



WHITE PAPER

XSOLLA LAUNCHER PRODUCT SPECIFICATION

v.1.0

CONTENTS

Introduction	3
Glossary	3
What is Xsolla?	3
What is Xsolla Launcher?	4
What is this document for?	4
What to get before?	4
Xsolla Launcher Architecture	5
Main configurations	7
Game delivery system	7
Game distribution system	10
UI customization	13
Authentication window	13
Related Xsolla Products	18
Xsolla Publisher Account	18
Xsolla Login	18
Xsolla Buy Button	19
References & Links	20



INTRODUCTION

Glossary

- **Players, end users, users**
The target audience for Xsolla Launcher.
- **Partners, game developers, publishers**
The game content providers.
- **Game, project**
The content delivered to end users of Xsolla Launcher.
- **Xsolla Launcher, Launcher**
The names of the product for which this document is intended.
- **CDN (content distribution network)**
A geographically distributed network of proxy servers and their data centers. Used to deliver games and their updates to users.
- **P2P (peer-to-peer) connection**
A distributed application architecture that partitions tasks or workloads between peers. Used to deliver games and their updates to users.
- **P2P announcer**
Provides meta-information data for the peer.
- **P2P web seed, seed.**
A computer that uploads one or more files to a file sharing network.
- **P2P peer, peer**
A computer that downloads one or more files from a file sharing network.

What is Xsolla?

Xsolla is the video game business engine. We provide game developers and publishers with payment, billing, distribution, and marketing tools. Xsolla is here to ensure the best ideas get made, get seen, and get played. To do that, we have built a suite of products that remove the barriers between game creators and the support they need, enabling an ecosystem where every industry player can connect to get our best work done.



What is Xsolla Launcher?

Xsolla Launcher is a customizable delivery tool for partners who develop video games. The product optimizes content delivery costs while centralizing modules for everything from virtual currency to news and banners. Main Xsolla Launcher features are:

- Multigame support with several distribution ways
- In-game store
- Authentication window
- Customizable UI localized into 20 languages
- Windows support
- Automated delivery of Launcher updates and games with P2P/CDN
- Convenient setup via Xsolla Publisher Account UI

What is this document for?

Use this document to get details about Launcher technical implementation and check it for compatibility with your games. It describes the main Xsolla Launcher components, features, and their technical realization.

What to get before?

1. [Xsolla Publisher Account](#)¹ to configure Xsolla Launcher.
2. [Xsolla Launcher integration guide](#)² to learn more about Xsolla Launcher integration.
3. [Xsolla Build Loader](#)³ to operate with game builds.

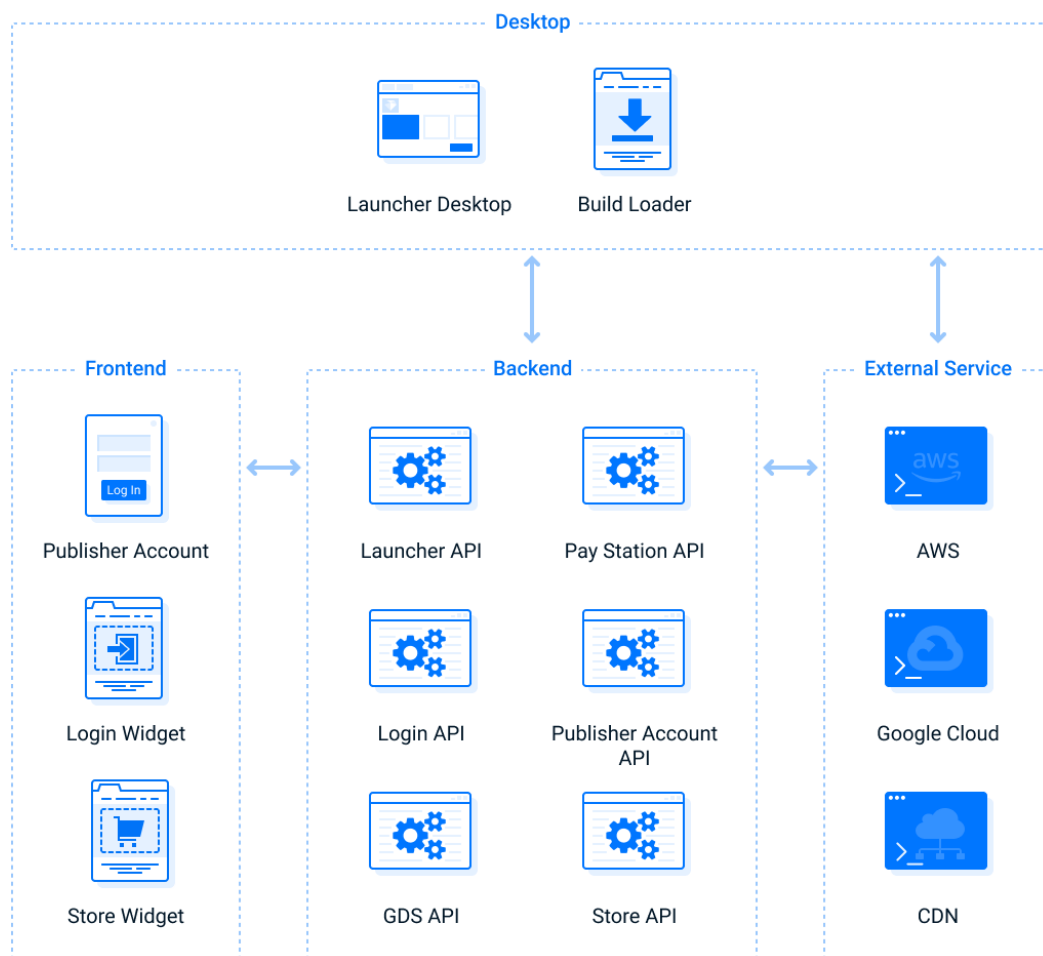


XSOLLA LAUNCHER ARCHITECTURE

Xsolla Launcher is a whole system connecting user interface, API methods, utilities, external services, and other Xsolla products. The system consists of the following functional components:

- **Main configurations**
Initial setup of Xsolla Launcher required for the system functioning.
- **Game delivery system**
Adding games and their updates to Xsolla Launcher.
- **Game distribution system**
Monetization of your games in Xsolla Launcher.
- **UI customization**
Changing themes, fonts, and images in Xsolla Launcher.
- **Authentication system**
Authentication in Xsolla Launcher.

