Cord

Game Launcher

v1.0a

October 2018

Protocol One

Who Might Be Interested in This Document

This document is mainly addressed to publishers, developers of PC games and technical specialists of these companies.

It describes the main technical aspects and supported functionality of the Cord game launcher of the Protocol One ecosystem. Description of the Ecosystem can be found within the Protocol One Whitepaper document.

Introduction

Cord is a simple and easy-to-use open-source launcher for PC games. We focused on stability and speed of the application, as well as ease of support and adapting to the needs of each developer. Cord can be used to launch one game or a full-scale platform with a large number of games.

Key features:

- **Analytics**. All user behavior data is available out-of-the-box in Google Analytics or Uttu.
- **Stability**. The launcher runs on PC with Microsoft Windows, Apple with MacOS, or Linux with a minimal amount of resources, does not conflict with antiviruses and is updated regularly.
- **Suitable for everything.** Cord can be used to install and update P2P/F2P PC games, create game directories and platforms, or launch Flash browser games.
- **Saving on CDN.** The launcher supports hybrid download and update of games using HTTP, P2P Torrent protocol (TCP + UDP) and binary patching.
- **Out-of-the-box security.** Cord supports integration with Thetta Anticheat out-of-the-box. The publisher gets protection for all games launched in the Cord without the need to integrate a protection system into each game or involve game developers.
- **Pass-through authorization.** A user authorized on the web page will be automatically authorized in the loaded Cord launcher.
- **Interacting with audience.** Cord allows you to display advertisements, announcements, maintenance and update notifications, and news.

Getting Started

Using Cord does not require special skills or in-depth programming knowledge. After downloading the installation package, you can proceed to customize the look and feel of the launcher and the installer using the visual designer. A typical integration process will require:

- Adding the game in the administration interface.
- Customizing the look and the name of the launcher.
- Configuring analytics.
- If required, connecting a solution for accepting payments.

Using Cord is free—we do not and will not charge for it. Additional fees may apply for using separate services, such as Auth One to store user accounts or Pay One if you need to accept payments from users. You will also have to pay separately for any supported CDN for file storage.

Roadmap

Cord is an open solution based on GameNet Launcher, which the Protocol One team has been developing and supporting over the past 8 years. This is a ready-to-use solution that we have been developing and preparing for free use recently. Below is our vision of the next 4 iterations of Cord.

Version	Date	Key tasks
1.0a	December 2018	A completely ready-to-use solution: launcher, installation, accounting and game management.
1.0b	February 2019	Early access for technology partners, integration with Qilin for working with the list of games in the Storefront mode and with Uttu for collecting statistics.
1.0rc	April 2019	Electron-based frontend and a web platform for game distribution with a unified look and feel. A solution that allows users to download and launch games directly from the browser.
1.0r	June 2019	Main release.

Installation

With Cord you can use any solutions for the client installation. We also supply separate installer, lightweight (less than 250kb) and adapted for conflict-free work with the latest antiviruses.

Authorization and Registration

By default, Cord uses AuthOne as a service for registration, authentication and authorization of users. You can use Auth One by installing it on your own hardware or as a service based on Protocol One.

Cord and the installer support pass-through authentication—after registering or authorizing on a web page, the user who downloaded the application will be automatically authorized in the launcher.

Authentication using email and password, authorization without a password, and logging in with external identification providers (Facebook, Vkontakte, Steam, Twitter, Twitch and others) are available.

Game Management

The REST API service based on the Open Game Data Exchange protocol is used for obtaining information about a particular game in Cord Mono mode or a list of games in Cord Storefront mode.

The developer has access to all the necessary tools for adding and configuring the game storefront: design, news, notifications, prices. The Cord package also includes administration tools based on Cord open administration panel or using the Qilin infrastructure.

In Cord, you can distribute PC games and games for web browsers. For the latter, a special mode with a built-in browser based on Chrome with the latest version of Adobe Flash is employed. That way, users will be able to play Flash-games via Cord even after browsers stop supporting the technology.

