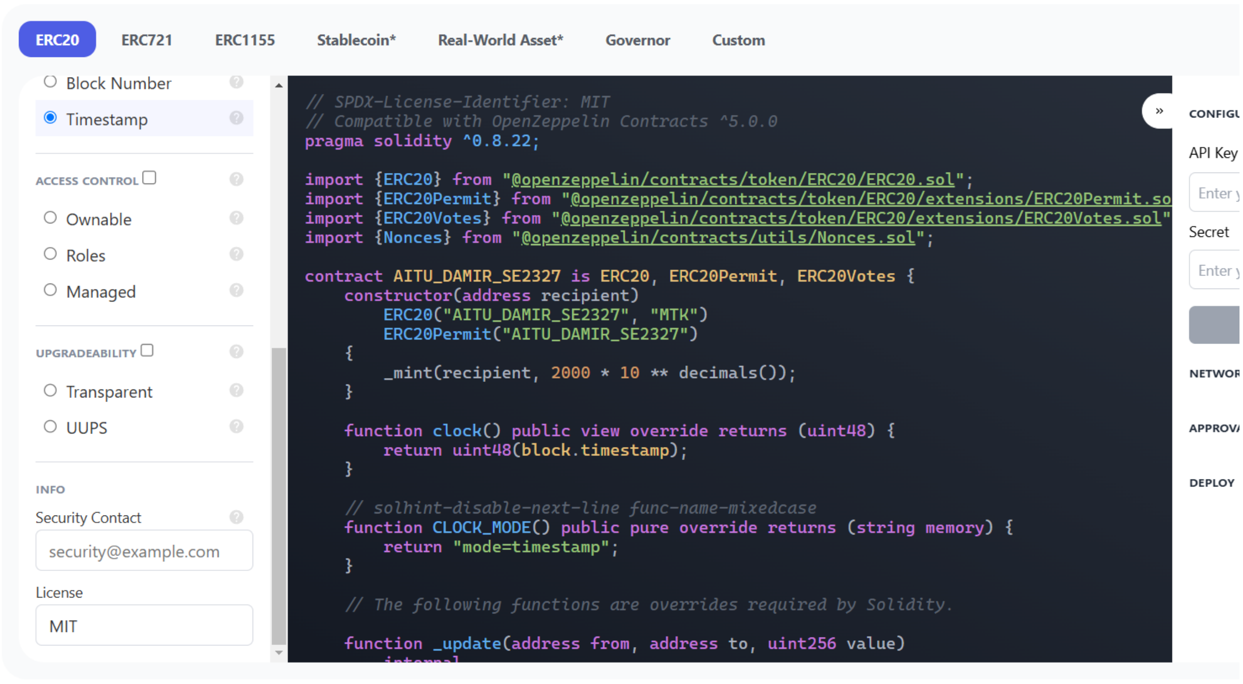


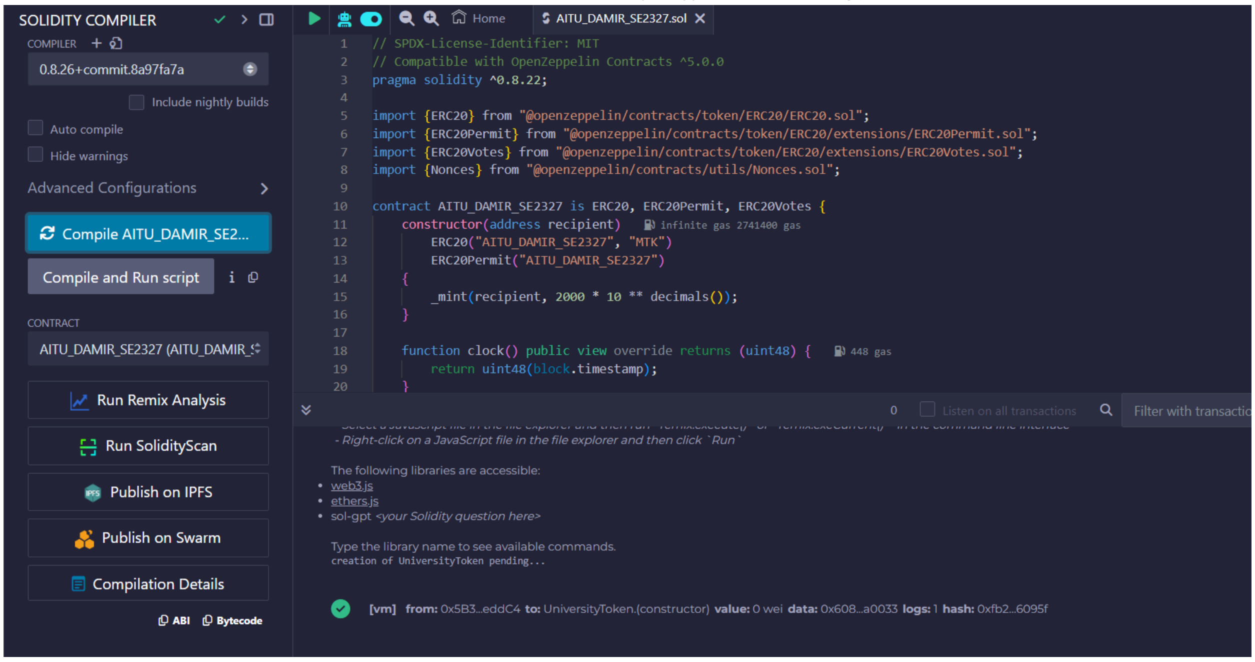
Block Number
[✓ Timestamp]

