

## Icareum Audit of smart contracts

### General remarks

The contract code has checked for software bookmarks and critical errors that can lead to loss of money by investors.

- For successful participation in all stages of crowdsale investor must be included in the "investor list";
- The ability to send/transmit tokens is blocked before the end of all stages of ICO.

Contract and crowdsale tokens are using some contracts from the OpenZeppelin library their code also corresponds to the official repository, but there are some minor changes in the code that do not affect the functionality or logic of the work.

The ICRM tokens are sold through the IcareumCrowdsale contract which implements all the logic of the stages of the crowdsale. IcareumCrowdsale inherits the Adminable contract, which allow to add administrators for manage the "investor list" and functional of Ownable contracts which realizing the mechanism of contract ownership, Pausable which allowing to put a sellout on a pause.

Contracts use the MathLib library methods for secure computing.

Emission of tokens is possible only for the period of the crowdsale and only through the IcareumCrowdsale contract.

```
130 function mint(address _to, uint256 _amount) onlyOwner canMint public returns (bool) {  
131     totalSupply = totalSupply.add(_amount);  
132     balances[_to] = balances[_to].add(_amount);  
133     emit Mint(_to, _amount);  
134  
135     return true;  
136 }
```

For each stage of the crowdsale, bonus programs and reservation the following limits for the token emission:

4000000 tokens - reserve for internal needs of the platform

```
138     _mintTokens(_reserveWallet, 4000000);
```

6000000 tokens – Final Stage of presale

```
20     uint256 public constant presaleTokenCap = 6000000;
```

56000000 tokens - stages of the main sale

```
25     uint256 public constant mainsaleTokenCap3 = 56000000;
```

24000000 tokens - bonuses and referral programs

```
28     uint256 public constant mainsaleBonusTokenCap = 24000000;
```

10000000 tokens – bounty and marketing

```
31      uint256 public constant marketingBonusTokenCap = 10000000;
```

100000000 tokens – Total emission limit

### **Code review**

[https://github.com/krogla/icrm\\_audit/tree/master/audit/code\\_review](https://github.com/krogla/icrm_audit/tree/master/audit/code_review)

### **Disclaimer of Liability**

This audit concerns only the source codes of smart contracts and should not be regarded as the approval of a platform, team or company.

### **Authors**

The audit was conducted by Mikhail Semenkin (EthereumWorks team). For issues related to conducting audits and developing smart contracts please contact: - [@SlavaPoe](#) (Telegram), - [v.poskonin, MousePo](#) (Skype).