The architecture of Xsolla Launcher is represented by the technical elements on the scheme below:



The backend of the system is mostly written in GO and partially in PHP. The backend elements used in the system include the following Xsolla APIs:

- **Publisher Account API**
Methods to get information about projects in Publisher Account.
- **Launcher API**
Methods to get settings from Publisher Account.
- **GDS API (Game Delivery System API)**
Methods to get information about game builds from the Xsolla database.
- **Login API**
Methods to register and authenticate users in Xsolla Launcher, as well as allow them to change their data.
- **Store API**
Methods to get information about virtual currency, virtual items and other Buy Button modules.
- **Pay Station API**
Methods to process game purchases and in-game transactions.

Desktop elements used in the system:

- **Launcher Desktop**
A C++ desktop app that uses the Qt framework. End users interact with this app.
- **Xsolla Build Loader**
A command-line utility used to load game builds to the file storage and add them to Launcher Desktop.

Frontend elements used in the system:

- **Xsolla Publisher Account**
The interface for Xsolla Launcher setup.
- **Xsolla Login Widget**
The interface for user registration and password recovery. It is displayed in a browser.
- **Xsolla Buy Button Widget**
The interface to purchase games and in-game goods.

External services used in the system:

- **Google Cloud**
The cloud to save user data.
- **AWS**
Amazon Web Services. They include a file storage for game builds, CDN, and elements of P2P connection.
- **CDN**
Content distribution networks for delivery of games and their updates to end users.



The process of setup is described below in sections named as Xsolla Launcher functional components. You will find further details on getting and sending data via APIs, connection with external services, and other technical information there.

Main configurations

To start Xsolla Launcher integration, you need to create your Launcher instance in [Xsolla Publisher Account](#), specifying the following configurations:

- Launcher name and UI languages.
- Authentication methods.
- Games delivered via Xsolla Launcher. The list of available games is formed by getting your projects via Publisher Account API.

Each Launcher instance has its own ID generated upon creation. This ID is used in API methods to identify the Launcher instance created in Xsolla Publisher Account.

Configurations saved in Xsolla Publisher Account are written to the Xsolla database via Launcher API methods with a Launcher instance ID. Launcher API methods are also used to get configurations from Xsolla Publisher Account when Launcher Desktop is started by the end user.

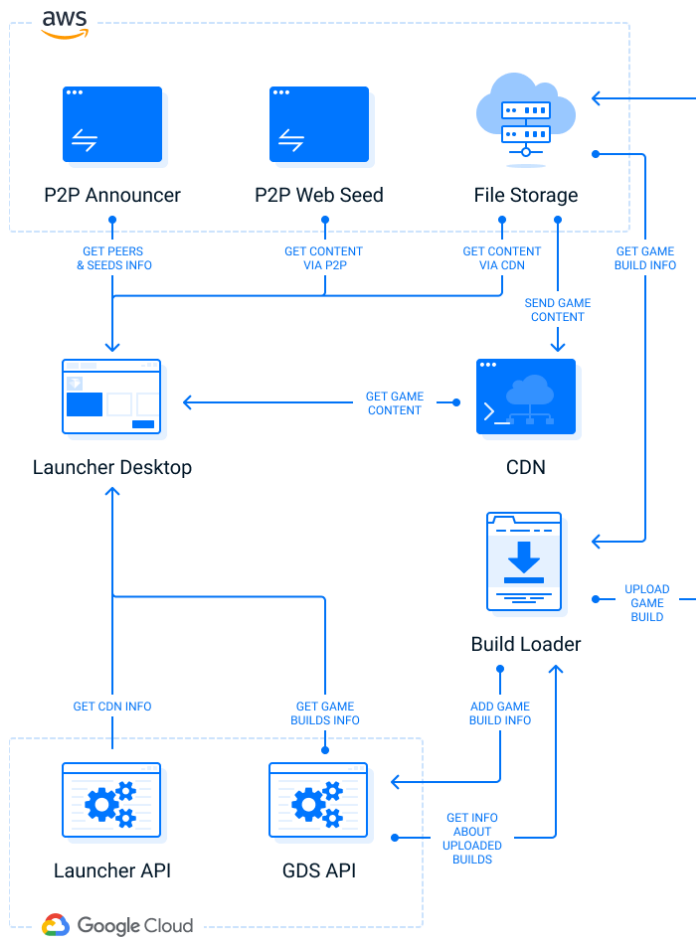
Game delivery system

The main Xsolla Launcher function is to deliver your games and their updates to players. In this case, Xsolla Launcher acts as a container for your games. To add a game to Launcher, you need to:

- Configure Launcher in Publisher Account
- Download Build Loader
- Generate the game build

Before the game is displayed in Xsolla Launcher and players can access it, the game build must be uploaded to the file storage via Build Loader. The data from the file storage is sent to Xsolla Launcher via P2P/CDN. The common game delivery system is presented on the scheme below.





Placing the game build to Launcher

Build Loader is a command-line utility used to send the game build to the Amazon file storage. The game build must be generated and sent to the file storage every time you update the game. It is essential for delivering game updates to players.

