FCE CRYPTO BRIDGE®

SERVICE TERMS AND CONDITIONS

Overview

Document information

Title

FCE CRYPTO BRIDGE SERVICE TERMS AND CONDITIONS

Version: 1.1

Welcome to FCE Crypto Bridge Service (“FCE Bridge”). These terms and conditions ("Terms") form a legal contract between you and FCE Group AG (“we” or “us” or “FCE”) and govern your use of FCE Bridge. By using FCE Bridge, you represent and warrant that you have read and understood, and agree to be bound by these Terms, the Transparenterra Terms of Use, and the Transparenterra Privacy Policy, which are incorporated herein by reference. If you do not agree with these Terms in their entirety, you may not use (and should immediately stop using) FCE Bridge.

Authors and Project Management

The FCE Group AG team

Place and Date

Lucerne (Switzerland). June 2023

Content

What is FCE Bridge?

FCE Bridge allows you to go beyond the FCE Ecosystem. It acts as a gateway to bring the FCEM, the native currency of the FCE Blockchain network, to the Ethereum mainnet network, subject to certain conditions and fees described below.

This service allows exchanging FCEM coins for tokens in the Ethereum network on a 1:1 basis. The token in the Ethereum network which appears as the result of the exchange is called the FCEMETH, which is an FCE-wrapped token. The wrapped FCEM (FCEMETH) is a smart contract in the Ethereum network that mints new tokens when the FCE user wants to freeze their assets in the bridge smart contract in the FCE private network. And in the opposite case, the smart contract in the public Ethereum network burns tokens to release FCEM coins from the contract in the FCE private network.

FCE Bridge allows users to use the assets of a private FCE network in a public Ethereum network, which expands the users' opportunities both when interacting with smart contracts and accessing new liquidity. FCE Bridge unlocks access to the mature, secure, developed Ethereum network for FCEM holders.

FCEMBridge smart contract

Special contract data types:

enum **Status** - options that hashes can take which stored on the contract.

Name	Type	Description
Nonexist	0	Default value: The signature either does not exist or has not yet been used.
Undone	1	Signature registered but not applied due to certain conditions. Can be used again.
Done	2	Signature used and tokens are minted, cannot be reused.

Table 1

Private properties:

Name	Type	Description
Nonce	Uint	Token transfer operations counter

Table 2

Public properties:

Name	Type	Description
DEFAULT_ADMIN_ROLE	bytes32 constant	Hash of the name of the deployer and chief admin role, could be only one.
ADMIN_ROLE	bytes32 constant	Hash of the name of the role whose members have the right to manage special functionality.
isFCEBridgeAvailable	bool	Availability to functions on this contract at current network.
isContractAvailable	bool	Availability of token swap from contracts balances.
bridgeValidators	mapping(uint => address)	Mapping of certain network ids to addresses, that validates hash only for specific networks at current.

isBridgeValid	mapping(
	uint =>
	uint => address =>
	address =>bool)
		Mapping with 4 levels of nesting indicates availability of swap and redeem, includes 2 networks ids and 2 token addresses between that needs to be proceed.

redeemStatus	mapping(bytes32 => Status)	The mapping stores the hashes and their **Status**. The hash is obtained by linking all the arguments with signature passed to the **redeem**() function

Table 3

Methods:

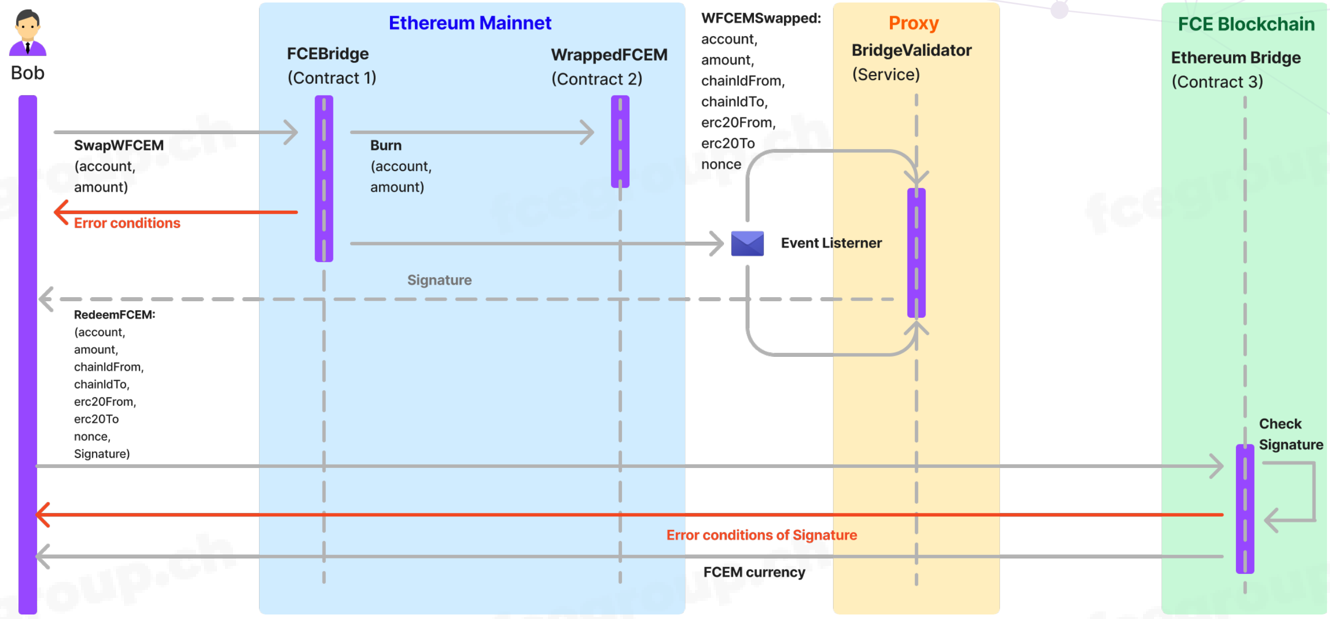
Name	Description
constructor (address **bridgeValidator** uint **chainIdA**, uint **chainIdB**, address **tokenA**, address **tokenB**)	Contract creation constructor.
isBridgeValid (uint **chainId,** address **erc20**) **view** returns(bool)	Returns true if the ERC20 bridge exists on the chainId network and it is valid on the given contract in the **Bridges** array. Anyone allows to call.
checkSign (address **sender,** address **recipient,** uint256 **amount,** uint256 **chainIdfrom,** uint256 **chainIdto,** address **tokenFrom,** address **tokenTo,** uint256 **nonce_,** bytes **signature)**) **view** returns(**bool**)	Checks if the **bridgeValidator** has actually signed the given signature, returns true or false. **sender** - sender's address **recipient** - address of the recipient. **amount** - amount of sent currency. **chainIdfrom** - identifier of the network from which the swap was sent. **chainIdto** - identifier of the network to which the **nonce_** - the value of the swap counter at the time of the transfer **signature** - hash signed by **bridgeValidator** Anyone allows to call.
hashMessage (bytes32 **message**) **pure** returns(bytes32)	Appends the required network prefix to the given hash for validation Anyone allows to call.
swap ((address **tokenTo**, uint256 **chainIdto**, uint256 **amount**, address **tokenFrom**)returns(bytes32 **hashToSign**)	Burns the function initiator's existing FCEM, FCEMETH tokens or other and returns a hash of the passed arguments to swap. All swap arguments are passed in the SwapInitialized event. **amount** - amount of sent currency. **chainIdto** - identifier of the network to which the transfer was sent. **tokenFrom** - address of token contract from which the swap was sent, if it is equal to EthereumBridge, then the FCEM swap was made from the FCE network. **tokenTo** - the address of token contract which needs to be minted from swap, if it is equal to EthereumBridge, then the swap was from WrappedFCEM. **hashToSign** - encrypted translation arguments that need to be signed by the bridgeValidator. Anyone allows to call.

