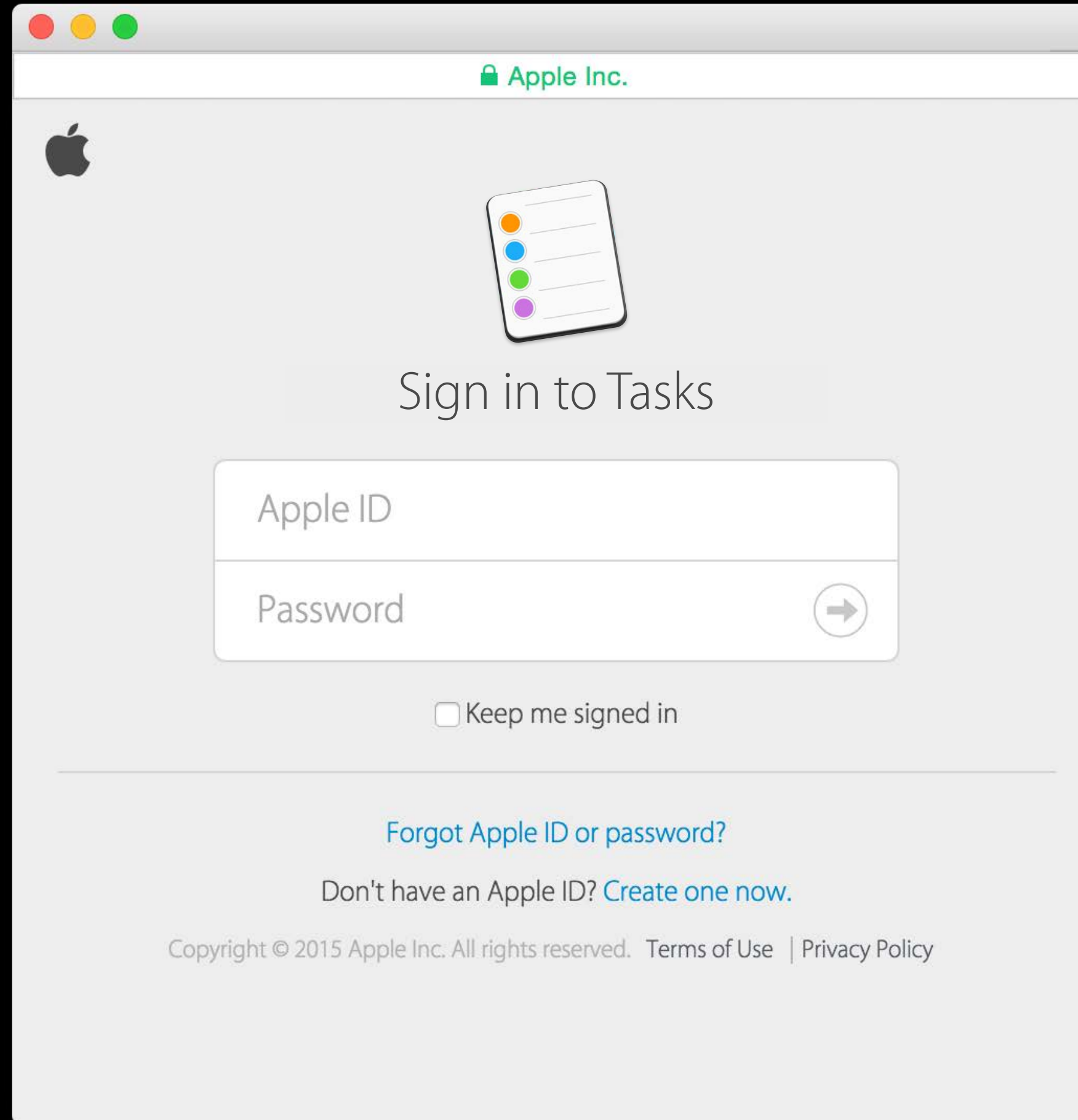
```
{  
  "uuid": "4f02f7aa-fbb5-4cf8ae8e-4dd463793841",  
  "serverErrorCode": "AUTHENTICATION_REQUIRED",  
  "reason": "request needs authorization",  
  "redirectUrl": "https://signin.apple.com/IDMSWebAuth/auth2?..."  
}
```

# JSON API Example


Unauthorized


```
{  
  "uuid": "4f02f7aa-fbb5-4cf8ae8e-4dd463793841",  
  "serverErrorCode": "AUTHENTICATION_REQUIRED",  
  "reason": "request needs authorization",  
  "redirectUrl": "https://signin.apple.com/IDMSWebAuth/auth2?..."  
}
```

# JSON API Example




Apple Inc.





Sign in to Tasks

Apple ID

Password 

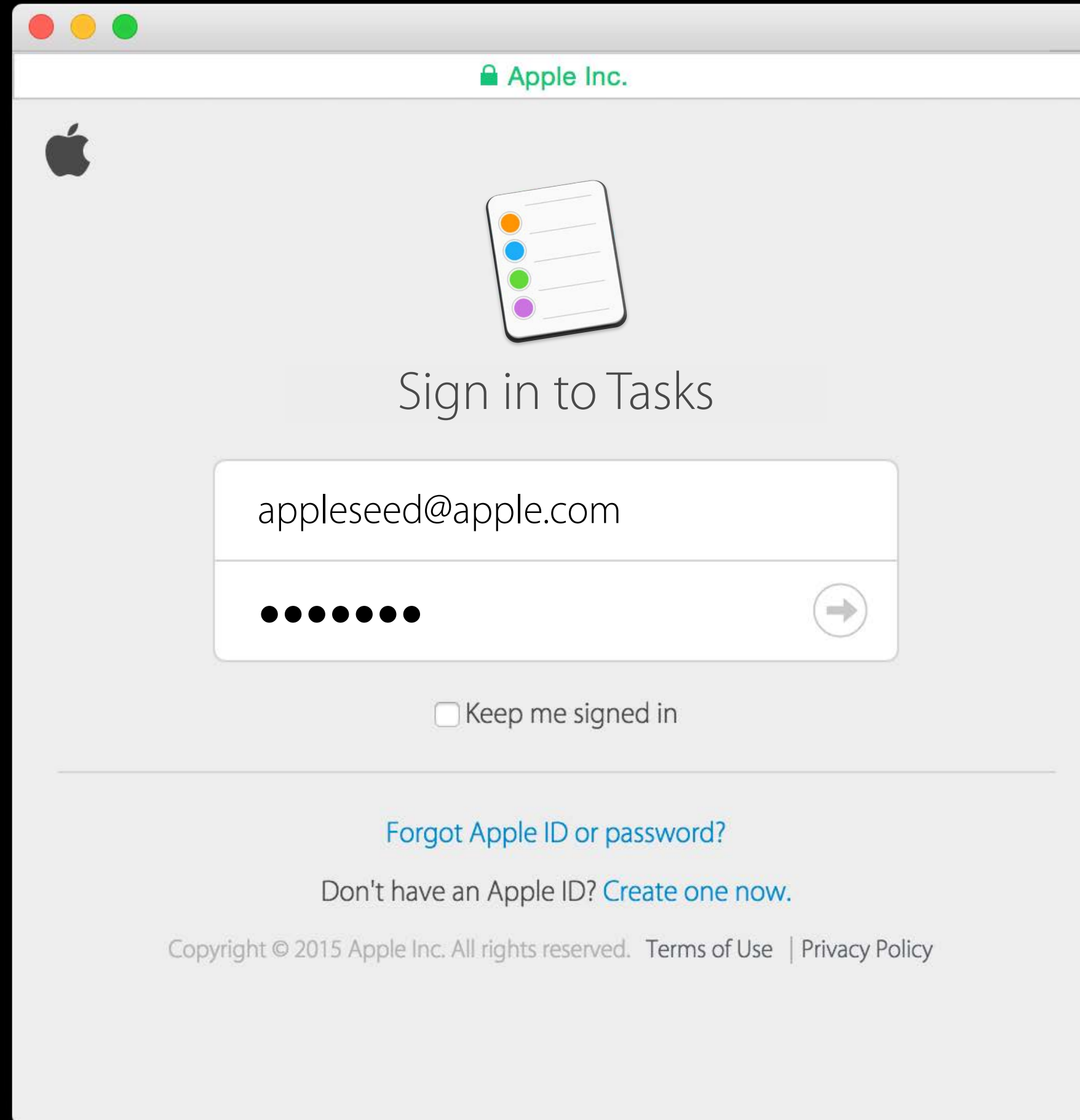
☐ Keep me signed in

[Forgot Apple ID or password?](#)

