



# GEFORCE NOW SDK 2.1

## INTEGRATOR PRIMER | FEBRUARY 2024

# INTRODUCING THE GEFORCE NOW SDK

THE PATH TO CLOUD GAMING

The GeForce NOW SDK is a set of **APIs, components, and samples** that enable developers to seamlessly add the power of **cloud gaming** to their Windows and Linux ecosystems.

# INTRODUCING THE GEFORCE NOW SDK

## LINUX OPERATING SYSTEM SUPPORT

With this release, the GeForce NOW SDK now supports integration with native Linux applications and games as well as continued Windows support.

Certain APIs are not yet supported on Linux because of the design of the GeForce NOW ecosystem. Support for these API will be added in future SDK releases.

Per-API OS limitations are defined in this document as well as the API documentation.

# THE POWER OF GEFORCE NOW

## APIS AND BENEFITS

### APPLICATION

LAUNCHER / STORE

GAME

ANTI-CHEAT

### SDK APIs

STREAM LAUNCH

ACCOUNT  
LINKING & SINGLE  
SIGN-ON

ENVIRONMENT  
CHECK

DEEP LINKING

CLIENT & SESSION  
INFO

### GEFORCE NOW



Single  
Sign-On



Play Without  
Installing



Launch In  
Seconds



Premium GFX  
Anywhere



Always  
Up-To-Date



Robust  
Gaming

# TOPICS COVERED

## BY TARGET AUDIENCE

This document provides a high-level overview of user and developer **benefits** along with integration **flows**, **interfaces** and **processes** required to stream.

### LAUNCHER INTEGRATION

This topic covers the details needed by User Experience Designers and Developers alike to integrate the GeForce NOW SDK into an existing launcher application, as well as obtain information about client systems and streaming settings while running inside GeForce NOW.

An example launcher application is shown along with source provided in the SDK.

### GAME INTEGRATION

This topic covers the details on APIs available to games running in GeForce NOW to obtain various client system information and streaming session state.

This topic is for Game Developers and Product Managers that want their games to run seamlessly in GeForce NOW.

### ACCOUNT LINKING

This topic covers linking third-party accounts with a user's GeForce NOW account to implement Single Sign-On to provide a seamless integration experience.

Understanding Identity Management is key for Developers.

### GFN DEEP LINKING

This topic covers linking into the GeForce NOW web browser client via URL deep links.

This topic is for Game Publishers that wish to publish links to their games in GFN as well as Gaming Platforms that wish to link users to playing games in GFN.

Example URLs are shown along with documentation provided in the SDK.