Table 4

redeem (address **recipient**, uint256 **amount**, uint256 **chainIdfrom**, address **tokenFrom**, address **tokenTo**, uint256 **nonce_**, bytes **signature**)	When passing valid parameters and signature, it transfers FCEM or mints FCEMETH or other tokens to the address of the recipient with certain amount. Based on the passed arguments, a hash is formed, then using the ecrecover function and the hash from signature, it checks if the bridgeValidator signed that hash. Signature can be used only once and only with certain parameters. All swap arguments are passed in the RedeemInitialized event. **recipient** - recipient/sender address **amount** - amount of sent currency **chainIdfrom** - identifier of the network from which the swap was sent. **chainIdto** - identifier of the network to which the swap was sent. **tokenFrom**- address of token contract from which the swap was sent, if it is equal to **EthereumBridge**, then the FCEM swap was made from the FCE network. **tokenTo** - address of the token contract that needs to be minted on other network, if it is equal to **EthereumBridge**, then the transfer will be made in FCEM at the FCE network. **nonce_** - the value of the swap counter at the time of the transfer. **signature** - encrypted translation arguments that need to be signed by the **bridgeValidator**. Anyone allows to call.
setBridgeAccess (uint **chainIdFrom**, uint **chainIdTo**, address **tokenFrom**, address **tokenTo**, bool **valid**)	Sets swap or redeem availability between **chainIdFrom** and **chainIdTo** networks and **tokenFrom**, **tokenTo** token addresses to **valid** value. Can only be called by members of ADMIN_ROLE.
setDoubleBridgeAccess (uint **chainIdFrom**, uint **chainIdTo**, address **tokenFrom**, address **tokenTo**, bool **valid**)	Sets swap and redeem availability between **chainIdFrom** and **chainIdTo** networks and **tokenFrom**, **tokenTo** token addresses to **valid** value for both sides. Can only be called by members of ADMIN_ROLE.
setBridgeValidator (uint **chainId**, address **newValidator**)	Sets specific **newValidator** address as bridge validator for certain network **chainId** Can only be called by DEFAULT_ADMIN_ROLE.
setContractAvalibality (bool **value**)	Sets **isContractAvailable** variable to **value**, changing contract call availability to swap and redeem functions. Can only be called by members of ADMIN_ROLE.
split(bytes **signature**) **pure** returns (uint8 **v**, bytes32 **r**, bytes32 **s**)	Divides signature (signed hash) into 3 signature parameters **v,r,s** on ECDSA elliptic curves Anyone allows to call.
getChainID() **view** returns (uint)	Returns the identifier of the network in which the current contract is deployed. Anyone allows to call

Table 4

How does FCE Bridge work?



To transfer an asset, a user must select the network from which the user wants to withdraw funds, the denomination, and the volume. After the transfer operation has been confirmed with the user's private key, the asset is transferred to a smart contract in the FCE network, on which it is blocked. The following action is for the user to call the bridge contract on the public network and receive the wrapped coins.

> ***The following is a detailed description of the transfer of FCE native coin from the FCE permissioned network:***

- The user accesses the EthereumBridge contract and calls the SwapFCEM function with a certain number of FCEMs that the user wants to transfer. The function call initiates an FCEMSwaped event with transfer parameters.
- The Bridge Validator collects the data parameters of the FCEMSwaped event and generates a Signature hash using the Validator's private key.
- Transition and Signature parameters are forwarded to the user and stored as a hash.
- The user switches to the Ethereum network and accesses the FCEMBridge contract with the same transfer and Signature parameters.
- The FCEMBridge contract confirms that the Bridge Validator has signed the Signature and creates FCEMETH - WrapedFCEM (ERC20) tokens to the user's address.

FCEMETH smart contract

Special contract data types:

enum **Permition** - options for values that address can take on the contract.

Name	Type	Description
global	0	Token transfers allowance determined by isTransactionsOn value
forbidden	1	Token transfers forbidden for this address, despite isTransactionsOn value
allowed	2	Token transfers allowed for this address, despite isTransactionsOn value

Table 5

Public properties:

Name	Type	Description
ADMIN_ROLE	bytes32 constant	The hash of the name of the role whose members have the right to manage contracts
isTransactionsOn	bool	Global token transfers allowance to all addresses
ChiefAdmin	address	**ChiefAdmin** address, assigned when the contract is deployed
permissionToReceive	mapping(address => **Permition**)	Special permissions to allow/prohibit transactions to get tokens for specific accounts
permissionToTransfer	mapping(address => **Permition**)	Special permissions to allow/prohibit transactions to move tokens for specific accounts
name	string	Name of token
symbol	string	Symbol of token

Table 6

Public methods:

Name	Description
mint(address **to**, uint256 **amount**)	Creates new emistion of tokens **amount** to certain address **to**. Only accounts with ADMIN_ROLE are allowed to call this function
burn(address **from**, uint256 **amount**)	Decreases **amount** of tokens on certain address **from** Only accounts with ADMIN_ROLE are allowed to call this function
grantRole(bytes32 **role**, address **account**)	Grants specific **role** to the certain **account** address. Only **ChiefAdmin** is allowed to call
revokeRole(bytes32 **role**, address **account**)	Revokes specific **role** from certain **account** addresses. Only **ChiefAdmin** is allowed to call
updateGlobalTransactions (bool **value)**	Sets **isTransactionsOn** to a new **value** Only **ADMIN_ROLE** members is allowed to call
updatePermissionToReceive (address **account**, Permission **_permissionValue**)	Updates permission to receive tokens to **account** address to **_permissionValue** at **permissionToReceive** mapping. Only **ADMIN_ROLE** members is allowed to call
updatePermissionToTransfer (address **account**, Permission **_permissionValue**)	Updates permission to transfer tokens to the **account** address to **_permissionValue** at **permissionToTransfer** mapping. Only **ADMIN_ROLE** members is allowed to call
isTransactionAllowed (address **accountFrom**, address **accountTo**) **view** returns(**bool**)	Checks if the transaction of tokens allowed between **accountFrom** and **accountTo** addresses.
transfer (address **to**, uint **amount**)) returns(bool)	Moves **amount** of tokens to address **to** from contract caller. Returns a boolean value indicating whether the operation succeeded.
allowance (address **owner**, address **spender**,) returns(uint)	Returns the remaining number of tokens that the **spender** will be allowed to spend on behalf of the **owner** through transferFrom function. Returns zero by default. This value changes when approving or transferFrom is called.
approve (address **spender**, uint256 **amount**,) returns(bool)	Sets **amount** as the allowance of **spender** over the caller's tokens. Returns a boolean value indicating whether the operation succeeded.
transferFrom (address **from**, address **to**, uint **amount**) returns (bool)	Moves **amount** tokens from **from** to **to** using the allowance mechanism. **amount** is then deducted from the caller's allowance. Returns a boolean value indicating whether the operation succeeded.

Table 7

balanceOf (address **account**) **view** returns (uint)	Returns the number of tokens owned by the **account**.
totalSupply() **view** returns **(uint)**	Returns the number of tokens in existence.

Table 7

Who is eligible to use FCE Bridge?

To use FCE Bridge, you must be of legal age to form a binding contract in your jurisdiction. You represent that you have the full right, power, and authority to enter into and comply with the Terms on behalf of yourself and any company or legal entity for which you may use FCE Bridge. You further represent that you are not a citizen, resident, or member of any jurisdiction where your use of FCE Bridge would be illegal or otherwise violate any applicable law.

To use FCE Bridge, you must be registered on the TransparenTerra website (https://transparenterra.com) and complete the KYC checks (for more information Video 2: KYC.).

Like the rest of the FCE permissioned network functionality, the functionality of FCE Bridge will be available only through the interface of the TransparenTerra website (https://transparenterra.com).

We reserve the right to refuse service to anyone who breaches these Terms, the Transparenterra Terms of Use, the Transparenterra Privacy Policy, or the Transparenterra Honor Code & Community Guideline.

Representations by you

You represent and warrant that all information you provide concerning FCE Bridge is accurate, complete, and not misleading. You understand and acknowledge that any inaccurate, incomplete, or misleading information provided by you for the swap transaction may result in the cancellation of the transaction and forfeiture of any fees paid.

Fees and taxes

FCE Bridge is free of charge, but a user shall pay a fixed FCE blockchain environmental fee (called "CaFFee") to compensate for the carbon footprint of the transaction. This fee is payable in FCEM, the exact amount appears on the transaction confirmation page. For more information and tutorial materials about the FCE blockchain environmental fee - CaFFee, and other functionality of the TrasparenTerra platform, please refer to Video 9. CaFFee brief introduction.).

Your use of FCE Bridge may result in tax obligations depending on the jurisdiction you are located in. You acknowledge that you are solely responsible for determining any tax obligations that may apply to your activities on FCE Bridge.

No professional advice. No fiduciary duties

You agree that FCE Bridge and/or any information provided by or obtained from these Terms is for informational purposes only, is not intended to be relied upon for professional advice, and is not a substitute for information from experts or professionals in the applicable area. You should not take, or refrain from taking, any action or decision based on any information contained in FCE Bridge or these Terms. Before making any financial, legal, or other decisions involving FCE Bridge, you should seek independent professional advice from an individual licensed and qualified in the area for which such advice would be appropriate.

You acknowledge and agree that (i) these Terms are not intended to, and do not, create or impose any fiduciary duties on us, (ii) we owe no fiduciary duties or liabilities to you or any other party.