ACCESS CONTROL
Ownable
Roles
Managed

UPGRADEABILITY
Transparent
UUPS

INFO
Security Contact: security@example.com
License: MIT

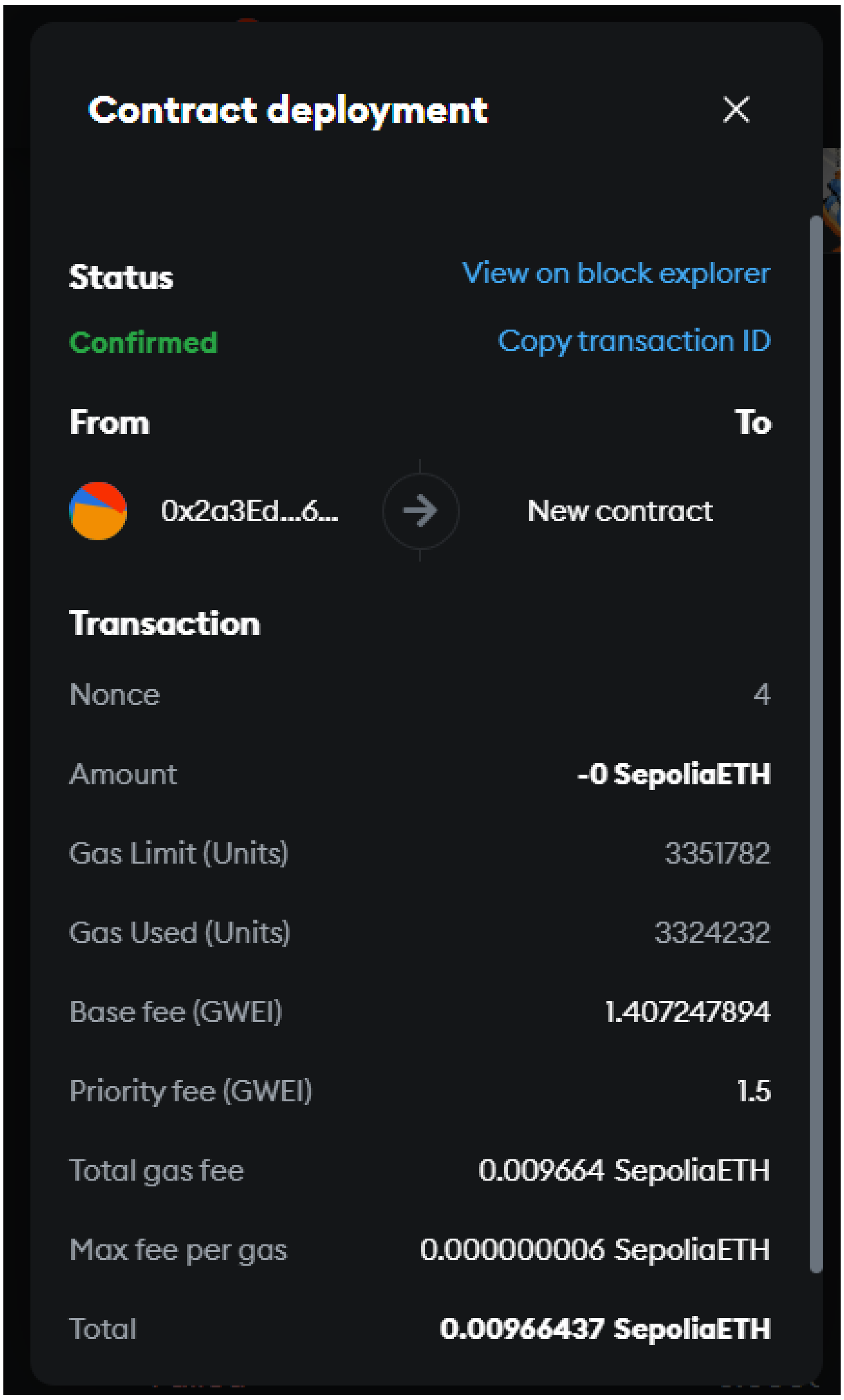
```
// SPDX-License-Identifier: MIT
// Compatible with OpenZeppelin Contracts ^5.0.0
pragma solidity ^0.8.22;

import {ERC20} from "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import {ERC20Permit} from "@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.so
import {ERC20Votes} from "@openzeppelin/contracts/token/ERC20/extensions/ERC20Votes.sol"
import {Nonces} from "@openzeppelin/contracts/utils/Nonces.sol";

contract AITU_DAMIR_SE2327 is ERC20, ERC20Permit, ERC20Votes {
    constructor(address recipient)
        ERC20("AITU_DAMIR_SE2327", "MTK")
        ERC20Permit("AITU_DAMIR_SE2327")
    {
        _mint(recipient, 2000 * 10 ** decimals());
    }

    function clock() public view override returns (uint48) {
        return uint48(block.timestamp);
    }

    // solhint-disable-next-line func-name-mixedcase
    function CLOCK_MODE() public pure override returns (string memory) {
        return "mode=timestamp";
    }

    // The following functions are overrides required by Solidity.

    function _update(address from, address to, uint256 value)
```