Don't have an Apple ID? [Create one now.](#)

Copyright © 2015 Apple Inc. All rights reserved. [Terms of Use](#) | [Privacy Policy](#)

# JSON API Example

A screenshot of a web browser window displaying the Apple Tasks sign-in page. The browser's address bar shows "Apple Inc." with a green lock icon. The page features the Apple logo in the top left corner. In the center, there is an icon of a notepad with four colored dots (orange, blue, green, purple). Below the icon, the text "Sign in to Tasks" is displayed. A sign-in form contains an email field with "appleseed@apple.com" and a password field with seven black dots. A right-pointing arrow button is located to the right of the password field. Below the form, there is a checkbox labeled "Keep me signed in". At the bottom, there is a link "Forgot Apple ID or password?" and a text line "Don't have an Apple ID? [Create one now.](#)". The footer contains the text "Copyright © 2015 Apple Inc. All rights reserved. [Terms of Use](#) | [Privacy Policy](#)".

Apple Inc.

Sign in to Tasks

appleseed@apple.com

●●●●●●●

☐ Keep me signed in

[Forgot Apple ID or password?](#)

Don't have an Apple ID? [Create one now.](#)

Copyright © 2015 Apple Inc. All rights reserved. [Terms of Use](#) | [Privacy Policy](#)

# JSON API Example

## Request

POST [https://api.apple-cloudkit.com/database/1/com.example.tasks/-production/private/records/modify?ckSession=1\\_2\\_AQV...](https://api.apple-cloudkit.com/database/1/com.example.tasks/-production/private/records/modify?ckSession=1_2_AQV...)

```
{
  "operations" : [{
    "operationType" : "create",
    "record" : {
      "recordType" : "Tasks",
      "fields" : { "name": { "value" : "Buy Milk" }},
      "recordName" : "task-1"
    }
  }]
}
```

# JSON API Example

## Response

```
{
  "records" : [{
    "recordName" : "task-1",
    "recordChangeTag" : "1e",
    "fields" : { ... },
    "created" : {
      "timestamp" : 1422312944104,
      "userRecordName" : "_9a065b60313a5937e75e03ed8e8f383d",
      "deviceId" : "_2"
    },
    "modified" : {
      ...
    }
  }
]
```

# JSON API Example

## Response

```
{
  "records" : [{
    "recordName" : "task-1",
    "recordChangeTag" : "1e",
    "fields" : { ... },
    "created" : {
      "timestamp" : 1422312944104,
      "userRecordName" : "_9a065b60313a5937e75e03ed8e8f383d",
      "deviceId" : "_2"
    },
    "modified" : {
      ...
    }
  ]
}
```



# CloudKit JS

Onar Vikingstad

# What Is CloudKit JS?

The image shows the JavaScript logo, which consists of the letters 'JS' in a bold, dark gray, sans-serif font. The logo is centered within a solid yellow square. The entire graphic is positioned on the right side of a black background.

JS

# What Is CloudKit JS?

A JavaScript library for using CloudKit on the web

The image shows the JavaScript logo, which consists of the letters 'JS' in a bold, dark gray, sans-serif font. The logo is centered within a solid yellow square. The background of the entire slide is black.

# What Is CloudKit JS?

A JavaScript library for using CloudKit  
on the web

Low-level transport API

The image shows the JavaScript logo, which consists of the letters 'JS' in a bold, dark gray, sans-serif font. The logo is centered within a solid yellow square. The background of the entire slide is black.

# What Is CloudKit JS?

A JavaScript library for using CloudKit on the web

Low-level transport API

Easy transition from CloudKit Framework

The image shows the JavaScript logo, which consists of the letters 'JS' in a bold, dark grey font, centered within a solid yellow square.

# What Is CloudKit JS?

A JavaScript library for using CloudKit on the web

Low-level transport API

Easy transition from CloudKit Framework

Natural to web developers

The image shows the JavaScript logo, which consists of the letters 'JS' in a bold, dark gray, sans-serif font. The logo is centered within a solid yellow square. The yellow square is positioned on the right side of the slide, to the right of the list of features.

# Browser Support

Safari

Firefox

Chrome

Internet Explorer

Microsoft Edge

# *Demo*

Let's check out CloudKit JS in action









<

>

icloud.com

iCloud Notes

Kitchen remodel ideas

15 minutes ago Modern kitchen desi...

Carson's birthday party

1:19 PM Call party supply store

Reference letter for Lee

1:19 PM Worked on the same team for 5 ye...

Contractor meeting

1:19 PM Gary says the inspector will send th...

Tropical vacation ideas

1:19 PM Na Pall properties

Miami conference notes

1:19 PM Sales in emerging markets up

Landscaping

1:19 PM Look into flagstone for backyard ou...

NYC dining greatest hits

1:19 PM Get reservations to the best restaur...

Team-building activities

1:19 PM Paintball tournament

Story idea

1:19 PM Circus clowns rob bank

Aa

John Appleseed

June 5, 2015 at 5:05 PM

Kitchen remodel ideas

Modern kitchen design in a small space. Should have an island. Plenty of storage. Need to research colors, materials, and appliances (stainless steel).

Things to do

Create a budget

Price appliances

Consult with a contractor

# Getting Started

# Configuration

Embedding CloudKit JS on your website

# Configuration

Embedding CloudKit JS on your website

```
<script src="https://cdn.apple-cloudkit.com/ck/1/cloudkit.js">
```



# Configuration

## Embedding CloudKit JS on your website

```
<script src="https://cdn.apple-cloudkit.com/ck/1/cloudkit.js">
```

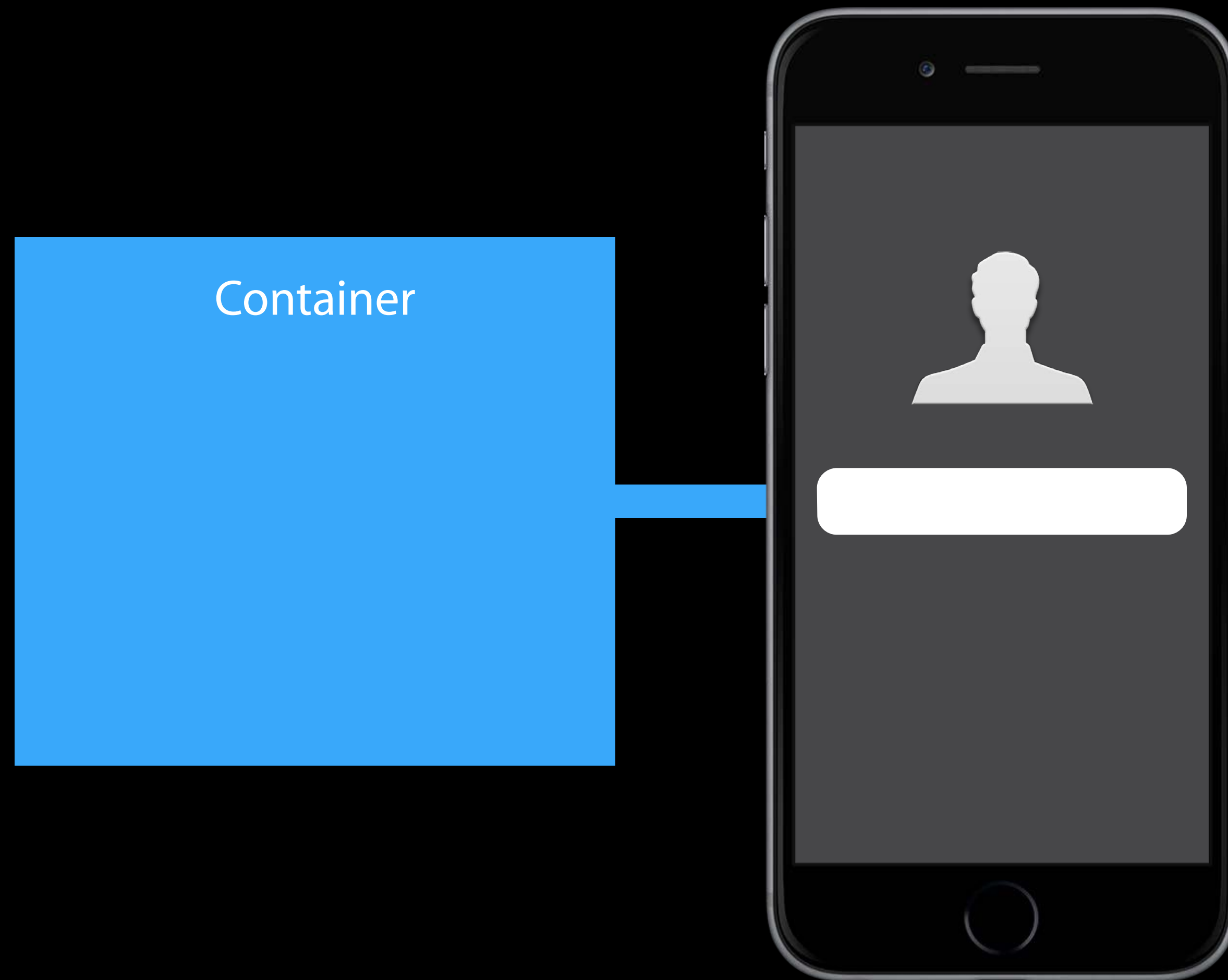
```
<script>  
  CloudKit.configure({  
    containers: [{  
      containerIdentifier: 'com.example.tasks',  
      environment: 'production',  
      apiToken: '<apiToken>'  
    }]  
  });  
</script>
```