Modes

Cord supports several visual presentation modes that depend on the number of games that need to be maintained:

- **Mono mode.** The launcher layout is adapted for launching and updating only one game.
- **Storefront.** The launcher layout is optimized for operating a catalog of games, updating and launching them, and managing digital goods.

After downloading the distribution package or installing it from the source code, the developer can independently configure the launcher for use in one of the modes. Modes can be changed at any time, but you will need to update the launcher for the changes to take effect.

Analytics

Google Universal Analytics is integrated into Cord using the Measurement Protocol. Each user click, session information and the ability to set goals are available out-of-the-box.

In addition to Google UA, you can use analytics based on Uttu. This gives you additional features:

- **Interactive marketing.** You can display personalized notifications or advertisements depending on the user portrait and segment.
- **Behavior analysis.** Uttu integration allows you to work directly with behavior scenarios and user preferences.
- **A/B testing.** Uttu integration allows you to set up individual experiments with design to find the optimal conversions into purchase or installation of the game.

When Uttu is integrated, a marketing specialist or analyst can build dynamic scenarios of user interaction based on events sent from the launcher. For example, send a special notice that the game is loaded into the user's messenger, or send a special letter. Launcher events can be used to highlight segments or funnels when analyzing user behavior.

Overlay

Overlay is a feature that allows users to interact with the game launcher without closing the game: communicate with other gamers or receive notifications. The same feature is used to integrate in-game purchases without changing the focus of the game client. The option works for almost all games using DirectX 9, 11, 12, OpenGL and Vulkan.

Cord overlay is compatible with similar solutions, such as Steam, GOG, and Discord, and can be used along with them.

User interaction

Cord enables you to use different scenarios of user interaction:

- Large full-screen notifications at the start of the launcher or on schedule.
- Tray notifications.
- Maintenance start and end notifications.
- Newsfeed.
- Personal notifications.

The Cord administration tool provides a visual script designer for displaying notifications. For example, a marketing specialist can choose notifications to show only to users who have been playing the selected game for longer than a certain time, or just to the paying audience.

After integration with Uttu, a specialist who works with the audience can receive information about the 360° profile of each launcher user, take into account their behavior and send notifications with individual adjustment for the optimal time of display.

If Auth One is used as an account service, notifications can be sent only to users who have completed a confirmed opt-in, or who have a confirmed phone number. Auth One also allows you to send notifications and interact with your audience using instant messengers and push notifications.

Hardware information

Apart from Uttu, users of Cord have access to a service for collecting and storing hardware data and creating reports on the hardware of their users. This is a tool for various teams that helps better understand the capabilities of their players and optimize the games for their players' hardware.

The service allows to collect and view summary data and reports on video card models, processors, number and parameters of monitors, RAM, microphone data, virtual reality helmets and operating systems. Hardware information can also be used to select the optimal game profile for launching and to analyze the audience when purchasing advertising.

Security

Cord supports several options for managing digital rights for games:

- **DRM Lock**. The launcher checks for a license to launch the game using the server API. Licenses are activated by direct purchase of the game in the launcher or by a digital key. A license is checked each time the game is started. This method requires the developer to integrate an authorization SDK into the game.
- **DRM Free**. The launcher doesn't check for a license to launch the game. No internet connection is required.

A significant problem for online games of any genre is the use of special software for modifying the game memory, introducing code and special bot programs that emulate the input of a real user. To solve this problem, Cord supports easy integration with Thetta Anticheat.