Compiler: 0.8.26+commit.8a97fa7a
Contract: AITU_DAMIR_SE2327 (AITU_DAMIR_S

```
// SPDX-License-Identifier: MIT
// Compatible with OpenZeppelin Contracts ^5.0.0
pragma solidity ^0.8.22;

import {ERC20} from "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import {ERC20Permit} from "@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.sol";
import {ERC20Votes} from "@openzeppelin/contracts/token/ERC20/extensions/ERC20Votes.sol";
import {Nonces} from "@openzeppelin/contracts/utils/Nonces.sol";

contract AITU_DAMIR_SE2327 is ERC20, ERC20Permit, ERC20Votes {
    constructor(address recipient)    infinite gas 2741400 gas
        ERC20("AITU_DAMIR_SE2327", "MTK")
        ERC20Permit("AITU_DAMIR_SE2327")
    {
        _mint(recipient, 2000 * 10 ** decimals());
    }

    function clock() public view override returns (uint48) {    448 gas
        return uint48(block.timestamp);
    }
```

```
- Right-click on a JavaScript file in the file explorer and then click `Run`

The following libraries are accessible:
• web3.js
• ethers.js
• sol-gpt <your Solidity question here>

Type the library name to see available commands.
creation of UniversityToken pending...

✓ [vm] from: 0x5B3...eddC4 to: UniversityToken.(constructor) value: 0 wei data: 0x608...a0033 logs: 1 hash: 0xfb2...6095f
```



