# 1 Goerli Testnet:

Goerli Testnet is an Layer 1 (Ethereum side) of the Testnet. The ERC20 tokens are normally deployed in Layer1.

## 1.1 Network details:

RPC address/ChainId : <https://rpc.goerli.mudit.blog> (5)

Transaction explorer : <https://goerli.etherscan.io/>

Faucet for test ETHER : https://faucet.goerli.mudit.blog/

## 1.2 Token deployment:

ERC20 test token is deployed to the Goerli Testnet

Token Symbol/Name : GNRA / GeneraToken

Address : 0x365f844477D3b2AE1174306DC36AEbA73818b1e7

# 2 Mumbai Testnet:

Mumbai Testnet is a Layer 2 (Polygon side) of the Testnet.

## 2.1 Network details:

RPC address/ChainId : <https://rpc-mumbai.maticvigil.com/> (80001)

Transaction explorer : https://mumbai.polygonscan.com/

Faucet for test MATIC : <https://faucet.polygon.technology/>

## 2.2 Token details:

The ERC20 token can be of two types. Both the token types allows the user to withdraw tokens from polygon to Ethereum

1. Poygon-non mintable : An Ethereum deployed ERC20 token can be bridged to a polygon ERC20 token. The ERC20 token minting will happen in Ethereum and then transferred to polygon.
2. Polygon mintable : ERC20 token that can be directly deployed and minted on Polygon.

## 2.3 Mapping ERC20 token from Ethereum to Polygon:

URL : [https://mapper.matic.today/mapGraphical user interface, application, Teams

Description automatically generated](https://mapper.matic.today/map#)

Bridge Type : PoS

Choose Type of token : ERC20

Choose Network : Goerli Testnet – Mumbai Testnet

Ethereum Token address : Address of the ERC20 token in the Goerli Network <0x365f844477D3b2AE1174306DC36AEbA73818b1e7>

Polygon Token : N/A Optional

Token symbol/name/decimals : Should be auto-populated

Mintable/Non-Mintable : Non-Mintable (Refer section 2.3 for the difference)

Email : Email address for mapping notification

The created Polygon ERC20 token is <0x185fb189548a2ac46bf40ab2bd656d8de91b9755>

## 2.4 Transferring ERC20 token from Ethereum to Polygon:

URL : <https://wallet-dev.polygon.technology/bridge/>

Graphical user interface, application

Description automatically generated

First user has to sign in-to their wallet (Metamask in our example)

Choose Deposit/Withdraw

From : Choose the ERC20 token available in the Metamask under Goerli network (Genera Token GNRA)

Mention the amount

To : Mumbai Chain and click transfer

The user has to approve the following two transactions:

* GNRA contract – approval(owner,spender,value)
  + Owner - sending account on Goerli
  + Spender - ERC20PredicateProxy/ MintableERC20PredicateProxy
  + Value - token value
* RootChainManagerProxy- depositFor(user,rootToken,depositData)
  1. ERC20PredicateProxy/ MintableERC20PredicateProxy method LockedERC20(owner, receiver, rootToken, amount)
     1. Owner - sending account on Goerli
     2. Receiver - receiving account on polygon
     3. Amount - token value
  2. GNRA contract on Goerli is called for approval(owner, spender)
     1. owner - sending account on Goerli
     2. spender - ERC20PredicateProxy/ MintableERC20PredicateProxy
  3. GNRA contract on Goerli is called for transfer(from, to, value)
     1. From - sending account on Goerli
     2. To - ERC20PredicateProxy/ MintableERC20PredicateProxy
     3. value - token value

## Multi-signature wallet details:

The multi signature wallet is deployed to the polygon network on the address 0x4A9Dd25644dFc11D7767c533Cb04319fC9333FF3