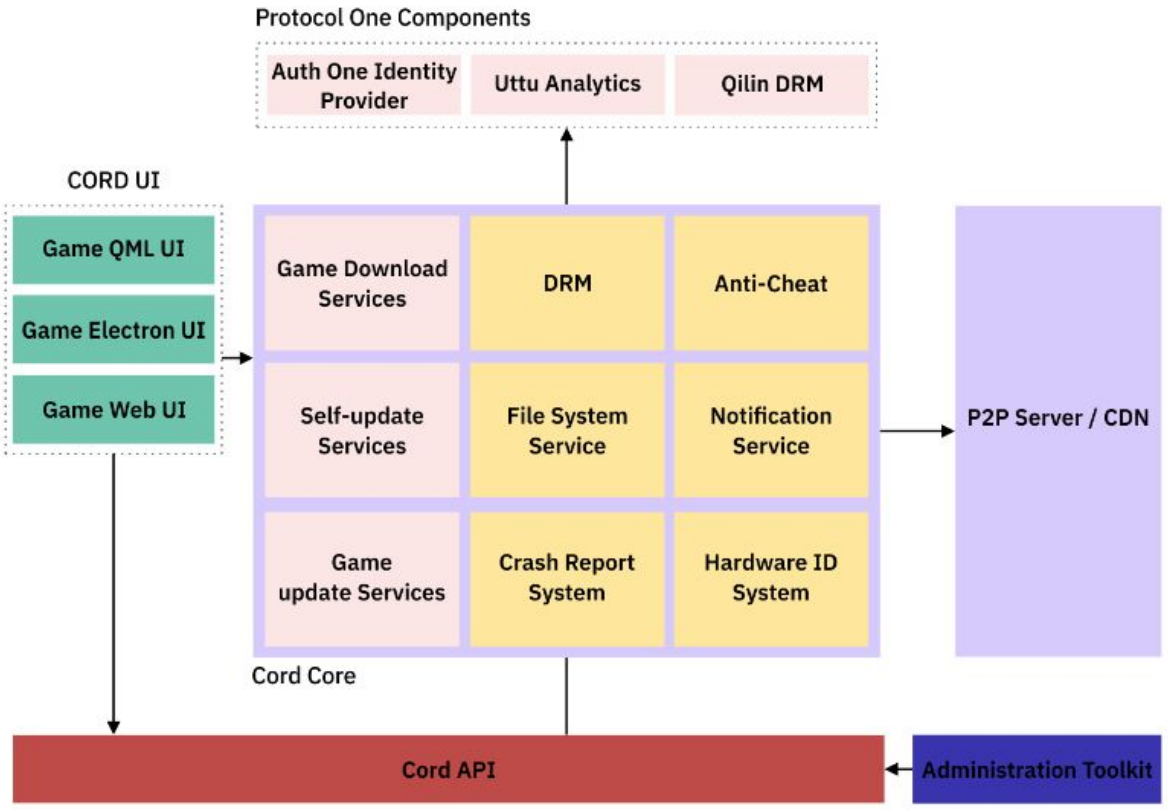
Hardware UID

Cord is able to identify users' PCs with a high level of reliability. The ID of each machine is based on the UUID RFC-4122 or hashed information about the user hardware (motherboard model, hard disk serial number, CPU model).

The hardware ID serves as an additional means of customer identification and can be used to restrict user access to games.

We use several options of look-alike checks and heuristics to identify situations with intentional substitution of hardware data or hardware ID change due to equipment upgrade.

Architecture



The Cord launcher consists of several large parts that work together and, if necessary, can be replaced, upgraded or customized to the needs of each particular game or platform.