Contract deployment

Status

Confirmed

View on block explorer

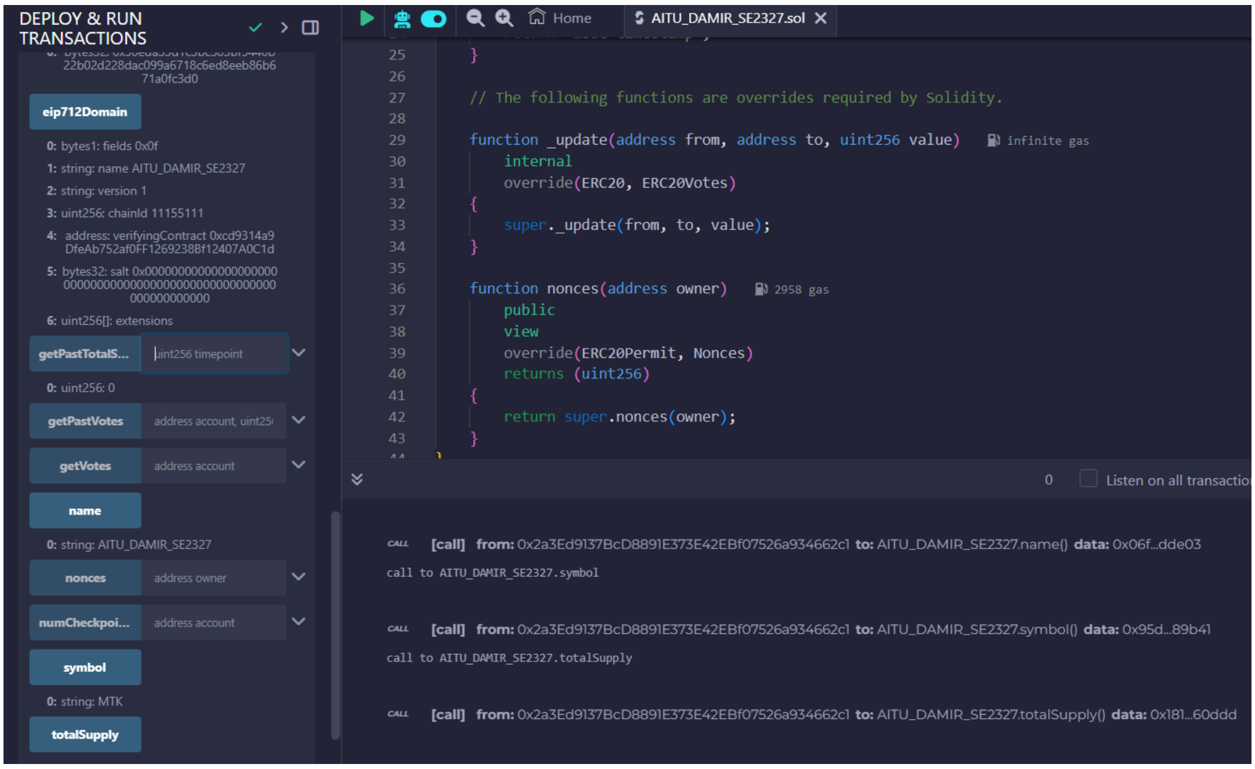
Copy transaction ID

From 0x2a3Ed...6...

To New contract

Transaction

Nonce	4
Amount	**-0 SepoliaETH**
Gas Limit (Units)	3351782
Gas Used (Units)	3324232
Base fee (GWEI)	1.407247894
Priority fee (GWEI)	1.5
Total gas fee	0.009664 SepoliaETH
Max fee per gas	0.000000006 SepoliaETH
Total	**0.00966437 SepoliaETH**



```
22b02d228dac099a6718c6ed8eeb86b6
71a0fc3d0
```

eip712Domain

```
0: bytes1: fields 0x0f
1: string: name AITU_DAMIR_SE2327
2: string: version 1
3: uint256: chainId 11155111
4: address: verifyingContract 0xcd9314a9
   DfeAb752af0FF1269238Bf12407A0C1d
5: bytes32: salt 0x00000000000000000000
   00000000000000000000000000000000
   000000000000
6: uint256[]: extensions
```