Build Loader generates archives for each game build:

- with a full game when the build is uploaded for the first time;
- or with a patch of updates when the build is uploaded another time.

The generated archive is sent to the file storage via AWS SDK for C++ in several threads. After that, Build Loader sends a request to the file storage to get information about the uploaded archive: its size, a CDN link, and signatures. This information is sent to the GDS API and written to the Xsolla database. This process can be automated and embedded into a CI/CD system.

Publisher Account lets you manage game builds. It sends a request to get builds information from Amazon and displays the result in the interface. Game builds can be placed to the **master** or **test** stage in Publisher Account. When the build is placed to the stage, the number of the game version is increased by 1.



- **Master stage**
Games are available to all players who have Xsolla Launcher installed.
- **Test stage**
Games are available to players who have Xsolla Launcher installed and started it with the `-game_stage` test parameter.

When Xsolla Launcher is started, it sends a request to Launcher API to get information about games uploaded with the current Xsolla Launcher ID. These games are displayed in Launcher with the **Install** button until the player installs it. When the game is installed, the **.version** file is created in its folder. The variable with the game version is written to this file.

Whenever Xsolla Launcher with an already installed game is started next time, it performs a check of the game version. If the version of the installed game is lower than the version of the last game build in the Xsolla database, the **Update** button is displayed on the game page in Xsolla Launcher. After the player updates the game, its version is increased in the **.version** file.

Binary patching

Binary patching means players only download the changed parts of game files. It enables:

- smaller downloads;
- faster game delivery and updates;
- faster update unpacking;
- ability to return to the previous game version with a reverse patch.

Patches are generated by the Build Loader command-line utility for updated game files. When a new game build is uploaded via Build Loader, the utility sends a request to GDS API to get information about the previously uploaded game builds. The difference between an already uploaded build and a new one is calculated. The generated patch is the difference between these two builds. The patch also includes the instruction on managing game files for Launcher Desktop. The generated patch is uploaded to the file storage via AWS SDK for C++ in several threads. After that, Build Loader sends a request to the file storage to get information about the uploaded archive: its size, a CDN link, and signatures. This information is sent to the GDS API and written to the Xsolla database.

The generated patch is uploaded to the file storage via AWS SDK for C++ in several threads. After that, Build Loader sends a request to the file storage to get information about the uploaded archive: its size, a CDN link, and signatures. This information is sent to the GDS API and written to the Xsolla database.

Reverse patches are generated simultaneously with update patches. Reverse patches are used to return to the previous game version. They contain an updated part of game files and instructions on how to turn the game version back. Reverse patches are applied when you place the previous game version to the **master** stage.



Getting game content from Launcher

The game content is downloaded from Launcher to a player's PC via a CDN/P2P. These connections help to make the download faster and more reliable. A CDN is used by default, but you can also configure a P2P connection.

CDN

When Xsolla Launcher is started, it sends a request to Launcher API to get information about the configured CDN. A free [Amazon CDN](#)⁴ is configured by default. You can use it or set up a paid [G-Core Labs CDN](#)⁵ to make content delivery faster.

P2P Connection

Content can be delivered to users via a P2P connection. It is set up via Xsolla Launcher configuration files. In this case, the **libtorrent** library and **AWS** are used, so that:

- when Launcher Desktop is started, torrent files are downloaded to a user PC first, followed by the game content;
- content is delivered and peers are defined via the Amazon P2P web seed and a P2P announcer.

To optimize game delivery costs, Xsolla Launcher uses the CDN/P2P balancing system. If there are no free seeds or the P2P connection has failed, Xsolla Launcher switches to a CDN.

Note: Regardless of the game delivery method, game content download starts from the point of failure if it has already been initiated. Xsolla Launcher performs a check for the content that has already been downloaded to the player's PC and installs only new content.

Game distribution system

Most game distribution settings are performed in Xsolla Publisher Account and recorded to the Xsolla database. These settings allow you to distribute two game types:

- **Free-to-play games**
The user does not need to make a purchase to access the game. After Launcher is installed, the user is able to play the game right away.
- **Premium games**
The user must make a purchase to access the game. After the game is purchased, the game key value is added to the user record in the Xsolla database.

Note: There are two types of game keys activation in premium games: manual and automatic. You can find more information about game keys in [Buy Button documentation](#)⁶.



After Xsolla Launcher is started by the player, the game types are indicated via Launcher API (get game settings from Publisher Account).

Free-to-play games

There are two conditions for the game to be free-to-play:

1. **Enable quick-buy** in Publisher Account is disabled.
2. **Enable manual activation of game keys** in Publisher Account is disabled.

If both of them are met, the game is available for installation and the **Install** button is displayed on the game page in Xsolla Launcher.

Premium games

If at least one condition below is met, the game is considered to be premium:

- **Enable quick-buy** in Publisher Account is enabled.
- **Enable manual activation of game keys** in Publisher Account is enabled.

As it is described above, keys for premium games can be activated manually or automatically.

Manual keys activation

You can send game keys to your users after the game is purchased or sent as a gift. In this case, these keys must be activated manually. To use manual activation, **Enable quick-buy** in Publisher Account. A special field for key input will be displayed on the game page in Xsolla Launcher. Your users will be able to input the received key and get access to the game. After the key is activated, the Play button is available on the game page, and the key value is added to the user record in the Xsolla database.

Automatic keys activation

Automatic game keys activation implies that after the game is purchased, the key is automatically activated and its value is added to the user record in the Xsolla database. To use automatic key activation, disable the **Enable manual activation of game keys** module and enable the **Show Store in Launcher** module in Publisher Account.