# Authentication



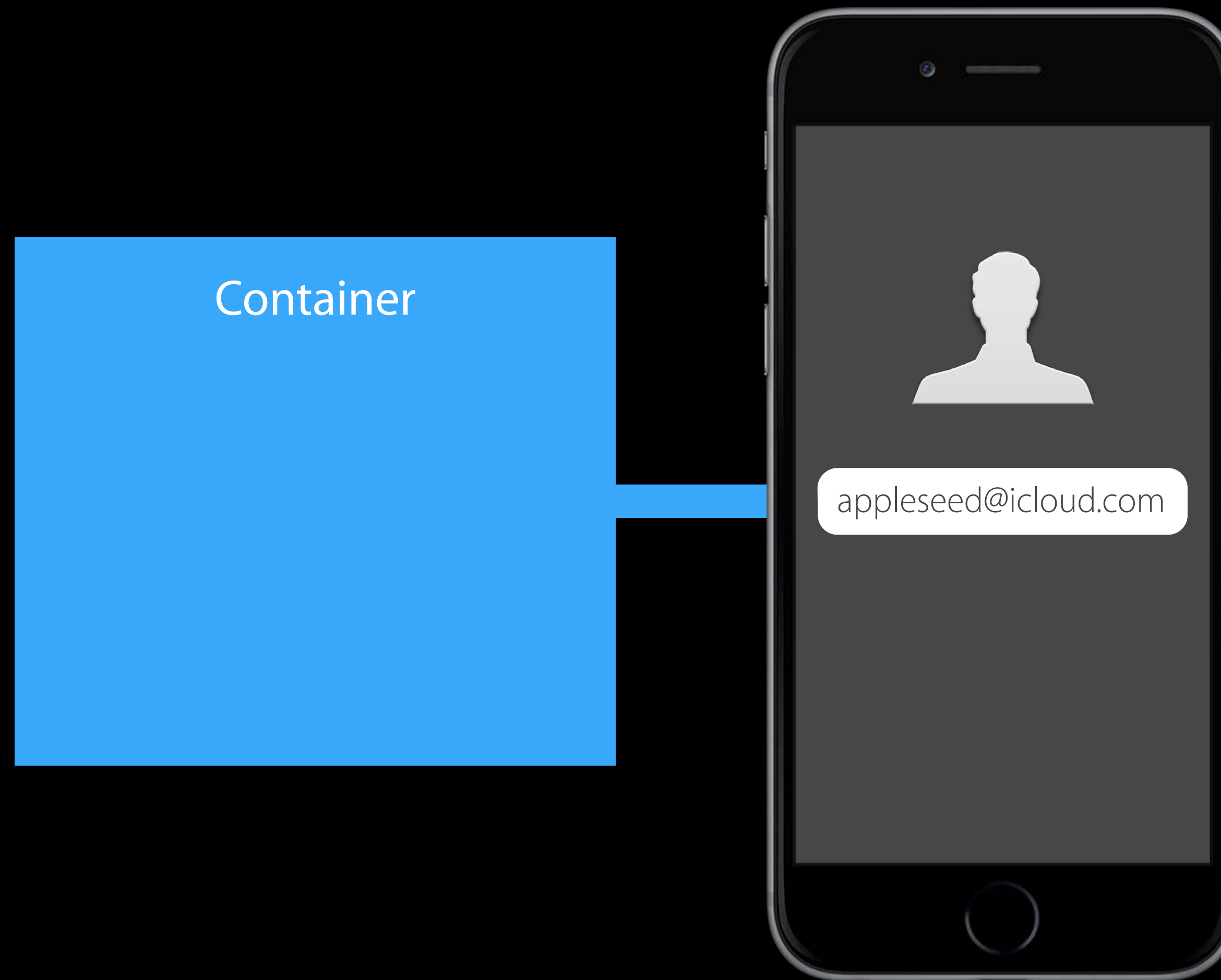
# Authentication

With iOS and OS X



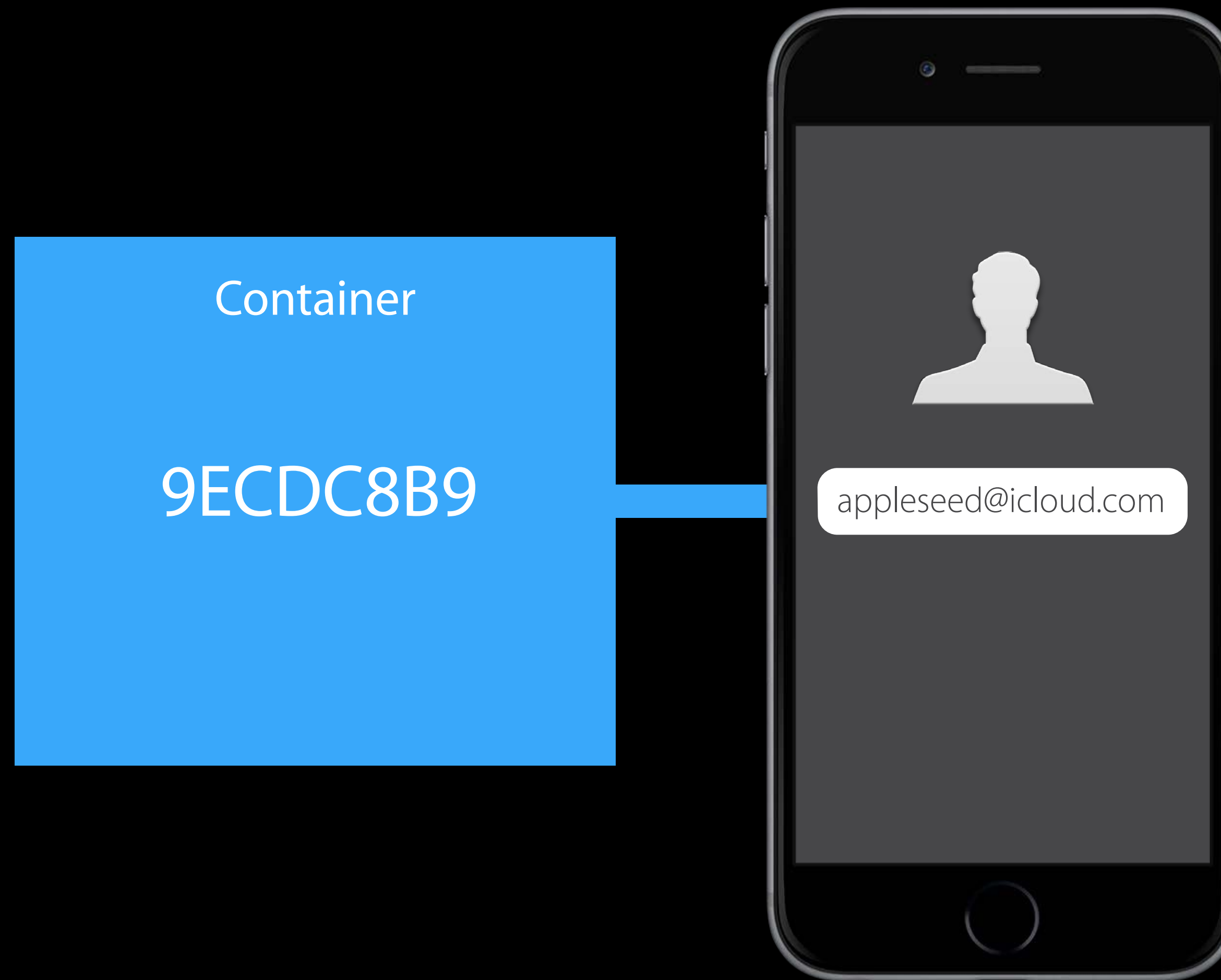
# Authentication

With iOS and OS X



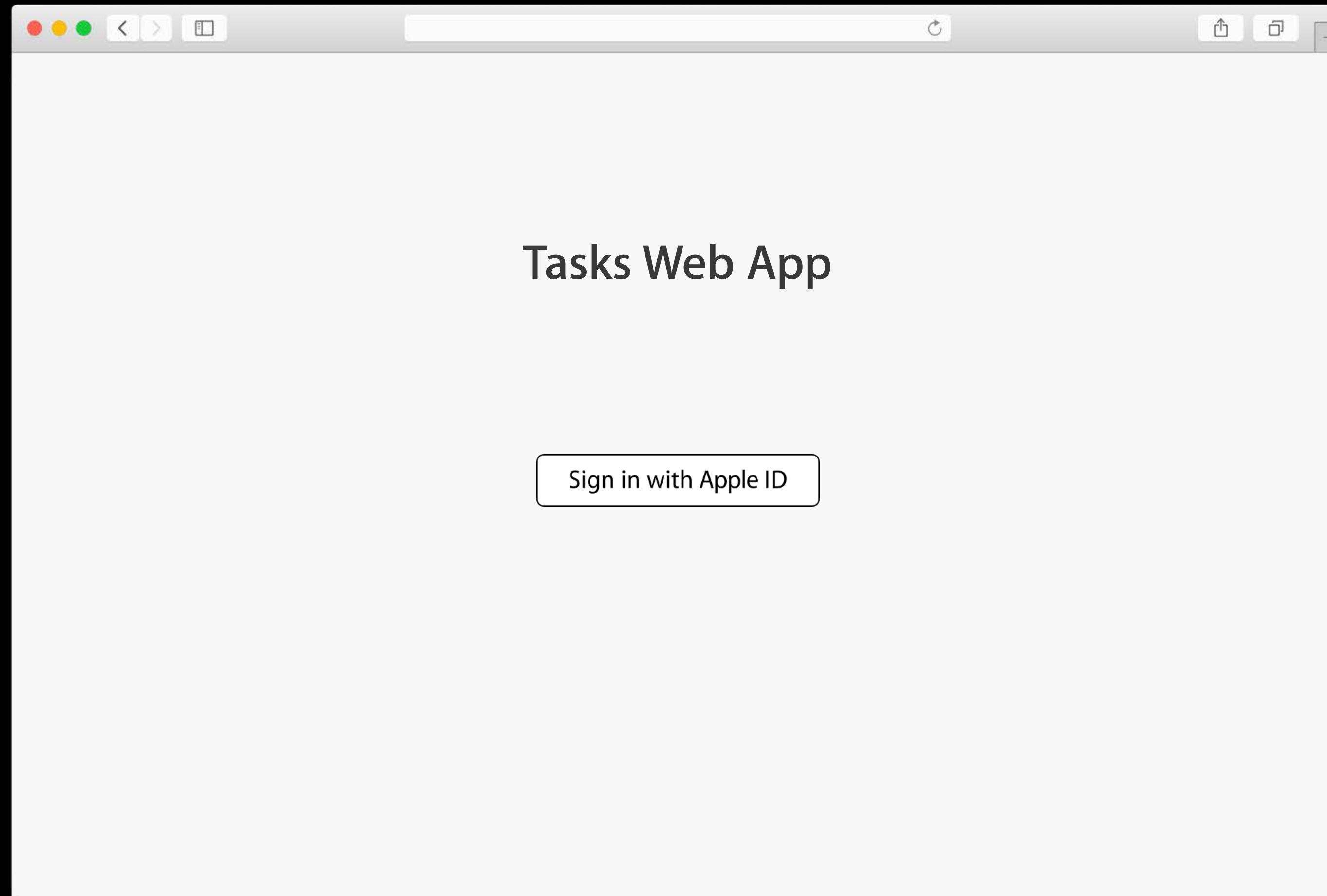
# Authentication

With iOS and OS X