# LAUNCHER INTEGRATION

ADDING CLOUD GAMING - WINDOWS PLATFORM ONLY



# STREAM GAMES DIRECTLY FROM YOUR WINDOWS LAUNCHER

## EXAMPLE EXPERIENCE

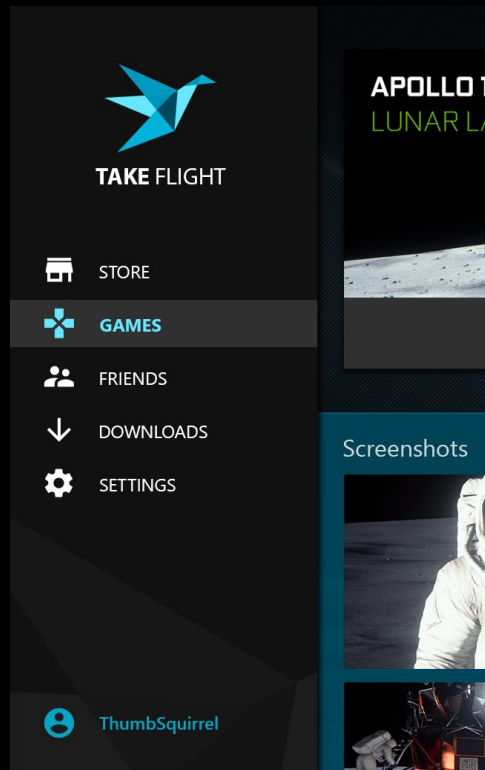
Log In

ThumbSquirrel

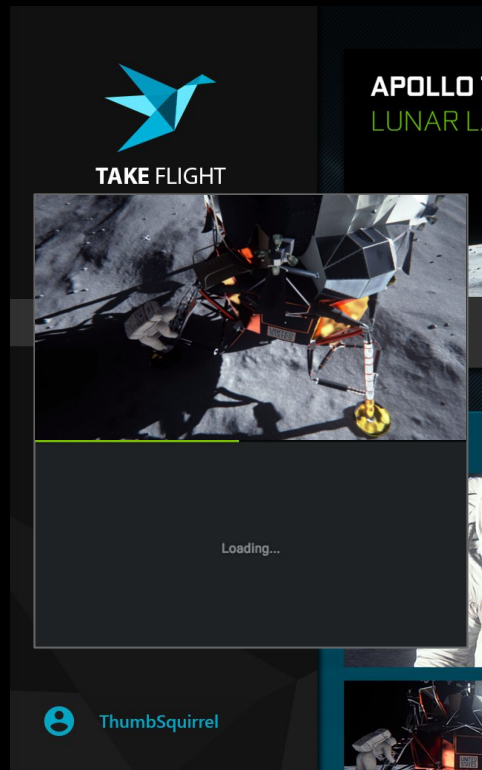
\*\*\*\*\*

SIGN IN

Single Sign-On  
w/ Linked Accounts



Seamless  
Integration



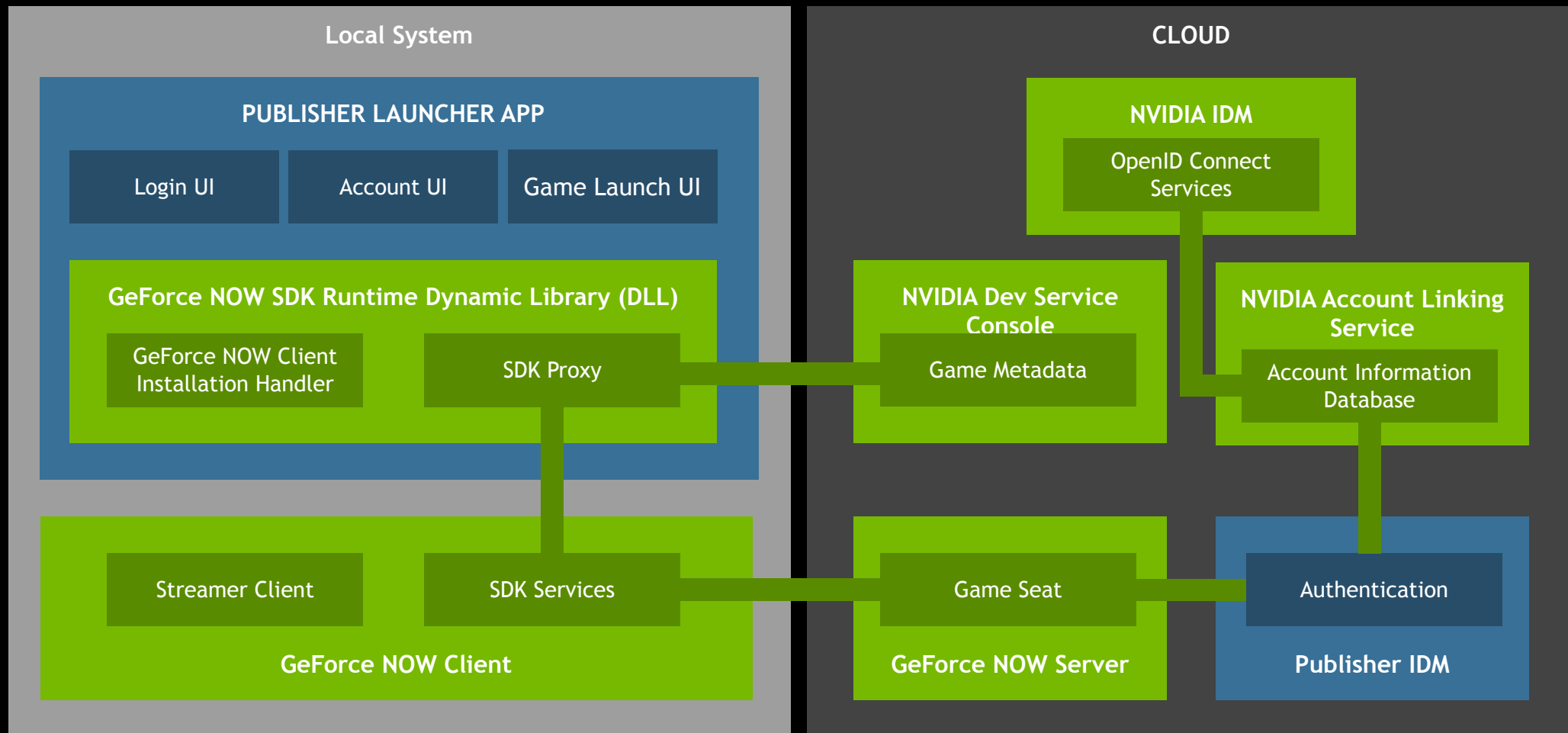
Launch  
In Seconds



Premium Experience  
Everywhere

# ECOSYSTEM

## COMPONENTS AND CONNECTIONS





# EXAMPLE LAUNCHER

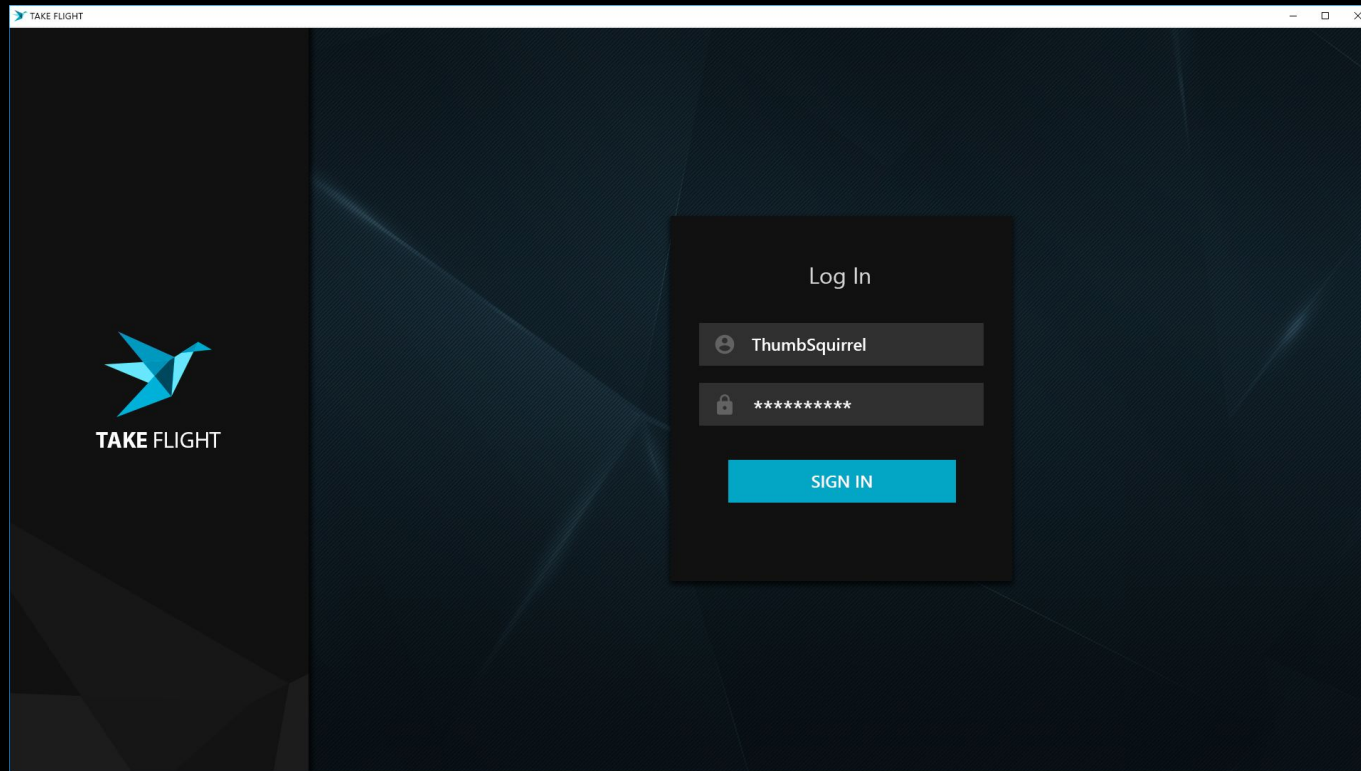
