[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/academy/3-ibc/8-ica.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/academy/3-ibc/8-ica.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 0 - Getting Started](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Getting Started](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Blockchain 101](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Blockchain History](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Public and Managed Blockchains](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Consensus in Distributed Networks](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Cryptography](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Self-Assessment Quiz](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Go Introduction - First Steps](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Go Basics](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Go Interfaces](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Control Structures in Go](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Arrays and Slices in Go](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Standard Packages in Go](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Concurrency in Go](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Docker Introduction](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Introduction to the Interchain](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[The Interchain Ecosystem](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Getting ATOM and Staking It](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[A Blockchain App Architecture](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Accounts](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Transactions](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Messages](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Modules](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Protobuf](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Multistore and Keepers](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[BaseApp](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Queries](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Events](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Context](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Testing](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Relaying with IBC](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Interchain Security](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Bridges](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Migrations](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 1 Quiz](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 2 - First Steps](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[First Steps](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Setup Your Work Environment](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Run a Node, API, and CLI](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Ignite CLI](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Store Object](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Create Custom Messages](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Create and Save a Game Properly](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Add a Way to Make a Move](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Emit Game Information](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Record the Game Winner](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 2 Exercise](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[What is IBC?](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[IBC Token Transfer](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[IBC Tooling](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[What is CosmJS?](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Your First CosmJS Actions](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Compose Complex Transactions](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Learn to Integrate Keplr](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Put Your Games in Order](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Auto-Expiring Games](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Let Players Set a Wager](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Handle wager payments](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Integration tests](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Incentivize Players](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Help Find a Correct Move](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Understand IBC Denoms](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Go Relayer](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Hermes Relayer](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[CosmJS Advanced](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Create Custom Objects](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Create Custom Messages](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Get an External GUI](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Backend Script for Game Indexing](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[IBC Deep Dive](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[IBC Application Developer Introduction](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Create a Leaderboard Chain](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Run in Production](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Prepare the Software to Run](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Prepare a Validator and Keys](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Prepare Where the Node Starts](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Prepare and Do Migrations](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Simulate Production in Docker](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Tally Player Info After Production](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Simulate a Migration in Docker](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Final Exam](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[What's Next?](https://ida.interchain.io/academy/3-ibc/8-ica.html)

[Continue Your Interchain Journey](https://ida.interchain.io/academy/3-ibc/8-ica.html)

Docs Version Switcher

On this page

[What are Interchain Accounts?](https://ida.interchain.io/academy/3-ibc/8-ica.html#what-are-interchain-accounts)

[ICA core functionality: controller & host](https://ida.interchain.io/academy/3-ibc/8-ica.html#ica-core-functionality-controller-host)

[Controller API](https://ida.interchain.io/academy/3-ibc/8-ica.html#controller-api)

[Host API](https://ida.interchain.io/academy/3-ibc/8-ica.html#host-api)

[Authentication](https://ida.interchain.io/academy/3-ibc/8-ica.html#authentication)

[Application callbacks](https://ida.interchain.io/academy/3-ibc/8-ica.html#application-callbacks)

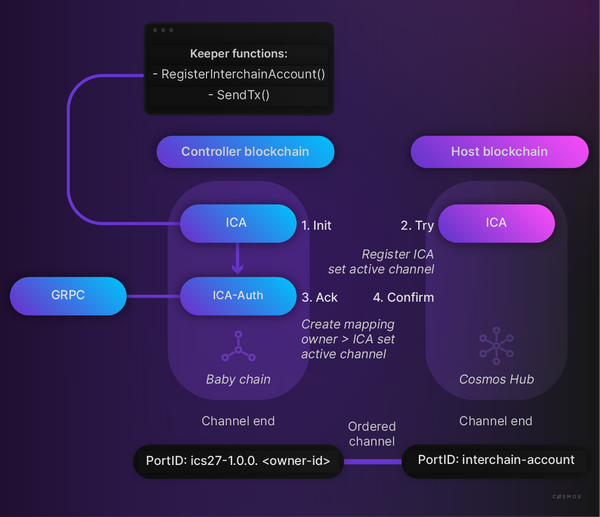
[Practical exercise](https://ida.interchain.io/academy/3-ibc/8-ica.html#practical-exercise)