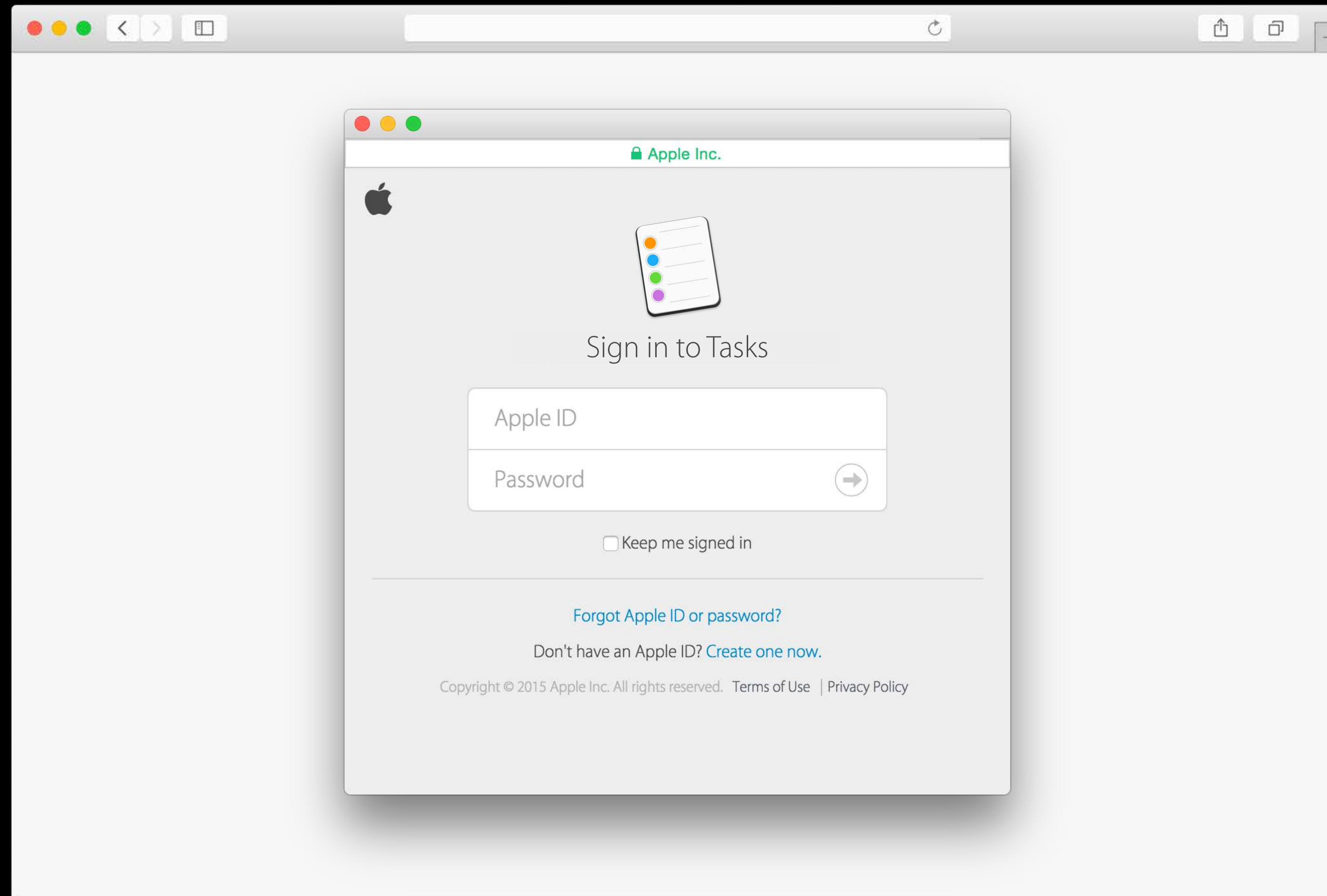
# Authentication

## With CloudKit web services



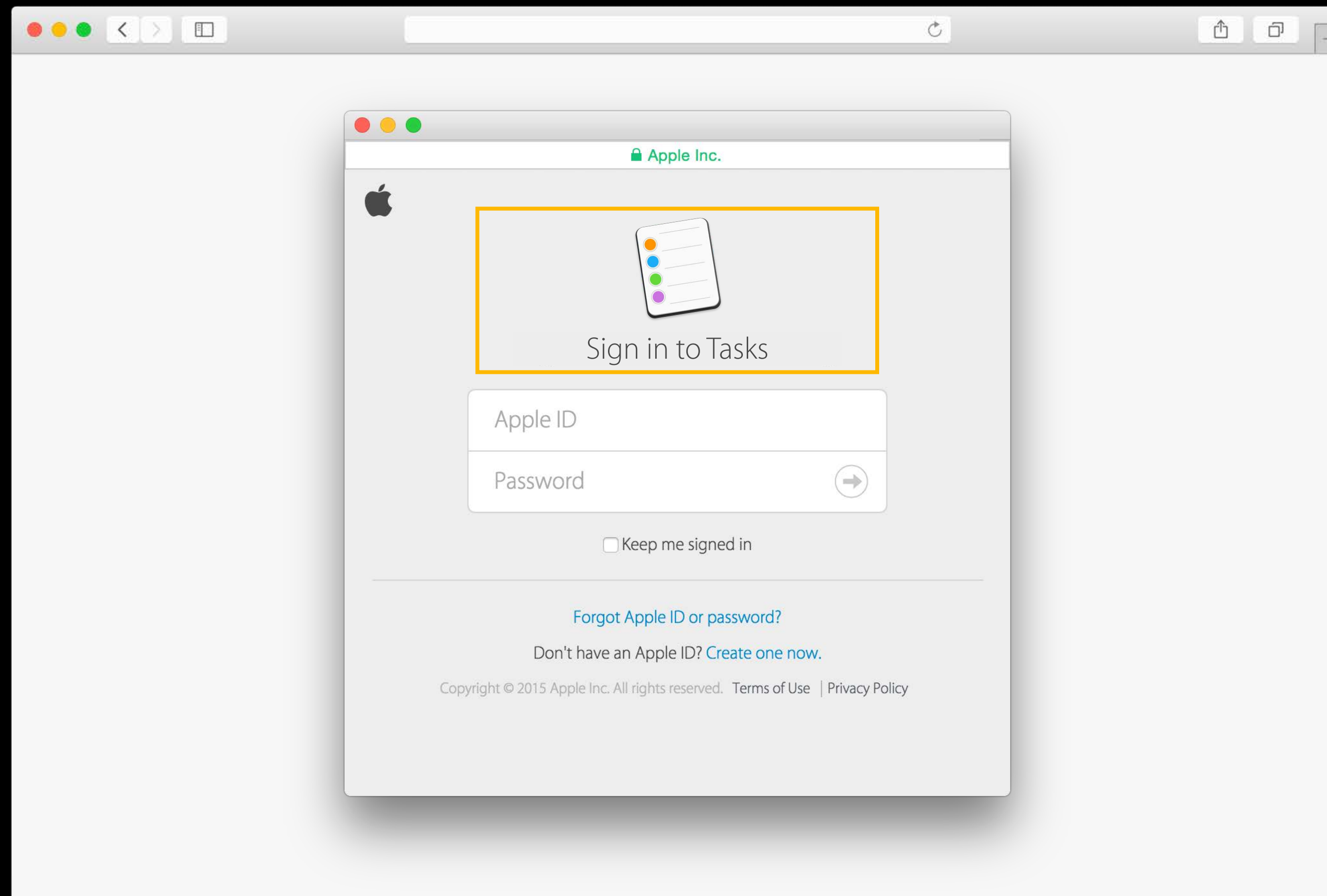
# Authentication

## With CloudKit web services



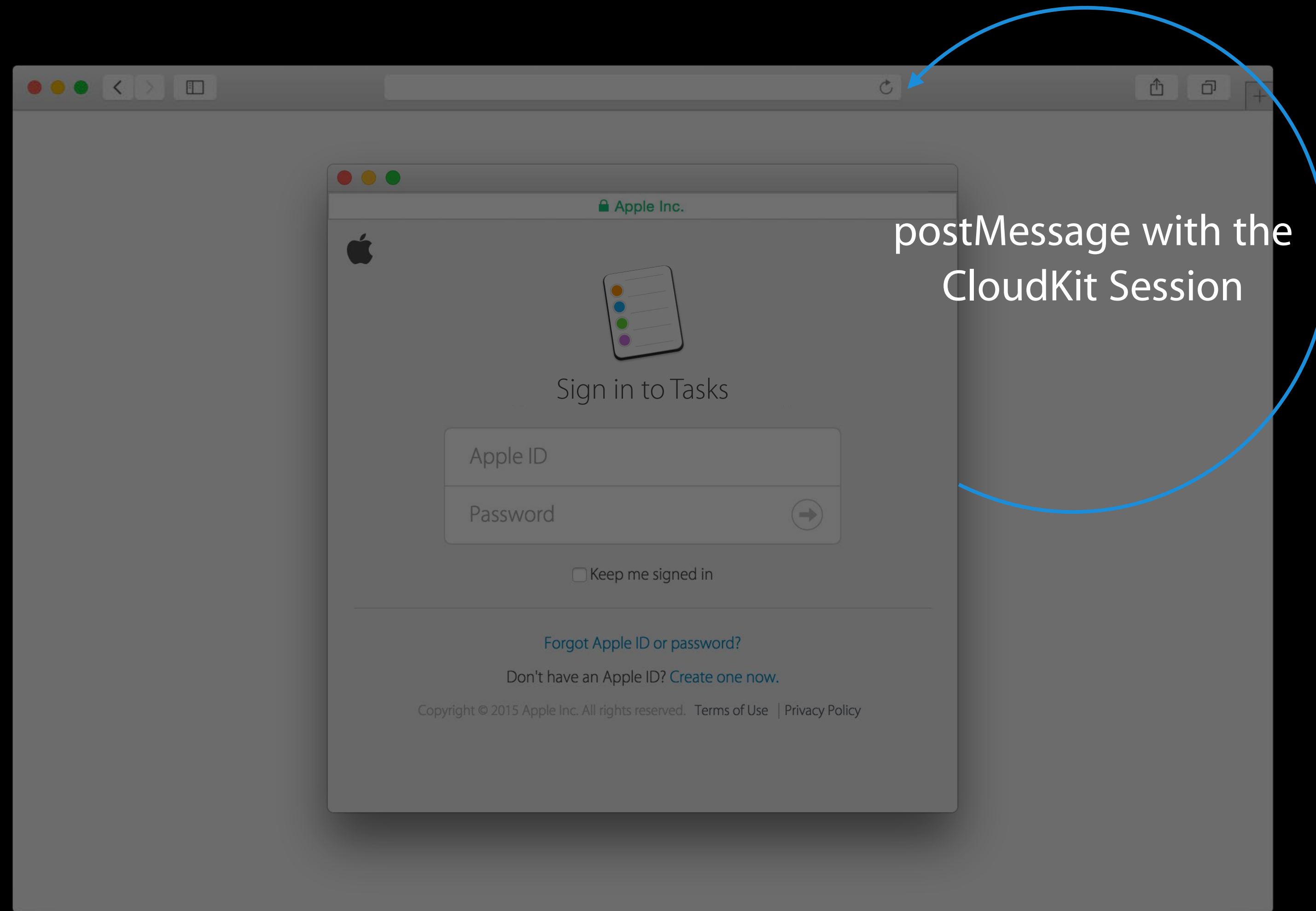
# Authentication

## With CloudKit web services



# Authentication

## With CloudKit web services



# Authentication

Sign-in button and checking authentication state



# Authentication

Sign-in button and checking authentication state

```
<div id="apple-sign-in-button"></div>
```

