**BlockChain : decentralized data-structure immutables (ex network :** Ethereum network)

https://web3j.io/

<https://docs.web3j.io/>

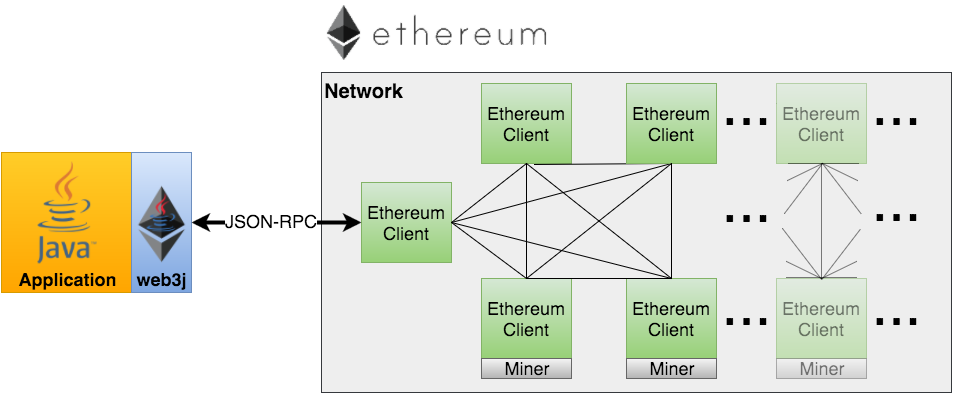
https://dzone.com/articles/spring-boot-and-web3j-easy-microservices-for-the-e?fromrel=true

### web3j seamlessly integrates with Spring Frameworks. Simply create a new Spring Boot app, add the starter to your configuration, and create a service with web3j.

* **web3j** is a highly modular, reactive, type safe Java and Android library for working with Smart Contracts and integrating with clients (nodes) on the Ethereum network:

This allows you to work with the [Ethereum](https://www.ethereum.org/) blockchain, without the additional overhead of having to write your own integration code for the platform.

The [Java and the Blockchain](https://www.youtube.com/watch?v=ea3miXs_P6Y) talk provides an overview of blockchain, Ethereum and web3j.



# Features

* Complete implementation of Ethereum’s [JSON-RPC](https://github.com/ethereum/wiki/wiki/JSON-RPC) client API over HTTP and IPC
* Ethereum wallet support
* Auto-generation of Java smart contract wrappers to create, deploy, transact with and call smart contracts from native Java code ([Solidity](http://solidity.readthedocs.io/en/latest/using-the-compiler.html#using-the-commandline-compiler) and [Truffle](https://github.com/trufflesuite/truffle-contract-schema) definition formats supported)
* Reactive-functional API for working with filters
* [Ethereum Name Service (ENS)](https://ens.domains/) support
* Support for Parity’s [Personal](https://github.com/paritytech/parity/wiki/JSONRPC-personal-module), and Geth’s [Personal](https://github.com/ethereum/go-ethereum/wiki/Management-APIs#personal) client APIs
* Support for [Infura](https://infura.io/), so you don’t have to run an Ethereum client yourself
* Comprehensive integration tests demonstrating a number of the above scenarios
* Command line tools
* Android compatible
* Support for JP Morgan’s Quorum via [web3j-quorum](https://github.com/web3j/quorum)