## USER AUTHENTICATION

### LOGIN DIALOG

Sign on to the system is handled by the launcher application itself which gives the gamer access to the ecosystem provided by the publisher.

Signing on to the system automatically restores connections to any third-party linked accounts, including the gamer's NVIDIA GeForce NOW account.

Gamers link third-party accounts by opening the Settings page.



# EXAMPLE LAUNCHER

## STREAMING WITH NVIDIA GEFORCE NOW

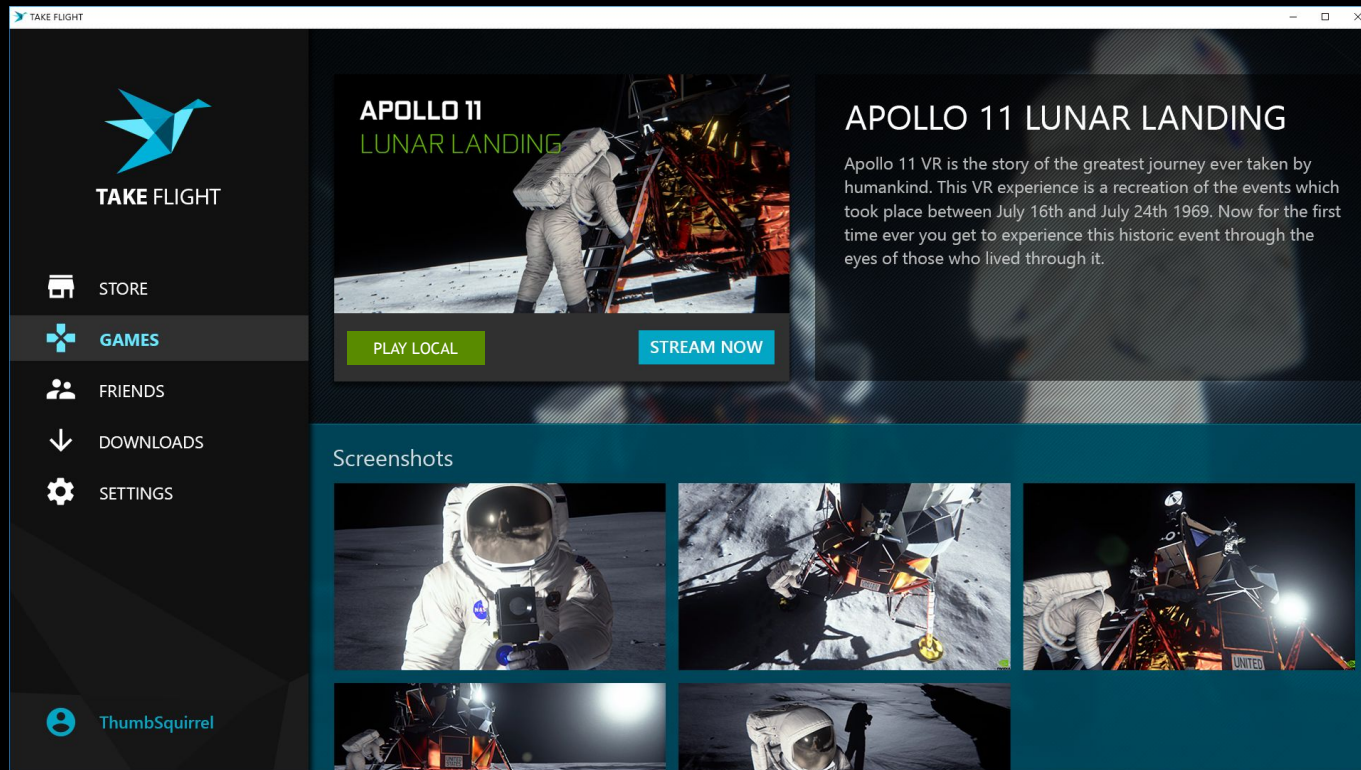
### GAMES PAGE

Gamers are shown an element in the application which opens the streamer window for those games which are supported by NVIDIA GeForce NOW.