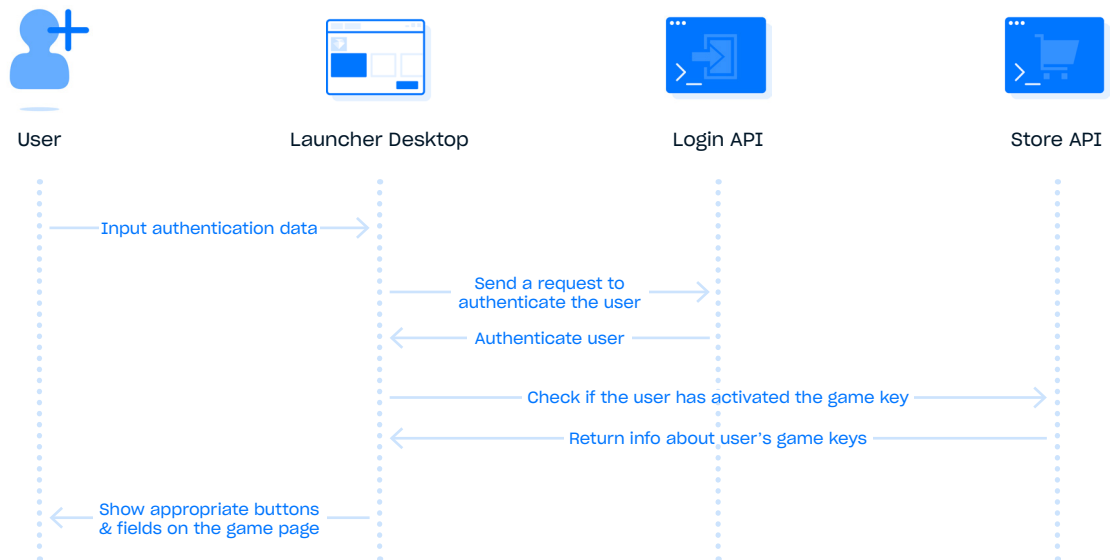
After the game key has been automatically activated, the game is available for installation. If the user has not purchased the game yet, but it can be purchased via Launcher, the Buy button is displayed on the game page in Xsolla Launcher.

The user can buy the game not only via Xsolla Launcher, but also on the game website. In this case, after the game key has been automatically activated, the user opened Xsolla Launcher and authenticated, they will skip the purchasing step and start playing the game right away.



Note: Both manual and automatic game keys activation can be set up for a single premium game. You can find instructions on manual and automatic activation [here](#).

Depending on the way in which the user purchased the game, their flow in Xsolla Launcher can be different. However, the way Xsolla Launcher checks if the user has activated the game key or not is the same for both (see the scheme).



When the user is authenticated, Launcher Desktop sends a request to Store API to check if the game key is linked to this user.

In case the game key is linked to the user, the **Install** button is displayed in the game window in Xsolla Launcher.

In case the game key is NOT linked to the user, the **Buy** button is displayed. It can also be the **Activate key** field if you distribute games with manual keys activation.

The following APIs are used in this process:

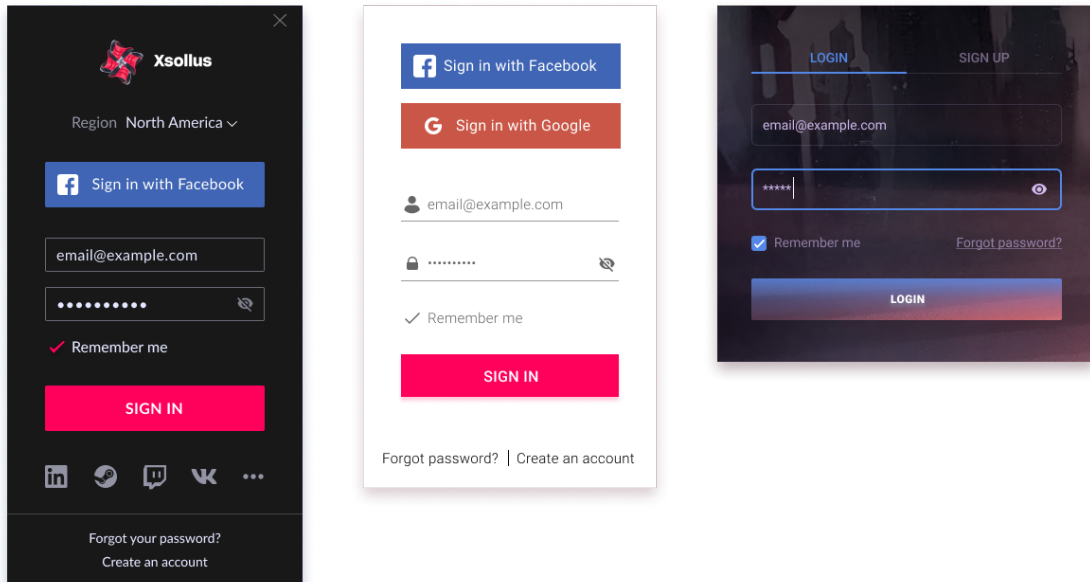
- Login API to get user data
- Store API to activate the game key when the user bought the game, to check if the game key is linked to the user
- Pay Station API to buy the game