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#interchain-accounts) **Interchain Accounts**



**Interchain accounts (ICAs)** allow you to control an account on a **host chain** from a **controller chain**.   
  
In this section, you will learn more about:

* Host chains and controller chains
* ICA (sub)module(s)
* The authentication module for custom authentication
* ADR 008 middleware for secondary application logic



[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#what-are-interchain-accounts) What are Interchain Accounts?

The interoperable internet of blockchains made possible by IBC opens up many new frontiers for cross-chain interactions, and for applications leveraging these primitives. In this interoperability narrative, it should be possible to interact with a given chain (call it the *host chain*) through a remote interface, i.e. from another chain (the *controller chain*). Interchain accounts, or ICA for short, enable just that: they allow for a chain, a module, or a user on that chain to programmatically control an account (the Interchain account) on a remote chain.

Sometimes Interchain accounts are referred to as *cross-chain writes*. This is in conjunction with Interchain queries (ICQ) or the ability to read data from a remote chain, i.e. *cross-chain reads*.

The specification describing the Interchain accounts application protocol is [ICS-27 (opens new window)↗](https://github.com/cosmos/ibc/tree/main/spec/app/ics-027-interchain-accounts).



The ibc-go implementation of ICA can be found [in the apps sub-directory (opens new window)↗](https://github.com/cosmos/ibc-go/tree/main/modules/apps/27-interchain-accounts).   
  
The corresponding documentation can be found in the [ibc-go docs (opens new window)↗](https://ibc.cosmos.network/main/apps/interchain-accounts/overview.html).

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#ica-core-functionality-controller-host) ICA core functionality: controller & host

From the description above, a distinction needs to be made between the so-called "host" and "controller" chains. Unlike ICS-20, which is inherently bi-directional in the sense that both chains can use the transfer module to send and receive tokens, ICA has a more unidirectional design. Only the controller chain can send executable logic over the channel, which will then always be executed on the host chain.

Several relevant definitions relating to ICA are as follows:

**Host Chain:** the chain where the Interchain account is registered. The host chain listens for IBC packets from a controller chain which should contain instructions (e.g. cosmos SDK messages) which the Interchain account will execute.

**Controller Chain:** the chain that registers and controls an account on a host chain. The controller chain sends IBC packets to the host chain to control the account.



The Interchain accounts application module is structured to **support the ability to exclusively enable controller or host functionality**. This can be achieved by simply omitting either the controller or host Keeper from the Interchain account's NewAppModule constructor function, and mounting only the desired submodule via the IBCRouter. Alternatively, [submodules can be enabled and disabled dynamically using on-chain parameters (opens new window)↗](https://ibc.cosmos.network/main/apps/interchain-accounts/parameters.html).

**Interchain account (ICA):** an account on a host chain. An Interchain account has all the capabilities of a normal account. However, rather than signing transactions with a private key, a controller chain's authentication module will send IBC packets to the host chain which contain the transactions that the Interchain account should execute.

**Interchain account owner:** an account on the controller chain. Every Interchain account on a host chain has a respective owner account on a controller chain. This owner account could be a module account (in Cosmos SDK chains) or an analogous account, it is not strictly limited to regular user accounts.

Now it's time to look at the API on both the controller and host sides.

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#controller-api) Controller API

The controller chain is the chain on which the controller account lives. This controller account is then able to open an ICA channel to a host chain, and create an Interchain account on the other side of the channel which lives on the host chain. The owner of the controller account can then send instructions (via transactions) with the ICA module to the account that it controls on the host chain. How to authenticate owners will be handled in a later section.