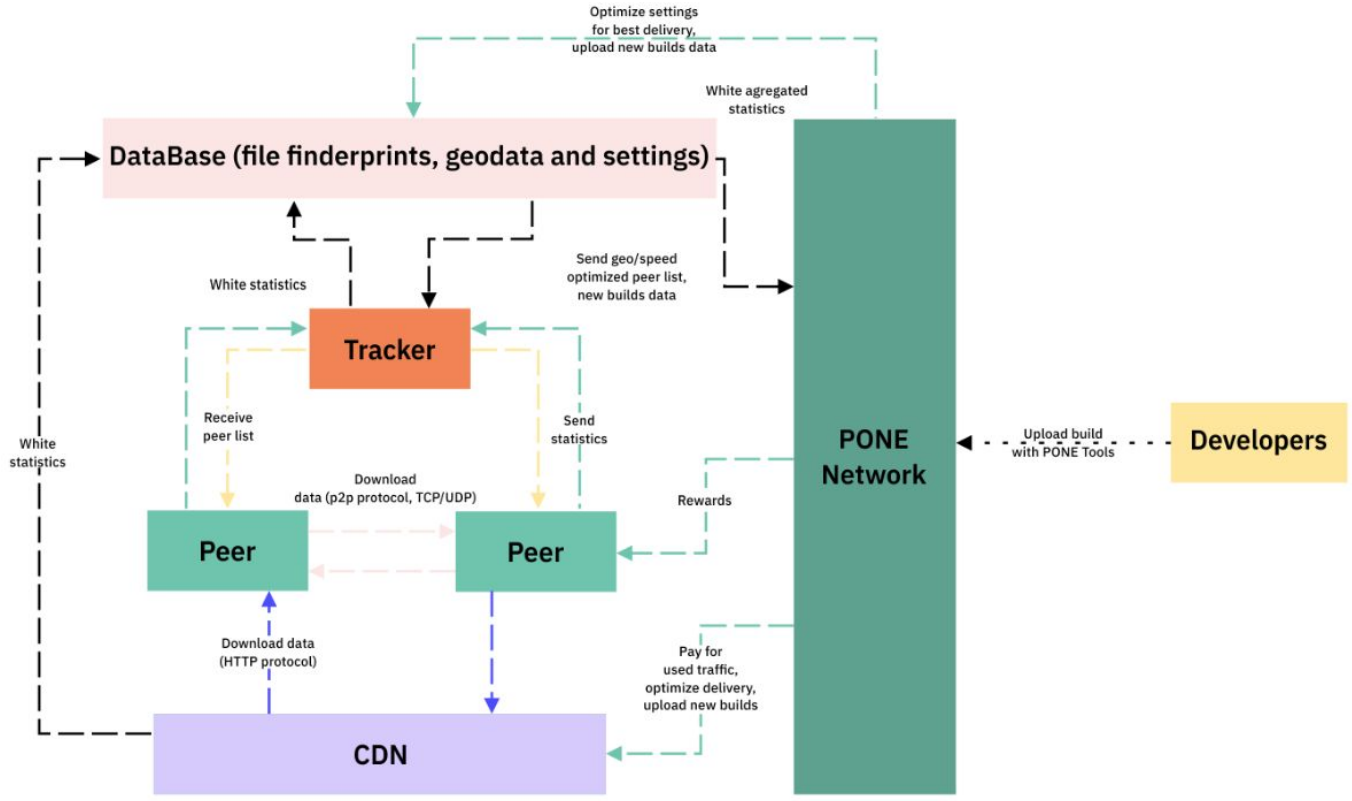
- **Core.** Responsible for downloading and updating games, and receiving information on the game launch. Contains the necessary components for the integration of game access control or anti-cheat systems.
- **Interface.** The end user can take advantage of one of the several interface design options — based on Qt QML, an Electron-based Vue App, or direct integration of a web page and the Cord core.
- **API**. Responsible for storing and managing the settings of the games and the look and feel of the application. It comes with a simple web application for easy administration of this data.
- **Integration**. Tools for shared use of the other Protocol One tools in Cord.
- **P2P tools**. A toolkit for creating and updating game distribution packages, a lightweight BitTorrent Tracker for distributing games using P2P and download-related analytics.

All Cord components come with open source code. All components executed on the server are deployed in a matter of minutes with Docker/Kubernetes and do not require developer involvement.

Game Distribution

Cord comes with a set of tools for uploading, downloading and updating games based on Protocol One or an external CDN. The launcher supports hybrid operation with updates and customer distribution:

- HTTP
- P2P Torrent (TCP+UDP, WebSeed)
- Binary patching
- Archive distribution
- Image distribution



Product

Product is the main object of the distribution system. It is a combination of repositories and add-ons. Each product has a unique application identifier, an associated list of channels and repositories, and a list of related products used as a set of add-ons (DLC).

Channels

A game developer can have several versions of a game for different purposes, for example, a version for testing and a version for users. The use of channels in Cord makes it possible. Channels are used to separate versions of the game for different operating systems, architectures, or languages.

The channels are divided into 3 groups:

- **Default**. There is only one of these. By default, the user downloads and updates the games using this channel.
- **Beta**. An arbitrary number of channels from which any user can download and update games for beta testing.
- **Alpha**. Private channels for developers and testers.