# Authentication

Sign-in button and checking authentication state

```
<div id="apple-sign-in-button"></div>
```

```
var myContainer = CloudKit.getDefaultContainer();

myContainer.setUpAuth().then(function(userInfo) {
    if(userInfo) {
        // The user is already authenticated
        // userInfo.userRecordName is the stable user identifier
    }
});
```

# Authentication

Handling user sign-in and sign-out

# Authentication

Handling user sign-in and sign-out

```
myContainer.whenUserSignsIn().then(function(userInfo) {  
    // The user just signed in  
});
```

# Authentication

Handling user sign-in and sign-out

```
myContainer.whenUserSignsIn().then(function(userInfo) {  
    // The user just signed in  
});
```

```
myContainer.whenUserSignsOut().then(function() {  
    // The user just signed out  
});
```

# Authentication

Persisting the auth token

# Authentication

Persisting the auth token

```
CloudKit.configure({  
  containers: [{  
    ...  
    auth: {  
      persist: true  
    }  
  }]  
});
```

# Authentication

## Persisting the auth token

```
CloudKit.configure({  
  containers: ...,  
  services: {  
    authTokenStore: {  
      putToken: function(containerIdentifier, token) { ... },  
      getToken: function(containerIdentifier) { ... }  
    }  
  }  
});
```



# Record Operations

# Records

The structure of a CloudKit record

# Records

The structure of a CloudKit record

```
{
  recordName: 'task-1',
  recordType: 'Tasks',
  created: { ... },
  modified: { ... },
  recordChangeTag: '234ljknwer',

  fields: {
    taskName: {
      type: 'STRING',
      value: 'Buy milk'
    }
  }
}
```

# Records

Record field values

# Records

Record field values

CloudKit Framework

CloudKit JS

---

# Records

Record field values

CloudKit Framework	CloudKit JS
NSString	JavaScript String

# Records

Record field values

CloudKit Framework	CloudKit JS
NSString	JavaScript String
NSNumber	JavaScript Number

# Records

## Record field values

CloudKit Framework	CloudKit JS
NSString	JavaScript String
NSNumber	JavaScript Number
NSData	Base64 encoded binary



# Records

## Record field values

CloudKit Framework	CloudKit JS
NSString	JavaScript String
NSNumber	JavaScript Number
NSData	Base64 encoded binary
NSDate	JavaScript Number (UNIX time, ms)

# Records

## Record field values

CloudKit Framework	CloudKit JS
NSString	JavaScript String
NSNumber	JavaScript Number
NSData	Base64 encoded binary
NSDate	JavaScript Number (UNIX time, ms)
CLLocation	Location object

# Records

## Record field values

CloudKit Framework	CloudKit JS
NSString	JavaScript String
NSNumber	JavaScript Number
NSData	Base64 encoded binary
NSDate	JavaScript Number (UNIX time, ms)
CLLocation	Location object
CKReference	Reference object

# Records

## Record field values

CloudKit Framework	CloudKit JS
NSString	JavaScript String
NSNumber	JavaScript Number
NSData	Base64 encoded binary
NSDate	JavaScript Number (UNIX time, ms)
CLLocation	Location object
CKReference	Reference object
CKAsset	Asset object

# Records

## Creating a record

```
var record = {
  recordType: 'Tasks',
  fields: {
    taskName: {
      value: 'Buy milk'
    }
  }
};

var privateDB = myContainer.privateCloudDatabase;

privateDB.saveRecord(record).then(function(response) {
  var record = response.records[0];
});
```

# Records

## Creating a record

```
var record = {  
  recordType: 'Tasks',  
  fields: {  
    taskName: {  
      value: 'Buy milk'  
    }  
  }  
};
```

```
var privateDB = myContainer.privateCloudDatabase;
```

```
privateDB.saveRecord(record).then(function(response) {  
  var record = response.records[0];  
});
```

# Records

## Creating a record

```
var record = {  
  recordType: 'Tasks',  
  fields: {  
    taskName: {  
      value: 'Buy milk'  
    }  
  }  
};
```

```
var privateDB = myContainer.privateCloudDatabase;  
  
privateDB.saveRecord(record).then(function(response) {  
  var record = response.records[0];  
});
```

Queries



# Fetching Records with a Query

# Fetching Records with a Query

```
var query = { recordType: 'Tasks' };  
  
privateDB.performQuery(query).then(function(response) {  
    console.log(response.records);  
});
```

# Fetching Records with a Query

```
var query = { recordType: 'Tasks' };  
  
privateDB.performQuery(query).then(function(response) {  
    console.log(response.records);  
});
```

```
privateDB.performQuery(query, {  
    desiredKeys: ['taskName'],  
    resultsLimit: 10,  
    zoneID: 'allTasks'  
}).then(function(response) {  
    console.log(response.records);  
});
```

# Fetching Records with a Query

Using a filter

# Fetching Records with a Query

Using a filter

```
var query = {  
  recordType: 'Tasks',  
  filterBy: [{  
    fieldName: 'priority',  
    comparator: 'EQUALS',  
    fieldValue: { value: 1 }  
  }]  
};
```

```
privateDB.performQuery(query).then(function(response) {  
  console.log('Tasks with priority 1', response.records);  
});
```

# Fetching Records with a Query

Using a filter

```
var query = {  
  recordType: 'Tasks',  
  filterBy: [{  
    fieldName: 'priority',  
    comparator: 'EQUALS',  
    fieldValue: { value: 1 }  
  }]  
};
```

```
privateDB.performQuery(query).then(function(response) {  
  console.log('Tasks with priority 1', response.records);  
});
```

# Fetching Records with a Query

Continuing queries

# Fetching Records with a Query

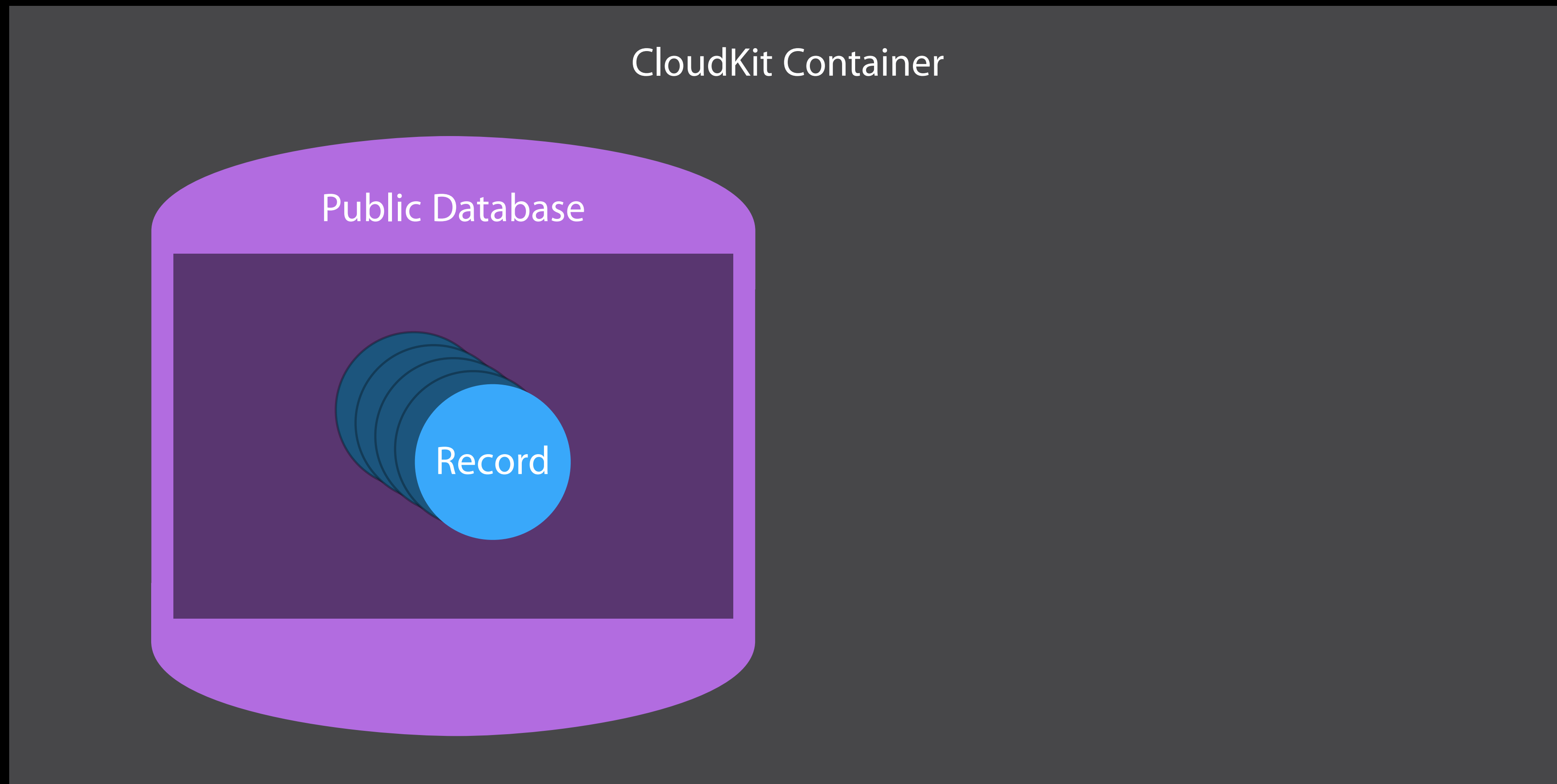
## Continuing queries

```
privateDB.performQuery(query, { resultsLimit: 10 })
  .then(function(response) {
    console.log('First 10 records:', response.records);
    if(response.moreComing) {
      return privateDB.performQuery(response);
    }
  })
  .then(function(response) {
    if(response) {
      console.log('Next 10 records:', response.records);
    }
  });
```