The provided API on the controller submodule consists of:

* RegisterInterchainAccount: this enables the registration of Interchain accounts on the host side, associated with an owner on the controller side.
* SendTx: once an ICA has been established, this allows you to send transaction bytes over an IBC channel to have the ICA execute it on the host side.

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#register-an-interchain-account) Register an Interchain account

RegisterInterchainAccount is a self-explanatory entry point to the process. Specifically, it generates a new controller portID using the owner account address; it binds an IBC port to the controller portID, and initiates a channel handshake to open a channel on a connection between the controller and host chains. An error is returned if the controller portID is already in use.

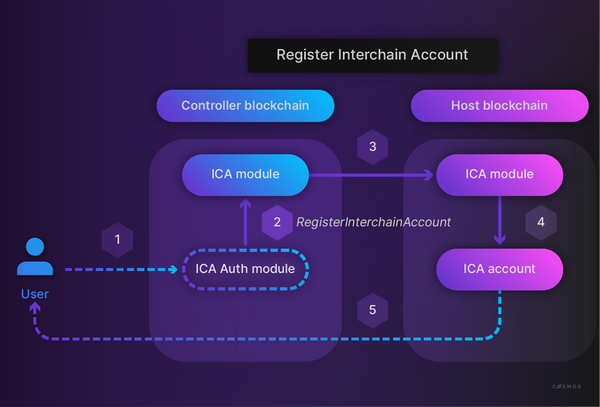
A ChannelOpenInit event is emitted that can be picked up by an off-chain process such as a relayer. The account will be registered during the OnChanOpenTry step on the host chain. This function must be called after an OPEN connection is already established with the given connection identifier.



Copy

// pseudo code

function RegisterInterchainAccount(connectionId: Identifier, owner: string, version: string) returns (error)





It is best practice that the portId for an ICA channel is icahost on the host side, while on the controller side it will be dependent on the owner account, for example icacontroller-<owner-account>.

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#sending-a-transaction) Sending a transaction

SendTx allows the owner of an Interchain account to send an IBC packet containing instructions (messages) to an Interchain account on a host chain.



Copy

// pseudo code

function SendTx(

capability: CapabilityKey,

connectionId: Identifier,

portId: Identifier,

icaPacketData: InterchainAccountPacketData,

timeoutTimestamp uint64): uint64 {

// check if there is a currently active channel for

// this portId and connectionId, which also implies an

// Interchain account has been registered using

// this portId and connectionId

activeChannelID, found = GetActiveChannelID(portId, connectionId)

abortTransactionUnless(found)

// validate timeoutTimestamp

abortTransactionUnless(timeoutTimestamp <= currentTimestamp())

// validate icaPacketData

abortTransactionUnless(icaPacketData.type == EXECUTE\_TX)

abortTransactionUnless(icaPacketData.data != nil)

// send icaPacketData to the host chain on the active channel

sequence = handler.sendPacket(

capability,

portId, // source port ID

activeChannelID, // source channel ID

0,

timeoutTimestamp,

icaPacketData

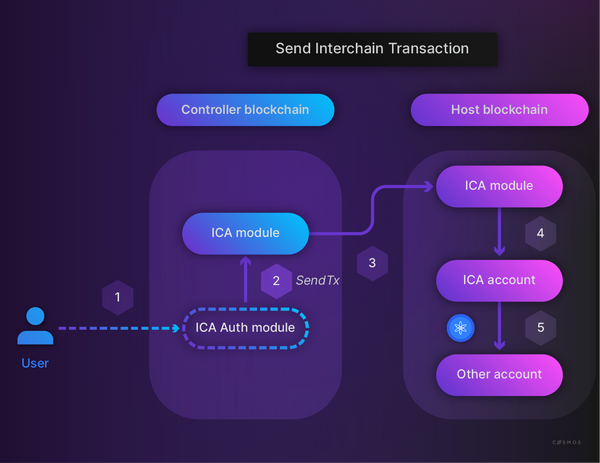
)

return sequence

}



The packet data that is sent over IBC, icaPacketData, should be of type EXECUTE\_TX and have a **non nil** data field.   
  