The application uses SDK API calls to verify the application is not running inside GFN and to verify a specific game is able to be streamed from GFN.

The application shows the **STREAM NOW** button, or doesn't, based on whether or not the specific game can be streamed.

\*Streaming a game in Launcher via SDK integration is limited to Windows.



# REQUIRED INTEGRATION

## ADDING A CLOUD STREAMING OPTION (WINDOWS PLATFORM ONLY)

### LOGIC FLOWS

- Check with SDK to confirm not in GFN to know when to allow streaming
- Use SDK to start streaming when user activates UI trigger, which first installs the GeForce NOW client as needed
- Use SDK session callback to know the status of the stream

### RELEVANT SDK METHODS

```
GfnRuntimeError gfnInitializeRuntimeSDK(GfnDisplayLanguage) ;  
bool gfnIsRunningInCloud(void) ;  
void gfnStartStreamAsync(StartStreamInput * input, StartStreamCallbackSig  
    cb, void * context, unsigned int timeoutMs) ;  
void gfnShutdownRuntimeSDK(void) ;
```

\* The gfnStartStream and gfnStartStreamAsync APIs are supported on Windows only.

# OPTIONAL INTEGRATION

## USE OF API WRAPPER FUNCTIONS

Use the C-based API wrapper functions to avoid needing to managing the lifecycle of the SDK library as well as calling export functions.

Use the C-based `GfnSdk_SecureLoadLibrary` API included with the SDK to automatically validate the SDK libraries to avoid tampering and spoofing.

### LOGIC FLOWS

- Load GFN SDK library and initialize the SDK
- Start streaming when user activates UI trigger
- Use SDK callbacks to know the status of the stream
- Release SDK and unload GFN SDK library

### RELEVANT SDK METHODS

```
GfnRuntimeError GfnInitializeSDK(gfnDisplayLanguage) ;  
bool GfnIsRunningInCloud() ;  
GfnRuntimeError GfnStartStreamAsync(StartStreamInput * input,  
StartStreamCallbackSig cb, void * context, unsigned int timeoutMs) ;  
void gfnShutdownRuntimeSDK() ;
```

\* The `GfnStartStreamAsync` and `StartStreamCallback` APIs are supported on Windows only.