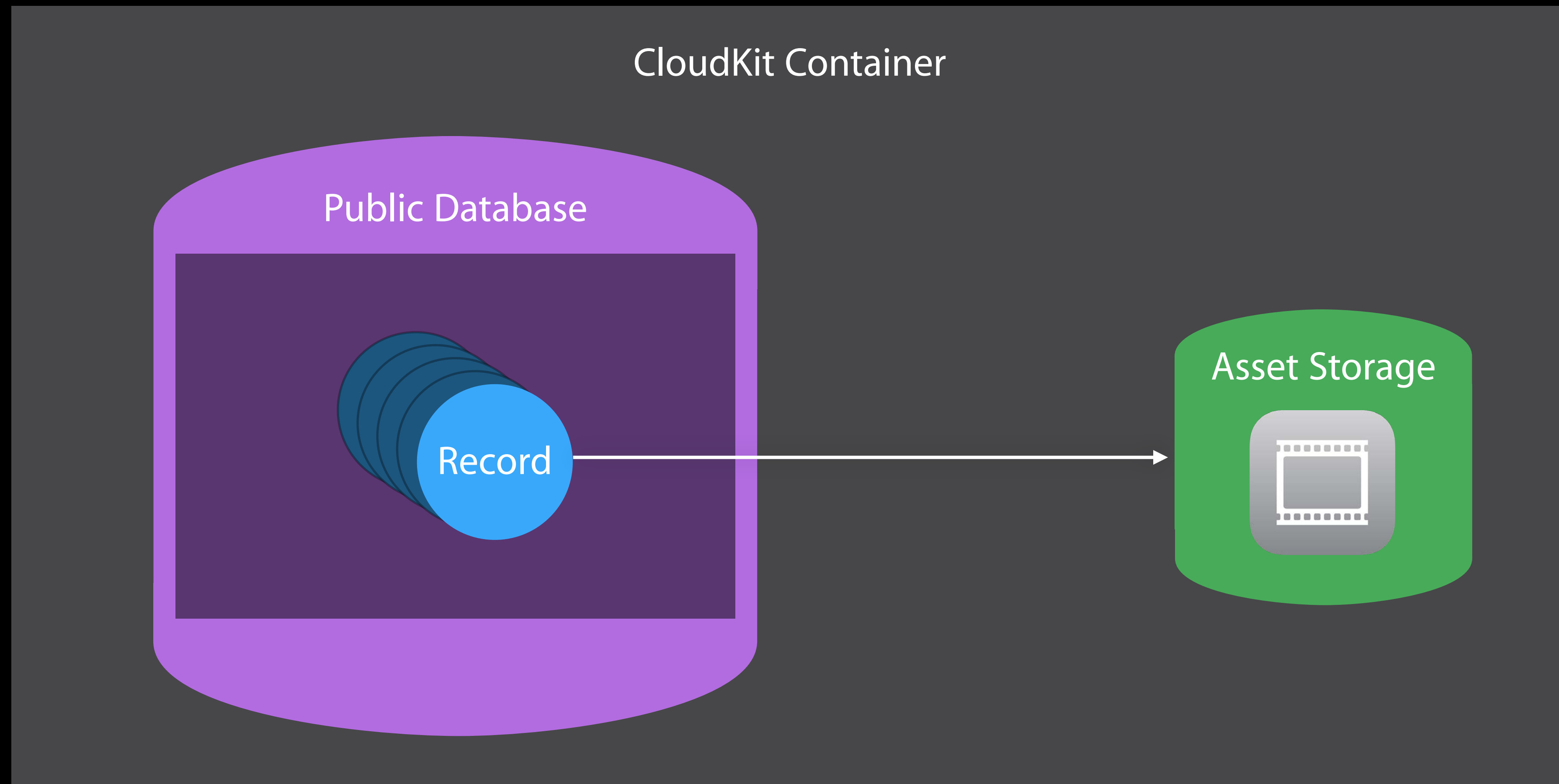


Assets

# Assets



# Assets



# Assets

Uploading an asset

# Assets

## Uploading an asset

```
<input type="file" onchange="handleFileSelect">
<script>
  function handleFileSelect(evt) {
    var myImage = evt.target.files[0];
    database.saveRecord({
      recordType: 'Users',
      fields: {
        profilePhoto: { value: myImage }
      }
    }).then(function(response) {
      console.log(response.records[0]);
    });
  }
</script>
```

# Assets

## Uploading an asset

```
<input type="file" onchange="handleFileSelect">
```

```
<script>
```

```
    function handleFileSelect(evt) {
```

```
        var myImage = evt.target.files[0];
```

```
        database.saveRecord({
```

```
            recordType: 'Users',
```

```
            fields: {
```

```
                profilePhoto: { value: myImage }
```

```
            }
```

```
        }).then(function(response) {
```

```
            console.log(response.records[0]);
```

```
        });
```

```
    }
```

```
</script>
```

# Assets

Downloading an asset

# Assets

## Downloading an asset

```
database.fetchRecord('user1').then(function(response) {  
    console.log(response.records[0]);  
});
```

```
/*  
    fields: {  
        profilePhoto: {  
            value: {  
                downloadUrl: 'https://...'  
            }  
        }  
    }  
*/  
});
```



# Assets

## Downloading an asset

```
database.fetchRecord('user1').then(function(response) {  
    console.log(response.records[0]);
```

```
    /*
```

```
        fields: {  
            profilePhoto: {  
                value: {  
                    downloadUrl: 'https://...'  
                }  
            }  
        }  
    }  
}
```

```
    */
```

