Online Subsystem

Overview of the various systems related to the online platform.

The **Online Subsystem** and its interfaces provide a common way to access the functionality of online services such as Steam, Xbox Live, Facebook, and so on. When working on a game that ships on multiple platforms, or with support for multiple online services, the Online Subsystem ensures that the only changes developers need to make are configuration adjustments for each supported service.

Design Philosophy

The Online Subsystem is fundamentally designed to handle asynchronous communication with a variety of online services. Since network connection speeds, server delays, and the running times of backend services are all unknown to the local machine, interactions with these systems take an unpredictable amount of time. To deal with this, the Online Subsystem uses Delegates for all remote operations and guarantees that those Delegates will be called whenever a supported, asynchronous feature is used. In addition to providing the ability to respond to requests as they finish, and query in-flight requests. Delegates also provide a single code path to follow, removing the need for developers to write custom code to catch different success or failure conditions.

Modular, service-specific interfaces group supported features together. For example, the Friends Interface handles everything related to friends lists, the Achievements Interface handles listing, checking, and awarding Achievements, and so on. Interfaces exist for each feature group on each online service that supports it, although specific functions that a service does not support may simply return `false`. This design ensures that developers can write the same code for all online services.

At a higher level, more complex operations use the Online Asynchronous Task Manager to support sequential tasks, or tasks that run on different threads. Asynchronous tasks can describe their dependencies, enabling unrelated tasks to run independently and in parallel, while sequential tasks run serially. All interfaces within the Online Subsystem schedule tasks in this fashion to preserve consistency of operation.

Basic Structure and Usage

The base module, `OnlineSubsystem`, defines and registers service-specific modules with the Engine. During initialization, the Online Subsystem will try to load the default online service module specified in the "Engine.ini" file. All access to an online service will go through this module.

```
[OnlineSubsystem]
DefaultPlatformService=<Default Platform Identifier>

```

Copy full snippet

If successful, the default online subsystem will be available via the static accessor when no parameter is specified.

```
static IOnlineSubsystem* Get(const FName& SubsystemName = NAME_None)


```

Copy full snippet

Additional services load on-demand, when calls to this function request them. Invalid identifiers or failures to load the module will gracefully return `null`.

Interfaces

The following interfaces are included in the Online Subsystem.

> Some interfaces exist only for certain online services, depending on what functionality each service supports.

Interface	Feature Group Description
Achievements	Listing all achievements in a game, unlocking Achievements, and checking your own unlocked Achievements and those of other users.
External UI	Opening the built-in user interfaces for specific hardware platforms or online services. In some cases, services grant access to certain core features exclusively through this interface.
Friends	Anything related to friends and friends lists, such as adding users to your friends list, blocking and unblocking users, and listing players you have recently met online.
Leaderboard	Accessing online leaderboards, including registering your own scores (or times), and checking leaderboards for scores from your friends list or players around the world.
Online User	Collecting metadata about a user.
Presence	Setting the way that a user's online status will appear to other users, such as "Online", "Away", "Playing a game", and so on.
Purchase	Making in-game purchases and reviewing the history of past purchases.
Session	Creating, destroying, and managing online game sessions. Also includes searching for sessions and matchmaking systems.

Interface	Feature Group Description
Store	Retrieving the categories and specific offers available for in-game purchase.
User Cloud	Provides the interface for per user cloud file storage.
Voice Chat (EOS)	Using Epic Online Services as a Voice Chat provider.