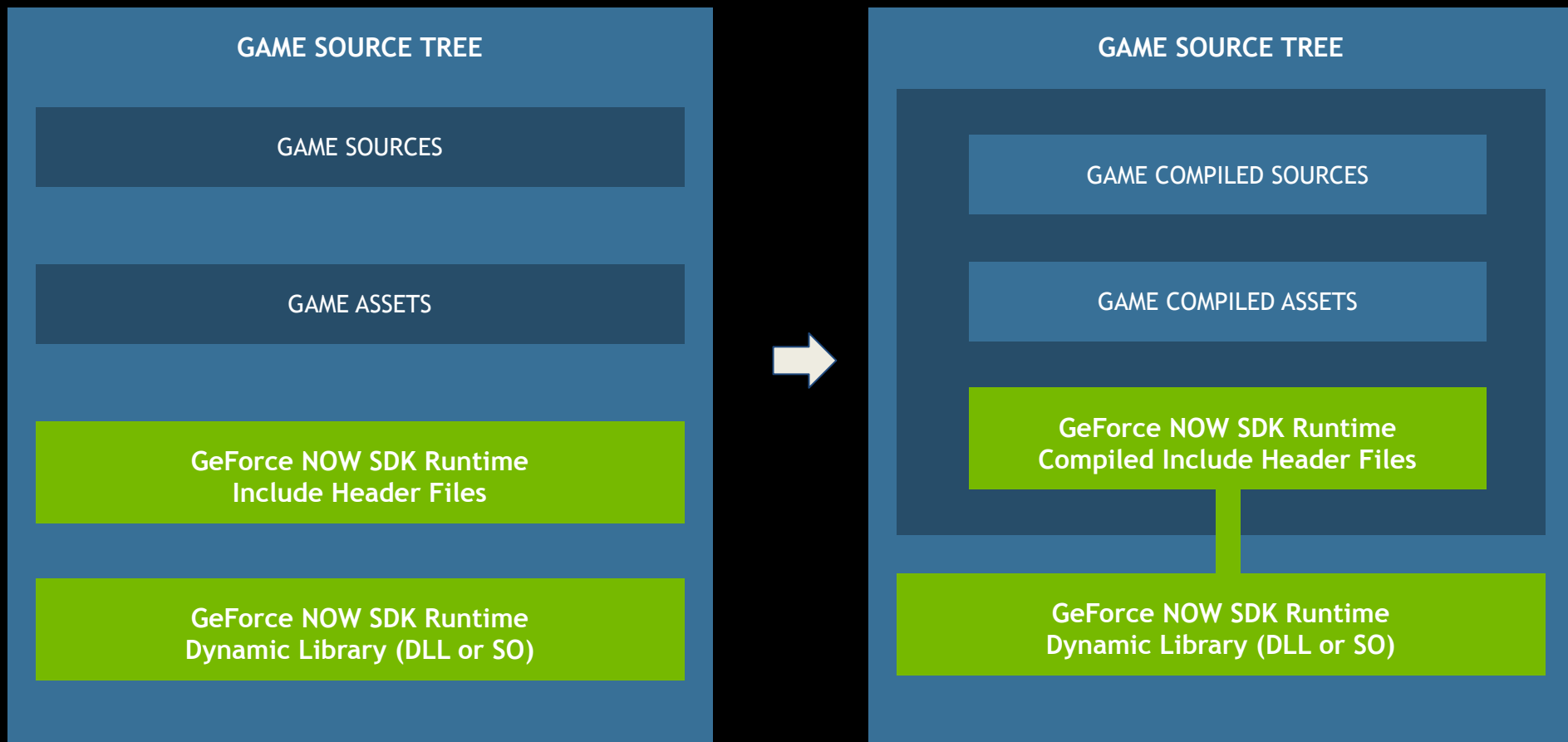


# GAME INTEGRATION

## APIS FOR SEAMLESS INTEGRATION

# INTEGRATION

## SDK COMPONENT INCLUSION





# CLOUD APIS

## OBTAINING INFORMATION ABOUT CLIENTS OR GFN SESSION STATE

### LOGIC FLOWS

- Check with SDK to know when running in GeForce NOW environment
- Use SDK to obtain various information about the connected client, network performance and streaming session
- Use SDK callbacks to know when GeForce NOW session status or client system changes to update game behavior

### RELEVANT SDK METHODS

```
GfnRuntimeError gfnInitializeRuntimeSDK(GfnDisplayLanguage);  
bool gfnIsRunningInCloud(void) or  
GfnRuntimeError GfnIsRunningInCloudSecure(GfnIsRunningInCloudAssurance* assurance);  
GfnRuntimeError gfnGetClientInfo(GfnClientInfo* clientInfo);  
GfnRuntimeError gfnGetSessionInfo(GfnSessionInfo* sessionInfo);  
GfnRuntimeError gfnRegisterClientInfoCallback(ClientInfoCallback clientInfoCallback, void* pUserContext);  
GfnRuntimeError gfnRegisterPauseCallback(PauseCallbackSig pauseCallback, void* pUserContext);  
GfnRuntimeError gfnRegisterSaveCallback(SaveCallbackSig saveCallback, void* pUserContext);  
GfnRuntimeError gfnRegisterExitCallback(ExitCallbackSig exitCallback, void* pUserContext);  
void gfnShutdownRuntimeSDK(void);
```



# CLOUD APIS

## COMMUNICATING BETWEEN CLIENT AND APPLICATIONS ON THE GAME SEAT

The `gfnSendMessage` API and corresponding callback allows platforms launching GFN sessions to have bi-directional communication with applications running on the game seat at any time during the session.

For more information, refer to the documentation and working example code in the sample applications.

### LOGIC FLOWS

- Send messages from the platform's launcher to an application running on the GFN Game Seat
- Receive messages within the platform's launcher which were initiated from the application running on the GFN Game Seat

### RELEVANT SDK METHODS

```
GfnRuntimeError gfnSendMessage(const char* pchMessage, unsigned int length);
```

```
GfnRuntimeError gfnRegisterMessageCallback(MessageCallbackSig messageCallback,  
void* pUserContext);
```

# CLOUD APIS

## SUPPORT PRE-WARM GAME SESSIONS

With Prewarm game session support, GFN allows users to get into their games faster and skip game loading states and creates a better user experience.

Developers can work with their GFN representatives to integrate this feature into their games.

### GAME ONBOARDING FOR PREWARM SESSIONS

- Game integrates GFN SDK to make use of pre-warm APIs
- Game developer works with NVIDIA to onboard prewarm-supported game builds

### LOGIC FLOWS

- GFN allocates a game seat for the game launch
- Game loads all generic data (no user specific data as no user is connected)
- Game registers for session initialization callback with SDK and goes idle waiting for callback
- GFN SDK triggers session init callback once a user session is initialized
- Game loads user data and calls gfnAppReady API to notify GFN that it is ready to be displayed
- GFN begins streaming game to user

### RELEVANT SDK METHODS

```
GfnRuntimeError gfnRegisterSessionInitCallback(SessionInitCallbackSig  
sessionInitCallback, void* pUserContext);  
GfnRuntimeError gfnAppReady(bool success, const char * status);
```



# ACCOUNT LINKING

## FOR SINGLE SIGN-ON

# LINKING ACCOUNTS TO SUPPORT SINGLE SIGN-ON

## INTEGRATION NECESSARY

The best experience for gamers requires a one time operation to link their store account with an NVIDIA account. Once accomplished, the user enjoys a seamless experience of playing games in GFN without needing to enter their publisher credentials during the streaming session.

### ACCOUNT LINKING

- User performs a one time account linking operation from the GFN client.
- Launcher IDM cooperates with NVIDIA Accounting Linking service through OpenID Connect protocols to exchange account information.
- Account information is used to create mapping between two accounts.

### SINGLE SIGN-ON

- User links accounts to share account information.
- User streams a game that uses linked accounts.
- In-stream launcher or game authorizes streaming via account information provided by GFN SDK.