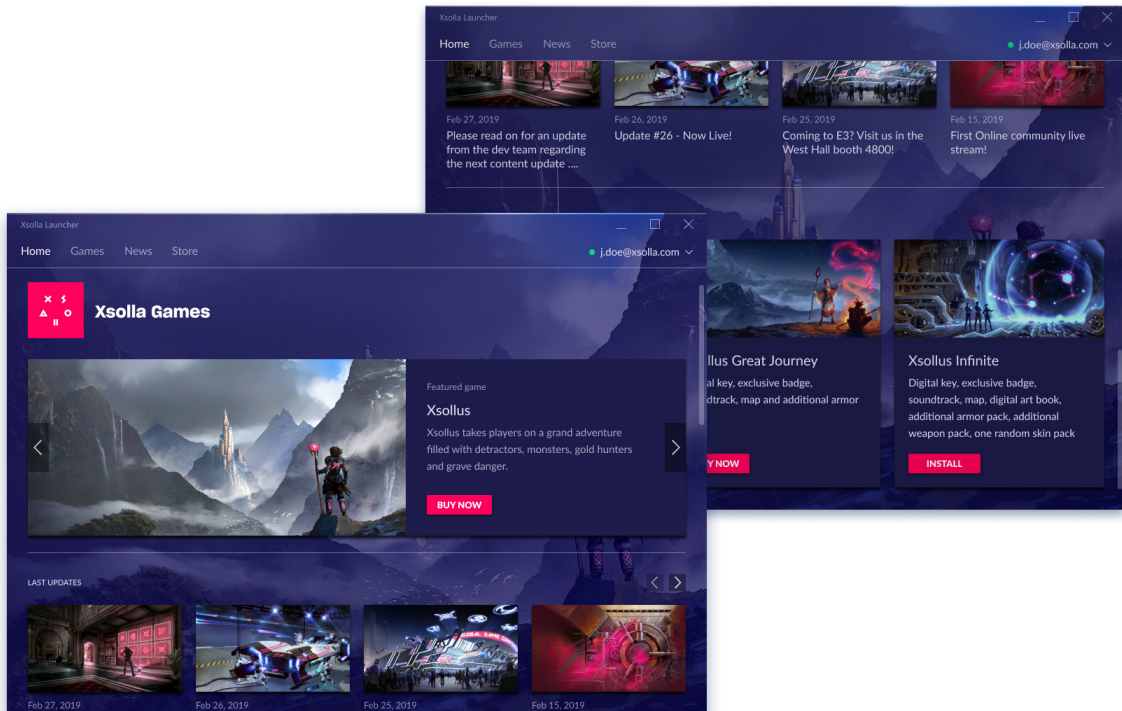
UI customization

Launcher Desktop UI elements are sections, windows, banners, etc. Some of them are presented below:

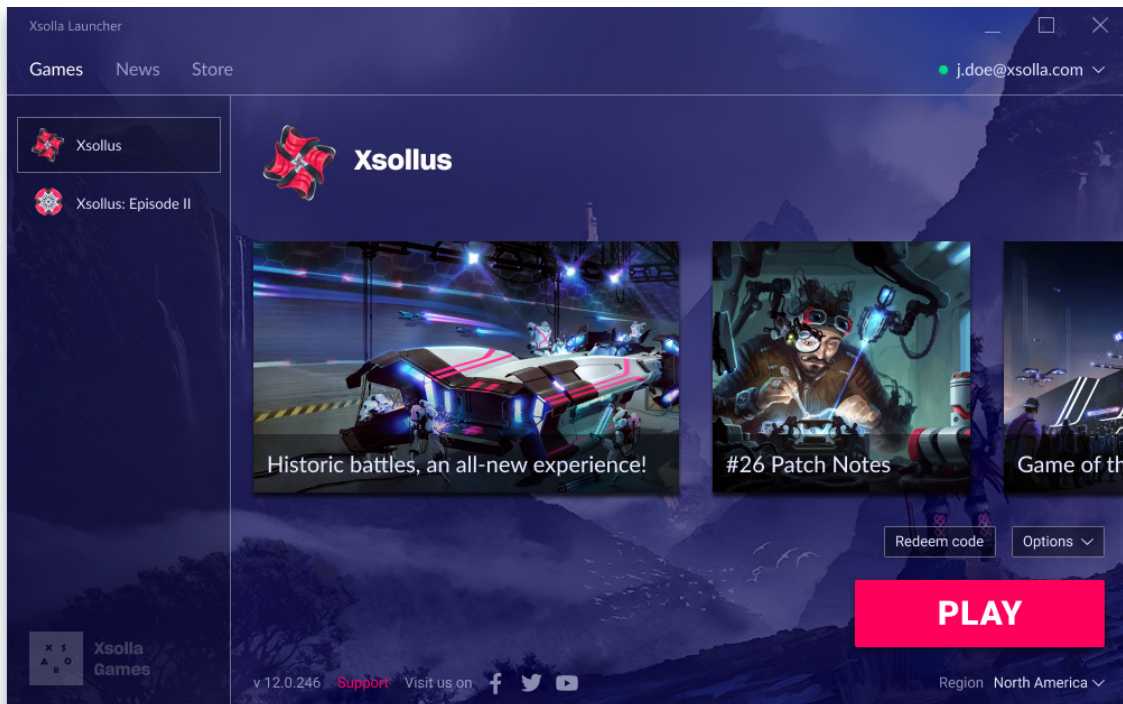
Authentication window



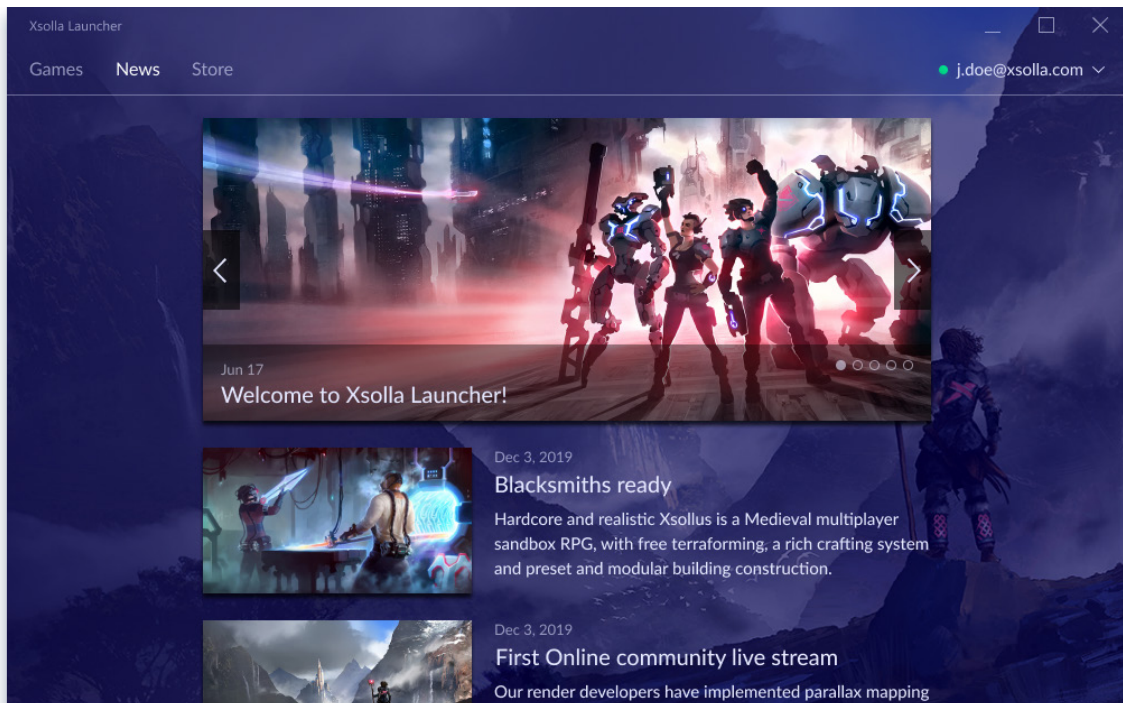
Start page



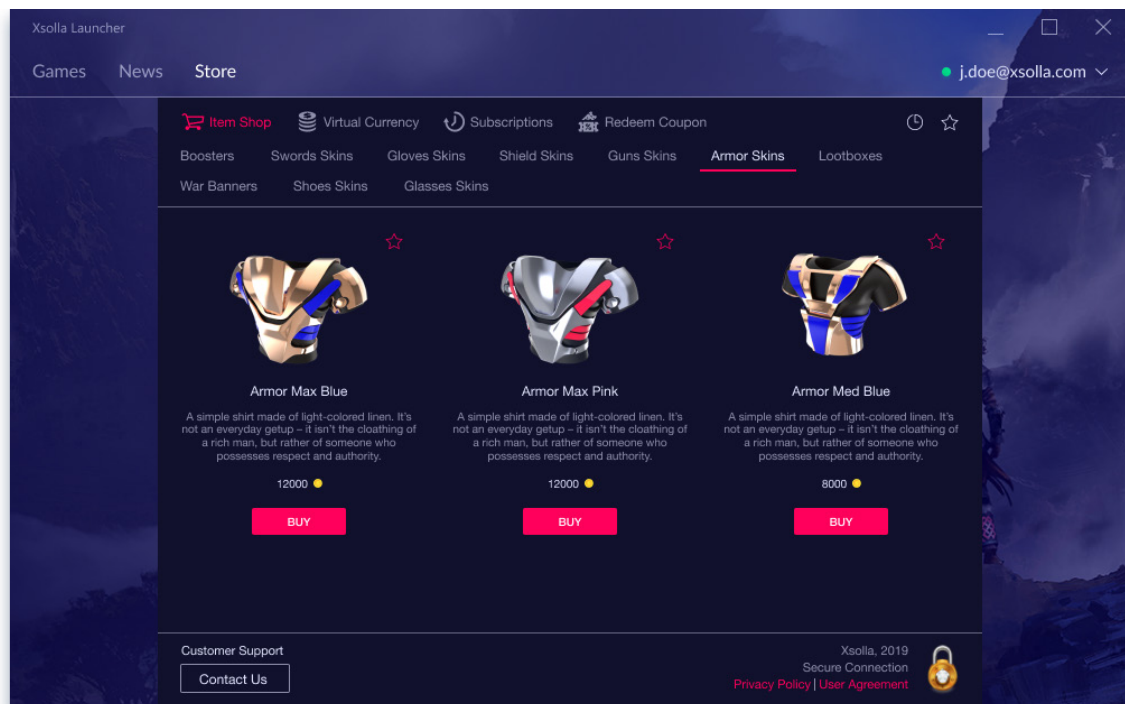
Game page



News page



In-game store



The Launcher Desktop UI can be fully customized. You can change images, fonts and the Launcher Desktop color scheme via Xsolla Launcher configuration files: *config.json* and *UIStyles.json*. The instruction on UI customization is available [here](#)⁹.

After you have changed these parameters, generate an Xsolla Launcher installer to check the result of customization. Xsolla Launcher uses the **NSIS** tools to generate a Windows installer in the EXE format with setup access rights. The instruction on how to generate the installer is available [here](#)⁹.

Authentication system

You can enable authentication in Xsolla Launcher. It will give your users access to the games after they enter their data. Xsolla Launcher uses Xsolla Login API methods for user authentication and has its own visual form.