Additionally, note that SendTx calls core IBC's sendPacket API to transport the packet over the ICS-27 channel.



[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#ics-27-channels) ICS-27 channels

After an Interchain account has been registered on the host side, the main functionality is provided by SendTx. When designing ICA for the ibc-go implementation, a decision was made to use [ORDERED channels](https://ida.interchain.io/academy/3-ibc/3-channels.html), to ensure that messages are executed in the desired order on the host side, akin to the use of the transaction sequence for regular accounts.

A limitation when using ORDERED channels is that when a packet times out the channel will be closed. In the case of a channel closing, it is desirable that a controller chain is able to regain access to the Interchain account registered on this channel. The concept of *active channels* enables this functionality.

When an Interchain account is registered using the RegisterInterchainAccount flow, a new channel is created on a particular port. During the OnChanOpenAck and OnChanOpenConfirm steps (on the controller and host chains respectively) the active channel for this Interchain account is stored in state.

It is possible to create a new channel using the same controller chain portID if the previously set active channel is now in a CLOSED state.



For example, **in ibc-go** one can create a new channel using the Interchain account programmatically by sending a new MsgChannelOpenInit message, like:



Copy

msg := channeltypes.NewMsgChannelOpenInit(

portID,

string(versionBytes),

channeltypes.ORDERED,

[]string{connectionID},

icatypes.HostPortID,

authtypes.NewModuleAddress(icatypes.ModuleName).String())

handler := keeper.msgRouter.Handler(msg)

res, err := handler(ctx, msg)

if err != nil {

return err

}

Alternatively, any relayer operator may initiate a new channel handshake for this Interchain account once the previously set Active Channel is in a CLOSED state. This is done by initiating the channel handshake on the controller chain **using the same portID** associated with the Interchain account in question.

It is important to note that once a channel has been opened for a given Interchain account, new channels cannot be opened for this account until the current Active Channel is set to CLOSED.

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#host-api) Host API

The host chain is the chain where the Interchain account is created and the transaction (sent by the controller) is executed.

Therefore, the provided API on the host submodule consists of:

* RegisterInterchainAccount: enables the registration of Interchain accounts on the host, associated with an owner on the controller side.
* ExecuteTx: enables the transaction data to be executed, provided successful authentication.
* AuthenticateTx: checks that the signer of a particular message is the Interchain account associated with the counterparty portID of the channel that the IBC packet was sent on.



The host API methods run automatically as part of the flow and need not be exposed to an end-user or module, as is the case on the controller side with RegisterInterchainAccount and SendTx.

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#register-an-interchain-account-2) Register an Interchain account

The RegisterInterchainAccount flow was discussed on the controller side already, where it triggered a handshake. On the host side is a complementary part of the flow, but here it's triggered in the OnChanOpenTry step of the handshake, which will create the Interchain account.



Although from the spec point of view we can call this the RegisterInterchainAccount flow, the actual function being called on the host side in ibc-go is called [createInterchainAccount (opens new window)↗](https://github.com/cosmos/ibc-go/blob/v7.0.0/modules/apps/27-interchain-accounts/host/keeper/account.go#L14).

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#executing-transaction-data) Executing transaction data

The host chain state machine will be able to execute the transaction data by extracting it from the InterchainPacketData:



Copy

message InterchainAccountPacketData {

enum type

bytes data = 1;

string memo = 2;

}

The type should be EXECUTE\_TX and data contains an array of messages the host chain can execute.



Executing the transaction data will depend on the execution environment (which blockchain you are on). An example for the ibc-go implementation can be found [here (opens new window)↗](https://github.com/cosmos/ibc-go/blob/v7.0.0/modules/apps/27-interchain-accounts/host/keeper/relay.go).