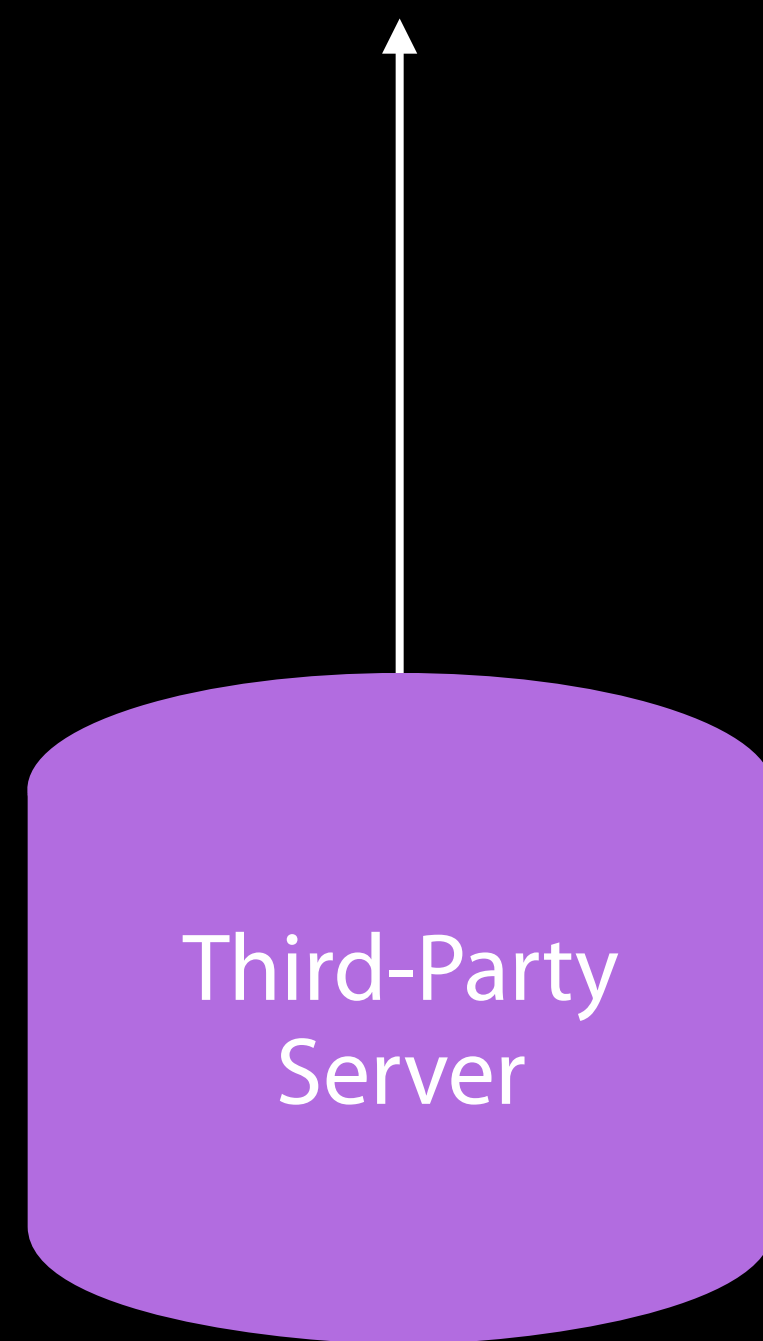
```
});
```

# Subscriptions and Push Notifications



Apple Push Notification Service







Apple Push Notification Service





Apple Push Notification Service



CloudKit Subscriptions

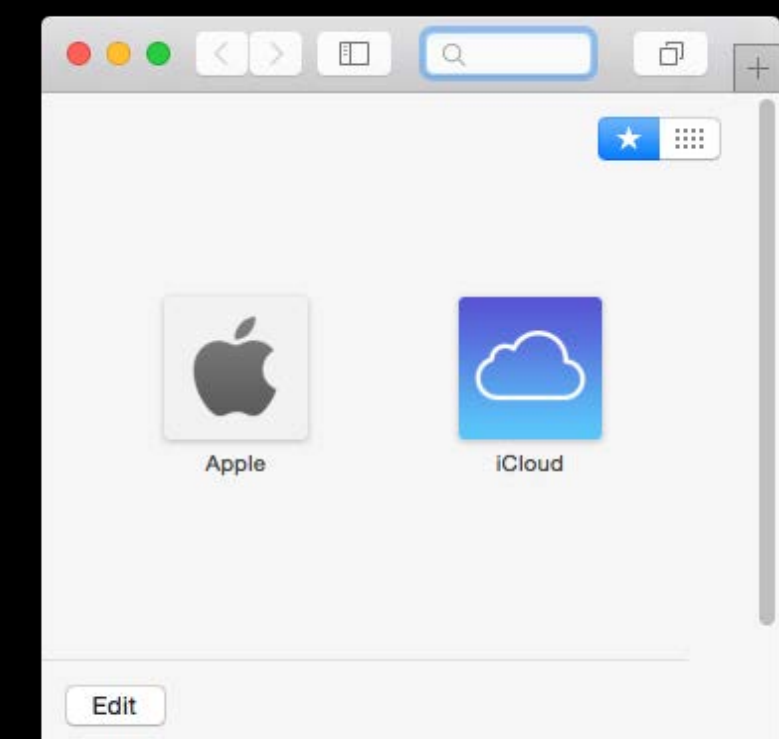




Apple Push Notification Service



CloudKit Subscriptions



# Subscriptions

Subscribing to changes in a zone



# Subscriptions

Subscribing to changes in a zone

```
var zoneSubscription = {  
  subscriptionType: 'zone',  
  subscriptionID: 'changedTasks',  
  zoneID: 'allTasks'  
};
```

```
database.saveSubscription(zoneSubscription);
```

# Subscriptions

Subscribing to changes in a zone

```
var zoneSubscription = {  
  subscriptionType: 'zone',  
  subscriptionID: 'changedTasks',  
  zoneID: 'allTasks'  
};
```

```
database.saveSubscription(zoneSubscription);
```

# Subscriptions

Subscribing to changes based on a query

```
var querySubscription = {  
  subscriptionType: 'query',  
  subscriptionID: 'changedTasks',  
  zoneID: 'allTasks',  
  firesOn: ['create', 'update', 'delete'],  
}
```

# Subscriptions

Subscribing to changes based on a query

```
var querySubscription = {  
  subscriptionType: 'query',  
  subscriptionID: 'changedTasks',  
  zoneID: 'allTasks',  
  firesOn: ['create', 'update', 'delete'],  
  query: {  
    recordType: 'Tasks',  
    filterBy: [{  
      fieldName: 'priority',  
      comparator: 'EQUALS',  
      fieldValue: { value: 1 }  
    }]  
  }  
};
```

# Push Notifications

Get notified when subscriptions are firing

# Push Notifications

Get notified when subscriptions are firing

```
var myContainer = CloudKit.getDefaultContainer();  
myContainer.registerForNotifications();
```

# Push Notifications

Get notified when subscriptions are firing

```
var myContainer = CloudKit.getDefaultContainer();
```

```
myContainer.registerForNotifications();
```

```
myContainer.addNotificationListener(function(notification) {  
    console.log(notification);  
});
```

# Best Practices with CloudKit JS



# Best Practices with CloudKit JS

Dynamically link to the CDN-hosted version

# Best Practices with CloudKit JS

Dynamically link to the CDN-hosted version

Consider loading CloudKit JS asynchronously on your page

# Best Practices with CloudKit JS

Dynamically link to the CDN-hosted version

Consider loading CloudKit JS asynchronously on your page

Handle request throttling responses

# Summary

# Summary

Full-parity interface

# Summary

Full-parity interface

Enabled on a per-container basis

# Summary

Full-parity interface

Enabled on a per-container basis

CloudKit handles authentication, metadata storage, and asset upload/download

# Summary

Full-parity interface

Enabled on a per-container basis

CloudKit handles authentication, metadata storage, and asset upload/download

You handle hosting your static assets and code



# Summary

Full-parity interface

Enabled on a per-container basis

CloudKit handles authentication, metadata storage, and asset upload/download

You handle hosting your static assets and code

Use CloudKit JS

# More Information

Documentation and Videos

CloudKit Documentation

[developer.apple.com/cloudkit](https://developer.apple.com/cloudkit)

Technical Support

Apple Developer Forums

[developer.apple.com/forums](https://developer.apple.com/forums)

Developer Technical Support

[developer.apple.com/support/technical](https://developer.apple.com/support/technical)

General Inquiries

[cloudkit@apple.com](mailto:cloudkit@apple.com)

# Related Sessions

What's New in CloudKit	Mission	Tuesday 3:30PM
CloudKit Tips and Tricks	Pacific Heights	Thursday 4:30PM

# Related Labs

CloudKit Lab	Frameworks Lab A	Thursday 9:00AM
CloudKit Lab	Frameworks Lab D	Friday 9:00AM