User actions during the authentication process:

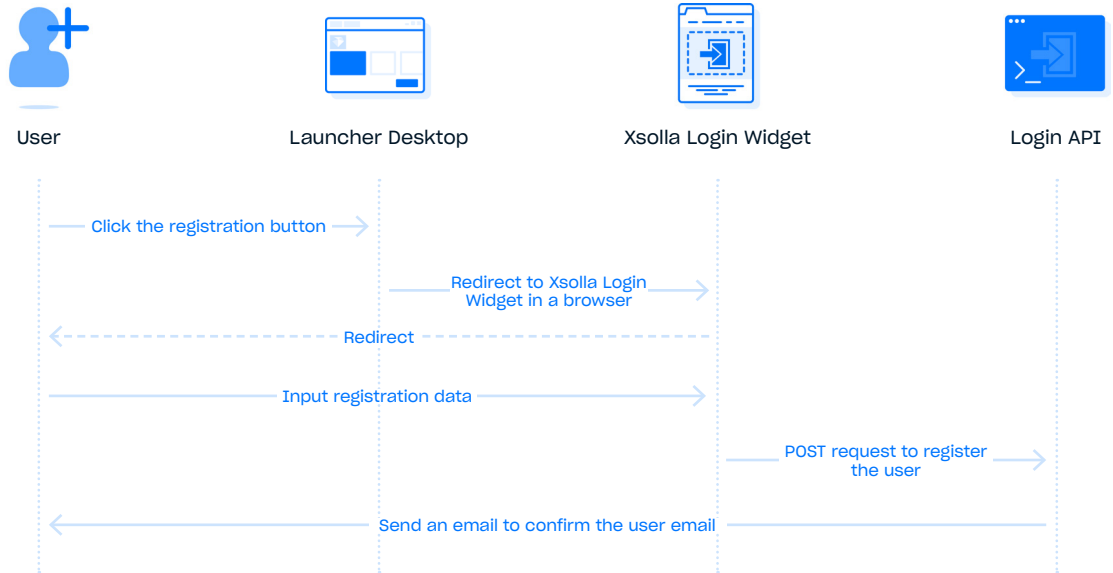
- Registration
- Authentication via username/password and social networks
- Password recovery

User data is saved to the Xsolla database and Google Cloud, including information about the in-game progress. It allows your users to continue the game where they left off.

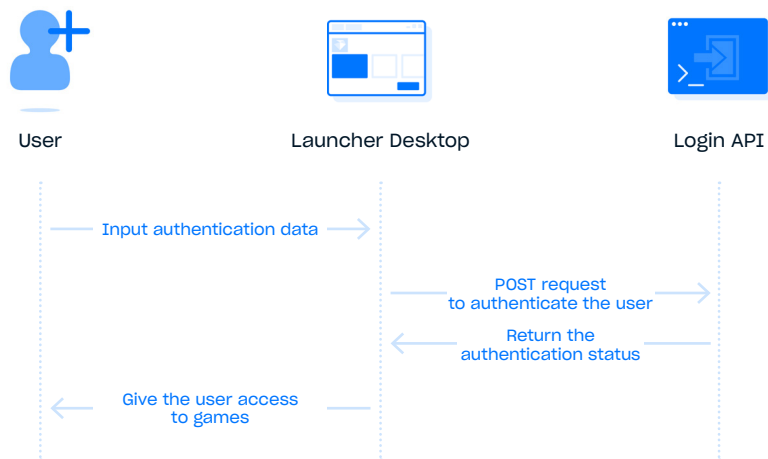


Below you will find schemes on the Xsolla Launcher system functioning during the authentication process.