All channels with the exception of Default can be deleted. You can specify the name of the channel for automatic user transfer. By default, all users will be moved to the Default channel.

Channel naming

The main channel (Default) has no alternative names. All other channels are named using a special notation with lowercase characters, underscores and hyphens. This notation allows you to specify the purpose of the channel and the files in it:

- If the channel contains “win” or “windows,” the channel will be available only to Windows platform users. Similarly, the keyword for the Linux platform is “linux” and for Apple MacOS it’s “mac.”
- The “x86” and “amd64” keywords allow you to specify channels based on the CPU architecture type.
- Keyword prefix in the IETF BCP 47 format allows you to manage the division of game builds by language. For example, lang_ru_RU for Russian.

Example of a game build name: *beta-win-x86-1.2.0-AddAssassinClass*—this build will be uploaded onto the public test channel for the Windows x86 platform.

Access management

All users who have the right to download the game have access to the Default channel. A user can take part in testing and switch to any of the created beta channels.

By default, only developers have access to the alpha channels. Access rights can be assigned individually or using access codes.

Repository

A repository is a logical group of files delivered to the user as a whole. An app can have several repositories. Files are downloaded and applied in the order of addition, first for the main product, then for the add-ons. Files in containers should not intersect, i.e. when installing a file, the file added to the previous container should not get overwritten.

Each repository contains a number of mandatory parameters:

- Id. A unique identificator.
- mountPoint. The path for installing files according to the user's basic game directory.
- Folder. The path to the repository files used to create and load the image into the system. All files in this folder are added to the repository by default.
- Ignore. A set of filters for ignoring a part of the files in the specified folder in the ant-style format.
- Mask. A set of filters for adding a part of the files from the specified folder in the ant-style format. By default, this equals **/*.

By default, the repository is downloaded for all users. Some parts of the application may be downloaded depending on the current operating system, architecture or language. For this, a notation similar to the one in channels is used:

- Languages. Supported languages, listed in IETF BCP 47 format.
- OS. Supported operating systems. Options: win, windows, osx, linux.
- Arch. Supported architectures. Options: x86, amd64

Game uploading

Cord uses a separate application CordLoad to upload and update game distributions. CordLoad allows you to view, add and delete channels and repositories, activate builds on the channel, prepare and upload updates using CI. After the build is uploaded, update patches are created automatically.

Disclaimer

This document does not represent a solicitation of investment or any other form of material support for the Protocol One project or Cord. The purpose of this document is to provide a detailed and comprehensive description of Cord, its approaches and principles.

Statements and other information of a declarative nature contained within this document must not be construed as direct assertions or promises unless they are expressly specified as such.

In the current edition, we actively collect feedback from the video game developers and publishers, stores and platforms to improve the specifications and make the product we create easier and more convenient for all participants.

Appendix

P2P authorization protocol

The Cord authorization protocol extends the original TCP BitTorrent Peer protocol. We supplemented the authorization protocol to prevent unauthorized downloading of games via the torrent protocol.

“User” is a person who wants to download some content from the “server.” By “server” we mean the term “torrent server.” After registration and verification on this server, the user can download the data. The process is described in this section.

Handshake

Handshake is a mandatory first packet sent by the client. Each packet consists of <pstrlen><pstr><reserved><info_hash><peer_id>, where:

- pstrlen: the length of the <pstr> string, one byte
- pstr: protocol string and ID
- reserved: eight (8) bytes All current implementations use zeros. Each bit in these bytes can be used to change the protocol’s behavior.
- info_hash: A 20-byte SHA1 hash info key in torrent metafile. A similar info_hash is transmitted in all requests to the tracker.
- peer_id: A 20-byte string with a unique user ID.

User verification

Interaction with the server begins with sending the handshake packet. The “check resource” procedure is used to identify the user and check permissions to download the data with the

specified info_hash. After that, peers exchange messages with the length prefix. If verification fails, the server terminates the connection.

A handshake packet is sent in accordance with the bittorrent specification. We use the “PONE_” + “user id” format.

After receiving the handshake packet, the server retrieves the “user id” from <pstr> and checks whether the user is allowed to download the resource associated with the <info_hash>.