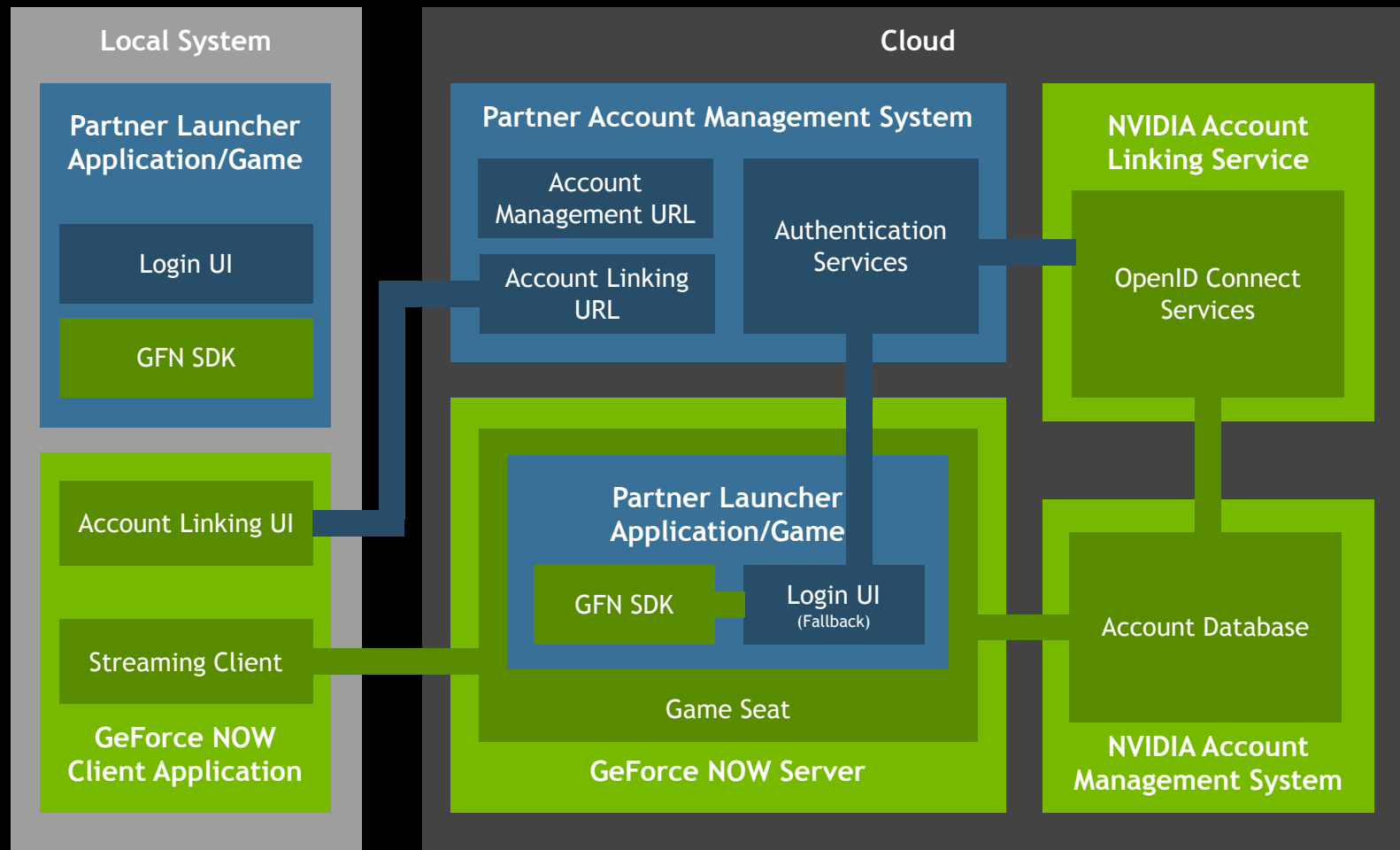
# ACCOUNT LINKING

## COMPONENTS AND CONNECTIONS

### GEFORCE NOW ACCOUNT LINKING ARCHITECTURE

This overview illustrates where each component is installed or hosted, and shows important connections between various components involved in account linking and Single Sign-On.

For more detailed information, see the Account Linking and Single Sign-On guide in ./doc folder.



# EXAMPLE LAUNCHER

## THIRD-PARTY ACCOUNT LINKING

### SETTINGS PAGE

Gamers who create and link an NVIDIA GeForce NOW account are able to stream games to their device without having to manually enter their publisher credentials.

Linking accounts is accomplished using the industry standard OpenID Connect (OIDC) workflow in a web browser window.

Authorization tokens and user account data are then cached by the backend services so login authorization data can be provided to the publisher application running in the GeForce NOW game seat.

### CONNECTIONS

Manage your connected accounts.



#### Ubisoft

Your account is connected to automatically sign in to games owned on Ubisoft Connect.

DISCONNECT



#### Epic Games

Your account is connected to automatically sign in to Fortnite.

DISCONNECT



#### Steam

Connect your account to add supported games from your Steam library.

CONNECT



# DEEP LINKING

## GEFORCE NOW GAME ACCESS VIA URL



# INTRODUCING GEFORCE NOW DEEP LINKS

## THE PATH TO SHARING CLOUD GAMING

A GeForce NOW Deep Link is a specialized URL that directs a user's web browser to the Details page of a specific game in the GFN web browser client.