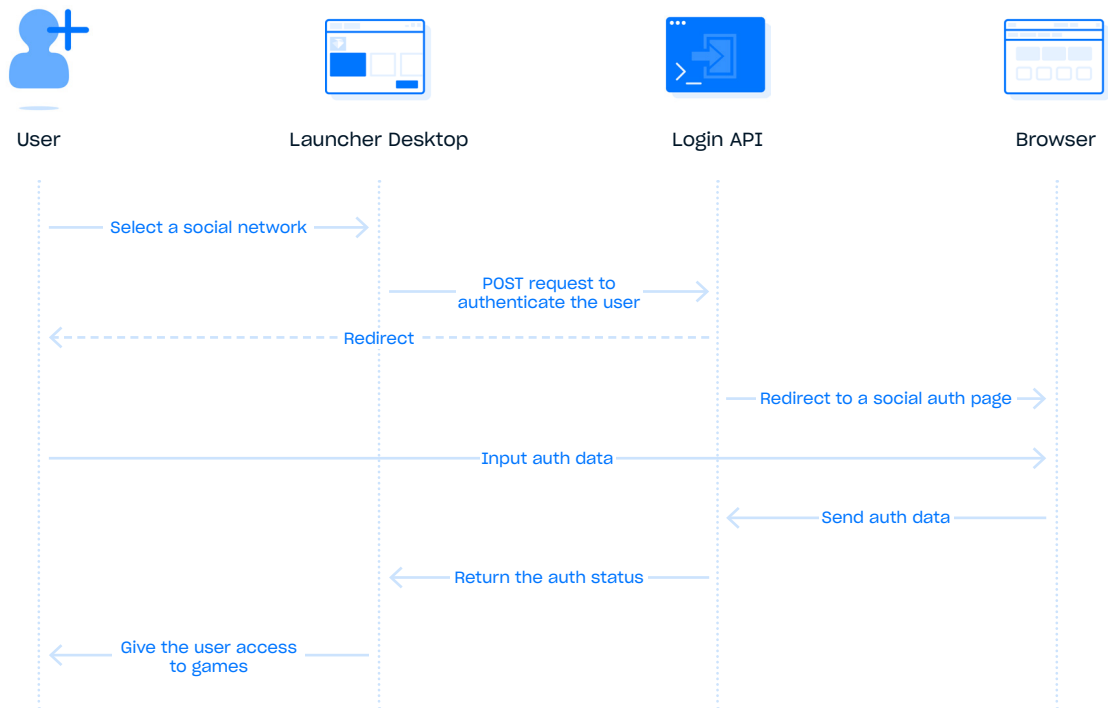
Registration



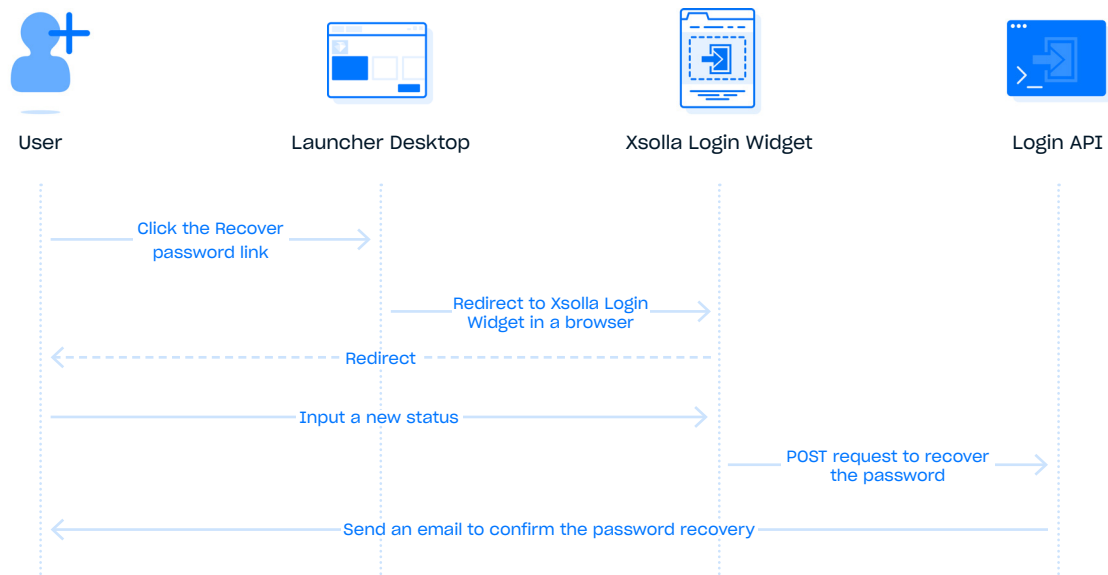
Authentication via username/password



Authentication via a social network



Recovering the password



RELATED XSOLLA PRODUCTS

Xsolla is an ecosystem with several products connected to each other. As described above, Xsolla Launcher supports integration with other Xsolla products, and one of them is used as an interface to set up Xsolla Launcher.

Xsolla Publisher Account

Xsolla Publisher Account is the interface for main Xsolla Launcher setup. All settings are sent to Launcher Desktop via Launcher API. The interface lets you manage the following settings:

1. Main configurations
 - Launcher name
 - Launcher interface languages
 - Authentication methods
 - Delivered games
2. Xsolla Launcher modules
 - News (see UI Customization to know how they are displayed in Launcher Desktop)
 - Banners (see UI Customization to know how they are displayed in Launcher Desktop)
 - In-game store and its modules (see [documentation](#)¹⁰)
3. Visualized statistics on Xsolla Launcher performance. It is collected via Google Analytics and displayed on your Launcher statistics page.
4. Authentication (namely, Xsolla Login setup).

Xsolla Login

Xsolla Login is a single sign-on tool that authenticates and secures user passwords on behalf of partners who develop video games. It creates a seamless, one-click user registration experience through 30+ third-party authentication providers that offer players convenient, safe, and fast methods for signing up or logging in to all of their favorite games.

Xsolla Launcher uses Xsolla Login API methods and Xsolla Login Widget for user registration and password recovery. Xsolla Login API methods are also used for authentication. You can customize the authentication interface in the same way as other elements of Launcher Desktop. The data between Launcher and Login is transferred via Login API and Launcher API.

See [Developer Documentation](#)¹¹ to get more information about Xsolla Login.



Xsolla Buy Button

Xsolla Buy Button lets you optimize your game's earning potential by selling everything from game keys to subscriptions directly to players.

Xsolla Buy Button is used in Xsolla Launcher to activate keys for the games added to Xsolla Launcher. It helps you monetize your games via an in-game store with virtual items and virtual currency that you can set up in your Xsolla Publisher Account.

See [Developer Documentation](#)¹⁰ to get more information about Xsolla Buy Button.

Summary

Xsolla Launcher is a constantly developing product. We work on its improvement and pay close attention to all feature requests. Integration with other Xsolla products will give you an opportunity to fully customize the solution. Learn more about other Xsolla products at xsolla.com or contact our integration team at integration@xsolla.com.



REFERENCES & LINKS

1. **Publisher Account:** <https://publisher.xsolla.com/>
2. **Xsolla Launcher Developer Docs:** <https://developers.xsolla.com/doc/launcher/>
3. **Launcher Build Loader:** https://cdn.xsolla.net/launcher/build/Partners/build_loader.zip
4. **Amazon CDN:** <https://aws.amazon.com/cloudfront>
5. **G-Core Labs CDN:** <https://gcorelabs.com>
6. **Xsolla Game Keys Developer Docs:** https://developers.xsolla.com/doc/buy-button/#guides_buy_button_monetization_options_game_keys
7. **Xsolla Launcher Keys Activation Guide:** <https://developers.xsolla.com/recipes/launcher/game-distribution-via-launcher/>
8. **Xsolla Launcher UI Customization Guide:** https://developers.xsolla.com/doc/launcher/#guides_launcher_ui_customization
9. **Xsolla Launcher Installer Guide:** https://developers.xsolla.com/doc/launcher/#guides_launcher_generate_archive_installation_file
10. **Xsolla Buy Button Developer Docs:** <https://developers.xsolla.com/doc/buy-button/>
11. **Xsolla Login Developer Docs:** <https://developers.xsolla.com/doc/login/>





Enjoy the game.