Disclaimers

FCE Bridge is provided on an "AS IS" basis. To the fullest extent permitted by law, we disclaim any representations and warranties of any kind, whether express, implied, or statutory, including (but not limited to) the warranties of merchantability and fitness for a particular purpose. You acknowledge and agree that using FCE Bridge is at your own risk. We do not represent or warrant that access to FCE Bridge will be continuous, uninterrupted, timely, or secure; or that FCE Bridge will be free from errors, defects, viruses, or other harmful elements. No advice, information, or statement we make should be treated as creating any warranty concerning FCE Bridge

Assumption of risks

By using FCE Bridge, you represent that you understand the inherent risks associated with using cryptographic and blockchain-based systems. You understand that the digital assets markets are highly volatile, and you acknowledge the risk that your digital assets may lose some or all of their value. You further acknowledge that we are not responsible for any of these variables or risks and cannot be held liable for any resulting losses you experience while accessing or using FCE Bridge. Accordingly, you understand and agree to assume full responsibility for all risks of accessing and using FCE Bridge.

Indemnification

You agree to hold harmless, release, defend, and indemnify us and our officers, directors, employees, contractors, agents, affiliates, and subsidiaries from and against all claims, damages, obligations, losses, liabilities, costs, and expenses (including attorney’s fees) arising from (a) your use of FCE Bridge; (b) your violation of these Terms, the right of any third party, or any applicable law; and (c) any other party’s use of FCE Bridge with your assistance or using any device or account that you own or control.

Limitation of Liability

Under no circumstances shall we or any of our officers, directors, employees, contractors, agents, affiliates, or subsidiaries be liable to you for any indirect, punitive, incidental, special, consequential, or exemplary damages, including (but not limited to) damages for loss of profits, goodwill, use, data, or other intangible property, arising out of or relating to any access or use of FCE Bridge, nor will we be responsible for any damage, loss, or injury resulting from hacking, tampering, or other unauthorized access or use of FCE Bridge or the information contained within it; interruption or cessation of function related to FCE Bridge; delays or failures in processing transactions; network congestion or other network issues; market fluctuations or price changes.

Communication and FCE TT.Concierge chatbot

When you communicate with us via electronic communication, you should be aware that electronic communications can fail, be delayed, may not be secure, and/or may not reach the intended destination.

If you use the FCE TT.Concierge chatbot integrated into the TransparenTerra platform ("Chatbot"), the following disclaimers shall also apply to your use of the Chatbot (if you disagree, you must not use the Chatbot):

- Chatbot is only intended to be a facilitative tool to assist you in obtaining answers efficiently, using content from the TransparenTerra platform. It is not intended to provide any form of specific advice to your intended query. Using Chatbot, you agree that any responses or information obtained is at your sole risk and discretion.
- Chatbot is integrated with and powered by ChatGPT. Depending on your questions, the responses provided by Chatbot to you may also contain publicly available third-party information, views, and opinions obtained from the internet. To the extent that such third-party content is incorporated as part of the Chatbot's response to you, please note that all such content is presented to you on an "as is" basis for general information purposes only, without representation or warranty of any kind.
- Chatbot's responses, as generated and returned to you, depend on user input and its underlying algorithms. Accordingly, and as stated above, we are not responsible for the same and do not guarantee the accuracy of the responses returned.

Governing Law and dispute resolution

These Terms shall be governed by and construed in accordance with the law of Switzerland without giving effect to any choice of law or conflict of law provisions. Any dispute, controversy, or claim arising out of, or in relation to, these Terms, including the validity, invalidity, breach, or termination thereof, shall be resolved by arbitration in accordance with the Swiss Rules of International Arbitration of the Swiss Arbitration Centre in force on the date on which the Notice of Arbitration is submitted in accordance with those Rules. The arbitration shall be held in Zurich, Switzerland, and the arbitral proceeding shall be conducted in English.

Modifications

In our sole discretion, we reserve the right to modify these Terms occasionally. If we make any modifications, we will notify you by updating the date at the top of this page and by maintaining a current version of the Terms. All modifications will be effective when posted, and you are responsible for reviewing these Terms periodically for any updates or changes. If you do not agree with any modifications, you must immediately stop using FCE Bridge.

Thank you

FCE Group AG

Techno-Park Luzern CH-6039, Root D4,
Switzerland

https://fcegroup.ch

info@fcegroup.ch