VI APPENDIX

PROGRAMMING SYNTAX

Javascript

Overall, the programming language of choice for RMC is Javascript. The reason for this is not only that JS is one of the core WWW technologies but also because it is ‚characterized as dynamic, weakly-typed, prototype-based and multi-paradigm'. RMC heavily makes use of JS's multi-paradigm abilities while incorporating event-driven, functional, and object-oriented programming according to the different components and corresponding functionalities of the two dApps. As a scripting language standard, for the client-side and the server-side, RMC abides to ECMAScript2015 (ES2015/ES6).

Solidity

For the two smart contracts (NFC, AQP, ALP), RMC uses Solidity, the most common programming language to write dApps on the Ethereum Blockchain. „Solidity is a contract-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python and JS and is designed to target the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features" (Solidity 2018). In that regard, RMC uses Solidity version 0.4.24 and above.

To be able to create the non-fungible cases, meaning, to provide with the ERC721 token standards, RMC's NFC contracts take OpenZeppelin's (a library for secure smart contract development) non-fungible token contracts and build upon them to be able to deal with issues such as ownership, transfer, pricing, and supply.