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#authenticating-the-transaction) Authenticating the transaction

AuthenticateTx is called before ExecuteTx. It checks that the signer of a particular message is the Interchain account owner associated with the counterparty portID of the channel that the IBC packet was sent on.



Remember that the port ID on the controller side was recommended to be of the format icacontroller-<owner-account>. Therefore, the owner account to be authenticated can be found from the counterparty port ID.

Up until this point you may have wondered how authentication is handled on the controller side. This will be the topic of the next section.



The above information is applicable to all implementations of ICS-27, unless explicitly stated otherwise.   
  
By contrast, be aware that the following information deals with the ibc-go implementation specifically.

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#authentication) Authentication

The ICA controller submodule provides an API for registering an account and for sending Interchain transactions. It has been purposefully made lean and limited to generic controller functionality. For authentication of the owner accounts, the developer is expected to provide an authentication module with the ability to interact with the ICA controller submodule.

Here are some relevant definitions:

**Authentication Module:**

* Generic authentication module: Cosmos SDK modules (x/auth, x/gov, or x/group) that offer authentication functionality and can send messages to the ICS-27 module through a MsgServer.
* Custom authentication module: a custom SDK module (satisfying only the AppModule but not IBCModule interface) that offers custom authentication and can send messages to the ICS-27 module through a MsgServer.
* Legacy authentication module: an IBC application module on the controller chain that acts as underlying application for the ICS-27 controller submodule middleware. It forms an IBC middleware stack with the ICS-27 controller module, facilitating communication across the stack.

An **authentication module** must:

* Authenticate Interchain account owners.
* Track the associated Interchain account address for an owner.
* Send packets on behalf of an owner (after authentication).



Originally, when ICA was first introduced in ibc-go v3, developers had to develop a custom authentication module as an IBC application that was wrapped by the ICS-27 module acting as middleware. In ibc-go v6, a refactor of ICA took place that enabled Cosmos SDK modules (x/auth, x/gov, or x/group) to act as generic authentication modules that required no extra development.   
  
A MsgServer was added to the ICA controller submodule to facilitate this.   
  
More information regarding the details and context for the redesign can be found in [ADR-009 (opens new window)↗](https://github.com/cosmos/ibc-go/blob/main/docs/architecture/adr-009-v6-ics27-msgserver.md) or in a [dedicated blog post (opens new window)↗](https://medium.com/the-interchain-foundation/ibc-go-v6-changes-to-interchain-accounts-and-how-it-impacts-your-chain-806c185300d7) on the topic.   
  
For now, the legacy API remains available for those developers who have already built custom IBC authentication modules, but it will be deprecated in the future.



**SDK Security Model**   
  
SDK modules on a chain are assumed to be trustworthy. For example, there are no checks to prevent an untrustworthy module from accessing the bank keeper.   
  
The implementation of [ICS-27 (opens new window)↗](https://github.com/cosmos/ibc/blob/master/spec/app/ics-027-interchain-accounts/README.md) on [ibc-go (opens new window)↗](https://github.com/cosmos/ibc-go/tree/main/modules/apps/27-interchain-accounts) uses this assumption in its security considerations. It assumes that:

* The authentication module will not try to open channels on owner addresses it does not control.
* Other IBC application modules will not bind to ports within the [ICS-27 (opens new window)↗](https://github.com/cosmos/ibc/blob/master/spec/app/ics-027-interchain-accounts/README.md) namespace.

More information on which type of authentication module to use for which development use case can be found [here (opens new window)↗](https://ibc.cosmos.network/main/apps/interchain-accounts/development.html).

There you will find references to development use cases requiring access to the packet callbacks, which are discussed in the next section.