getPastTotalS...

```
0: uint256: 0
```

name

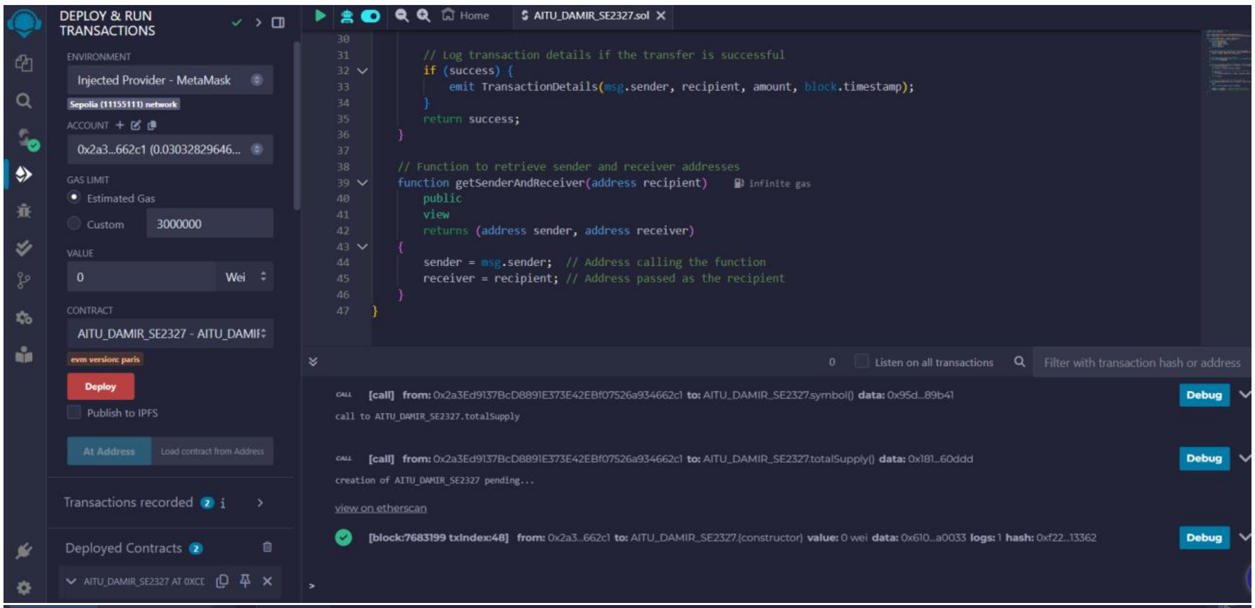
```
0: string: AITU_DAMIR_SE2327
```

symbol

```
0: string: MTK
```

```solidity
    }

    // The following functions are overrides required by Solidity.

    function _update(address from, address to, uint256 value)    infinite gas
        internal
        override(ERC20, ERC20Votes)
    {
        super._update(from, to, value);
    }

    function nonces(address owner)    2958 gas
        public
        view
        override(ERC20Permit, Nonces)
        returns (uint256)
    {
        return super.nonces(owner);
    }
```

```
CALL  [call]  from: 0x2a3Ed9137BcD8891E373E42EBf07526a934662c1 to: AITU_DAMIR_SE2327.name() data: 0x06f...dde03
call to AITU_DAMIR_SE2327.symbol

CALL  [call]  from: 0x2a3Ed9137BcD8891E373E42EBf07526a934662c1 to: AITU_DAMIR_SE2327.symbol() data: 0x95d...89b41
call to AITU_DAMIR_SE2327.totalSupply

CALL  [call]  from: 0x2a3Ed9137BcD8891E373E42EBf07526a934662c1 to: AITU_DAMIR_SE2327.totalSupply() data: 0x181...60ddd
```



DEPLOY & RUN TRANSACTIONS
Home
AITU_DAMIR_SE2327.sol
Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded 2
Deployed Contracts 2
AITU_DAMIR_SE2327 AT 0XCC
Listen on all transactions
Filter with transaction hash or address
Debug

ENVIRONMENT
Injected Provider - MetaMask
Sepolia (11155111) network

ACCOUNT
0x2a3...662c1 (0.03032829646...