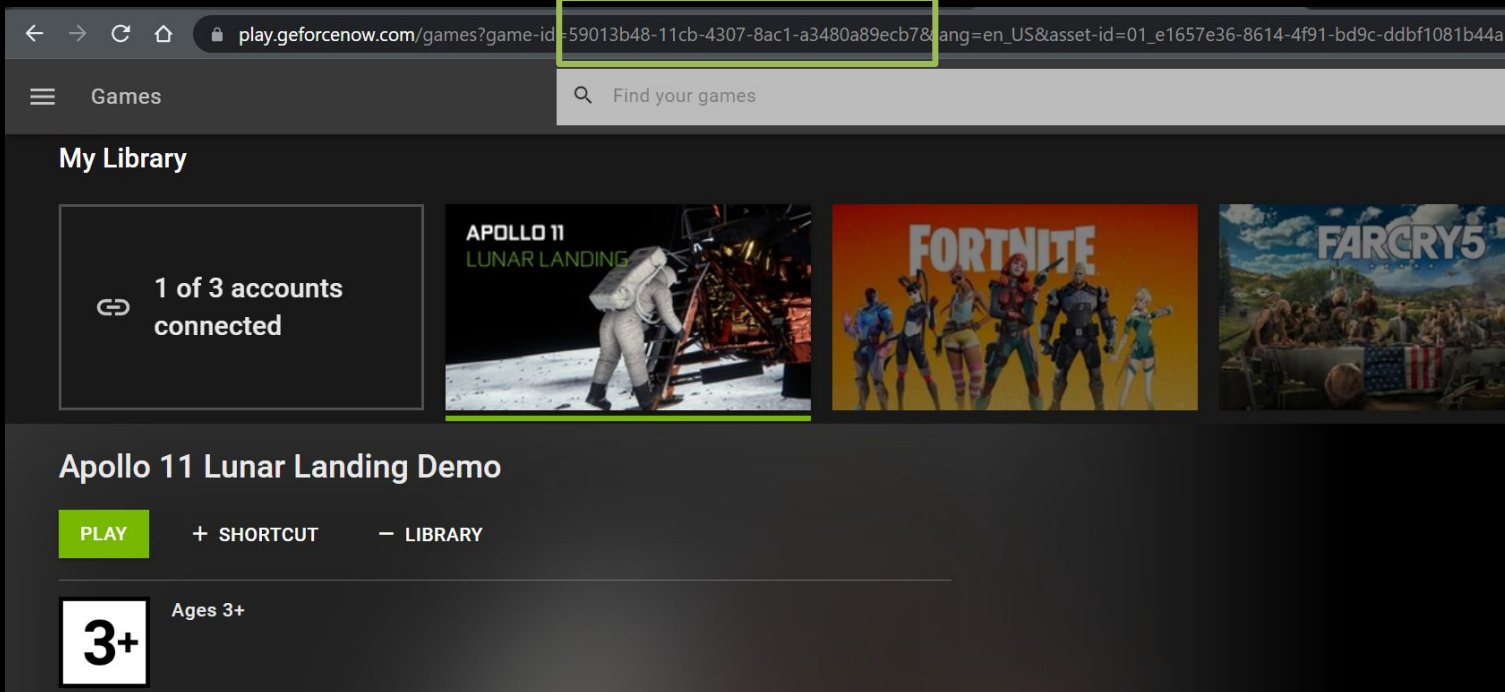
This can be used to invite users via social or web media to discover and play games in GeForce NOW.

The Deep Link requires obtaining the game ID, which can be obtained from the web client's URL in the browser's address bar and a source that is user defined. The source and campaign are user defined.

`https://play.geforcenow.com/games?game-id=<game-id>&utm_source=<utm_source>&utm_campaign=<utm_campaign>`

“GameWebStore”

“NewLaunch”





# NEXT STEPS

## INTEGRATING GEFORCE NOW SDK

# START WITH THE QUICK START GUIDE

## EXAMPLES OF MOST COMMON INTEGRATION USE CASES

### GEFORCE NOW QUICK START GUIDE

This document gives an overview of the GeForce NOW SDK, best practices of SDK usage, and how-to code examples for the most common GeForce NOW use cases.

#### Use cases include

- Check if running in GeForce NOW cloud environment
- Obtaining Client and Session Info
- Supporting Pre-Warm
- Two-way message handling
- Obtaining Partner Data

Refer to SDK-GFN-QUICK-START.pdf in ./doc folder.

### Introduction

NVIDIA GeForce NOW (GFN) allows users to enjoy their library of PC games on many internet-enabled devices via the power of cloud gaming. This includes PCs, Macs, Smart TVs, and mobile devices such as smartphones and tablets.

GFN has the ability to onboard your Windows and Linux games and applications as-is for your users to enjoy. For tighter integration of these applications with GFN, NVIDIA provides the GeForce NOW Runtime Software Development (GFN SDK) to developers so that their applications can perform various tasks; from knowing when the application is running in GFN to getting information about the GFN session or the user's client system.

This document is provided as part of the SDK to aid in getting the developer up to speed with the SDK quickly, as well as providing source code pointers for the most common of integration scenarios.

### Audience

This document is directed towards developers of applications and games that are part of the GFN ecosystem, and wish to provide their users a seamless experience between their application and GFN.

### SDK Overview

The GFN Runtime SDK consists of an ever-growing collection of C-based native APIs that support both Windows and Linux applications. Along with the C source files that define the APIs, the SDK also includes various documentation to aid in development, as well as a collection of samples in source code form that provide example execution of the APIs in various scenarios.