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#application-callbacks) Application callbacks

Custom authentication is one potential use case for the use of Interchain accounts; however, another important use case quickly became apparent: Interchain accounts packets being sent as part of a composable programmatic flow.

As an example, consider a remote-controlled atomic swap:

1. Send an ICS-20 packet to a remote chain.
2. If it is successful, then send an ICA-packet to swap tokens on a liquidity pool (LP) on the host chain.
3. Return the funds back to the sender (on the controller chain).



The request from the community to enable a standard for this type of flow resulted in [ADR-008 (opens new window)↗](https://github.com/cosmos/ibc-go/blob/main/docs/architecture/adr-008-app-caller-cbs/adr-008-app-caller-cbs.md), which extends the ability for general use cases.   
  
It is advisable to follow developments around ADR-008 and the so-called *callback interface for IBC actors*, i.e. secondary applications (like smart contracts for example) that want to call into IBC apps as part of their state machine logic.

[#Copy link](https://ida.interchain.io/academy/3-ibc/8-ica.html#practical-exercise) Practical exercise

Ready to get your hands dirty now that you understand how ICA works? Try out [this tutorial (opens new window)↗](https://github.com/cosmos/ibc-go/wiki/How-to-use-groups-with-ICA) on how to use groups with Interchain accounts.

previous

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

**[IBC Token Transfer](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)**

up next

**[IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/9-ibc-mw-intro.html)**

[[](https://ida.interchain.io/academy/3-ibc/9-ibc-mw-intro.html)](https://ida.interchain.io/academy/3-ibc/9-ibc-mw-intro.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

[](https://ida.interchain.io/ida-course/discord-info.html)

On this page

[What are Interchain Accounts?](https://ida.interchain.io/academy/3-ibc/8-ica.html#what-are-interchain-accounts)

[ICA core functionality: controller & host](https://ida.interchain.io/academy/3-ibc/8-ica.html#ica-core-functionality-controller-host)

[Controller API](https://ida.interchain.io/academy/3-ibc/8-ica.html#controller-api)

[Host API](https://ida.interchain.io/academy/3-ibc/8-ica.html#host-api)

[Authentication](https://ida.interchain.io/academy/3-ibc/8-ica.html#authentication)

[Application callbacks](https://ida.interchain.io/academy/3-ibc/8-ica.html#application-callbacks)

[Practical exercise](https://ida.interchain.io/academy/3-ibc/8-ica.html#practical-exercise)

#### **Get Cosmos updates**

Unsubscribe at any time. [Privacy Policy↗](https://v1.cosmos.network/privacy)

     Next

Documentation

[Cosmos SDK](https://docs.cosmos.network/)[Cosmos Hub](https://hub.cosmos.network/)[CometBFT](https://docs.cometbft.com/)[IBC Protocol](https://ibc.cosmos.network/)

Community

[Interchain blog](https://blog.cosmos.network/)[Forum](https://forum.cosmos.network/)[Discord](https://discord.gg/cosmosnetwork)

Contributing

[Source code on GitHub](https://github.com/cosmos/sdk-tutorials)

[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)

**[](https://blog.cosmos.network/)[](https://twitter.com/cosmos)[](https://discord.gg/cosmosnetwork)[](https://www.linkedin.com/company/interchain-foundation/about/)[](https://reddit.com/r/cosmosnetwork)[](https://t.me/cosmosproject)[](https://www.youtube.com/c/CosmosProject)**



Dark mode

† This website is maintained by the Interchain Foundation (ICF). The contents and opinions of this website are those of the ICF. The ICF provides links to cryptocurrency exchanges as a service to the public. The ICF does not warrant that the information provided by these websites is correct, complete, and up-to-date. The ICF is not responsible for their content and expressly rejects any liability for damages of any kind resulting from the use, reference to, or reliance on any information contained within these websites.

Cosmos is a registered trademark of the [Interchain Foundation.](https://interchain.io/)[Privacy](https://v1.cosmos.network/privacy)