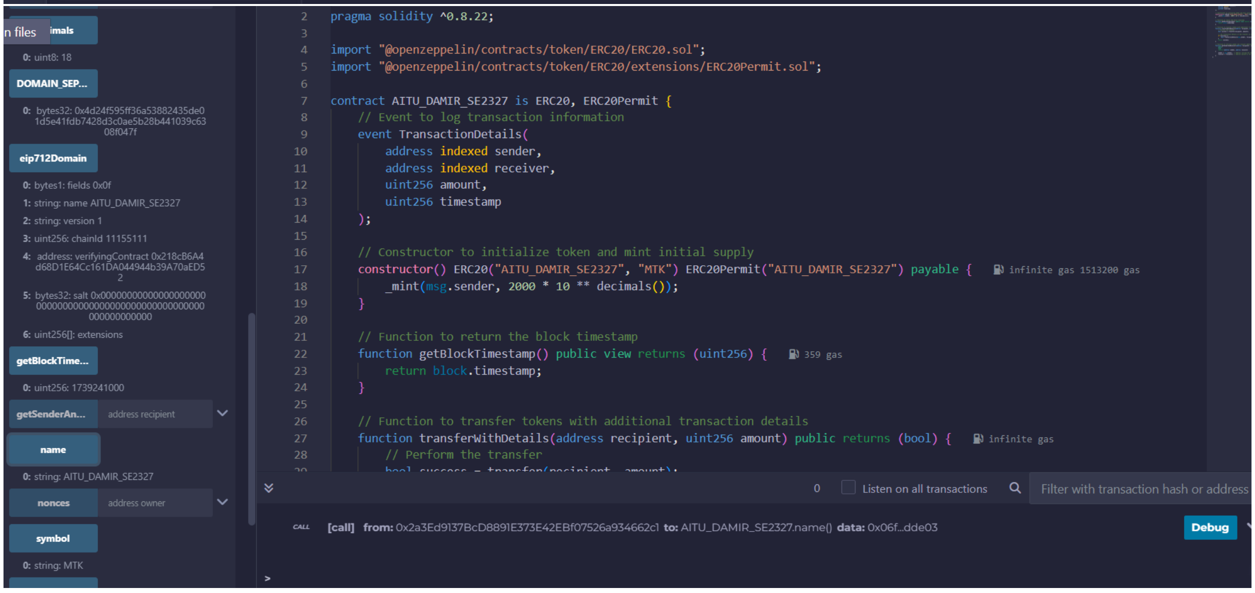
GAS LIMIT
Estimated Gas
Custom 3000000

VALUE
0 Wei

CONTRACT
AITU_DAMIR_SE2327 - AITU_DAMIR
evm version: paris

```
        // Log transaction details if the transfer is successful
        if (success) {
            emit TransactionDetails(msg.sender, recipient, amount, block.timestamp);
        }
        return success;
    }

    // Function to retrieve sender and receiver addresses
    function getSenderAndReceiver(address recipient)    infinite gas
        public
        view
        returns (address sender, address receiver)
    {
        sender = msg.sender;  // Address calling the function
        receiver = recipient; // Address passed as the recipient
    }
}
```

```
CALL [call] from: 0x2a3Ed9137BcD8891E373E42EBf07526a934662c1 to: AITU_DAMIR_SE2327.symbol() data: 0x95d...89b41
call to AITU_DAMIR_SE2327.totalSupply

CALL [call] from: 0x2a3Ed9137BcD8891E373E42EBf07526a934662c1 to: AITU_DAMIR_SE2327.totalSupply() data: 0x181...60ddd
creation of AITU_DAMIR_SE2327 pending...

view on etherscan

[block:7683199 txIndex:48] from: 0x2a3...662c1 to: AITU_DAMIR_SE2327.(constructor) value: 0 wei data: 0x610...a0033 logs: 1 hash: 0xf22...13362
```



```
0: uint8: 18

0: bytes32: 0x4d24f595ff36a53882435de01d5e41fdb7428d3c0ae5b28b441039c6308f047f

0: bytes1: fields 0x0f
1: string: name AITU_DAMIR_SE2327
2: string: version 1
3: uint256: chainId 11155111
4: address: verifyingContract 0x218cB6A4d68D1E64Cc161DA044944b39A70aED52
5: bytes32: salt 0x0000000000000000000000000000000000000000000000000000000000000000
6: uint256[]: extensions

0: uint256: 1739241000

0: string: AITU_DAMIR_SE2327

0: string: MTK
```

```
pragma solidity ^0.8.22;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.sol";

contract AITU_DAMIR_SE2327 is ERC20, ERC20Permit {
    // Event to log transaction information
    event TransactionDetails(
        address indexed sender,
        address indexed receiver,
        uint256 amount,
        uint256 timestamp
    );

    // Constructor to initialize token and mint initial supply
    constructor() ERC20("AITU_DAMIR_SE2327", "MTK") ERC20Permit("AITU_DAMIR_SE2327") payable {    infinite gas 1513200 gas
        _mint(msg.sender, 2000 * 10 ** decimals());
    }

    // Function to return the block timestamp
    function getBlockTimestamp() public view returns (uint256) {    359 gas
        return block.timestamp;
    }

    // Function to transfer tokens with additional transaction details
    function transferWithDetails(address recipient, uint256 amount) public returns (bool) {    infinite gas
        // Perform the transfer
```

```
CALL [call] from: 0x2a3Ed9137BcD8891E373E42EBf07526a934662c1 to: AITU_DAMIR_SE2327.name() data: 0x06f...dde03
```