The SDK is designed with ease of use in mind, with API design meant to allow function calls and data consumption to be straightforward. In addition, the SDK is designed to support backwards compatibility as a core requirement, which allows developers to move to new versions of the SDK without the worry of code breakage related to existing API calls due to API definition alterations or deprecation.

# TRY THE INCLUDED SAMPLES

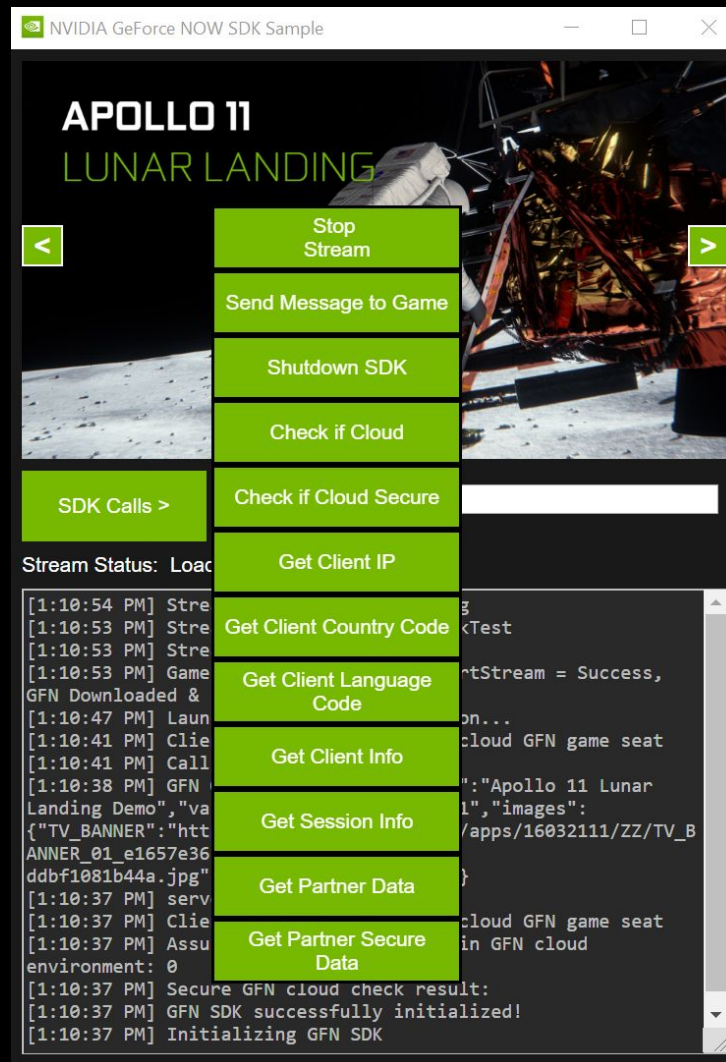
## EXAMPLE CODE FOR ESSENTIAL FLOWS

### GEFORCE NOW SDK SAMPLE LAUNCHER APPLICATION

This sample application contains working code with just enough UI controls showing essential integration points and functionality provided by the GeForce NOW SDK.

The SDK APIs used in this sample provides examples to:

- Check if running in GeForce NOW cloud environment
- Obtain the list of games supported by GeForce NOW
- Start streaming of a game and get stream state status
- Get Client and Session info (when run in Geforce NOW cloud environment)



# TRY THE INCLUDED SAMPLES

## EXAMPLE CODE FOR ESSENTIAL FLOWS

### GEFORCE NOW SDK GAME API SAMPLE

This sample application contains working code for calling game-related APIs available in the SDK when run inside the GFN environment.

The SDK APIs used in this sample provides examples to:

- Check if running in GeForce NOW cloud environment
- Register for Session change callbacks and handle the notifications when there are changes
- Get Client and Session info

```
// Example application main
int _tmain(int argc, _TCHAR* argv[])
{
    ApplicationInitialize();

    // Sample C API call
    bool bIsCloudEnvironment = false;
    GfnIsRunningInCloud(&bIsCloudEnvironment);
    printf("\nApplication executing in GeForce NOW environment: %s\n", (bIsCloudEnvironment == true) ? "true" : "false");

    if (bIsCloudEnvironment) // More sample C API calls.
    {
        GfnError runtimeError = gfnSuccess;

        char* clientIp = NULL;
        runtimeError = GfnGetClientIpV4(&clientIp);
        if (runtimeError == gfnSuccess)
        {
            printf("Retrieved GeForce NOW Client I.P.: %s\n", clientIp);
        }
        else
        {
            printf("Failed to retrieve GeForce NOW Client I.P. GfnError: %d\n", (int) runtimeError);
        }

        char* clientLanguageCode = NULL;
        runtimeError = GfnGetClientLanguageCode(&clientLanguageCode);
        if (runtimeError == gfnSuccess)
        {
            printf("Retrieved GeForce NOW client language code: %s\n", clientLanguageCode);
        }
        else
        {
            printf("Failed to retrieve GeForce NOW client language code. GfnError: %d\n", (int) runtimeError);
        }

        char clientCountryCode[3];
        runtimeError = GfnGetClientCountryCode(clientCountryCode, 3);
        if (runtimeError == gfnSuccess)
        {
            printf("Retrieved GeForce NOW client Country code: %s\n", clientCountryCode);
        }
        else
        {
            printf("Failed to retrieve GeForce NOW client Country code. GfnError: %d\n", (int) runtimeError);
        }
    }
}
```





**THANKS FOR USING**  
**THE GEFORCE NOW SDK**