#MachineLearning
#Blockchain
@EclipseScout

June 2017, @ZimMatthias


Deeplearning4j: Open-sc
Secure | https://deeplearning4j.org
DEEPLEARNING4J
QUICKSTART
DOCUMENTATION
GPUS
SPARK
WORD2VEC
ABOUT
FAQ
Fork me on GitHub
Deep Learning for Java
Open-Source
web3j — web3j 2.2.1 doc
Secure | https://docs.web3j.io
web3j
web3j is a lightweight, reactive, type safe Java and Android library for integrating with clients (nodes) on the Ethereum network:
ethereum
Network
Ethereum Client
Miner
Java Application
web3j
JSON-RPC
Lightweight Java library for integration with Ethereum clients
Star 141
build passing
codecov 80%
Navigation
Getting Started
Transactions
This allows you to work with the Ethereum blockchain, without the additional overhead of having to write your own integration code for the platform.
The Java and the Blockchain talk provides an overview of blockchain, Ethereum and web3j.

Eclipse Scout

What is Eclipse Scout?

Business Appliation Framework

- ➔ Open Source **Eclipse Project**
- ➔ Based on **Java** and **HTML5**
- ➔ **Multi Device** support, **Modular** Apps, …

Goals

- ➔ **Long Term** Sustainability (enterprise apps live > 10 years)
- ➔ Boosts **Productivity** (producing software in Switzerland …)
- ➔ **Easy** to learn (newbies productive in 1-2 weeks)

Eclipse Scout The Java Story

Application Model

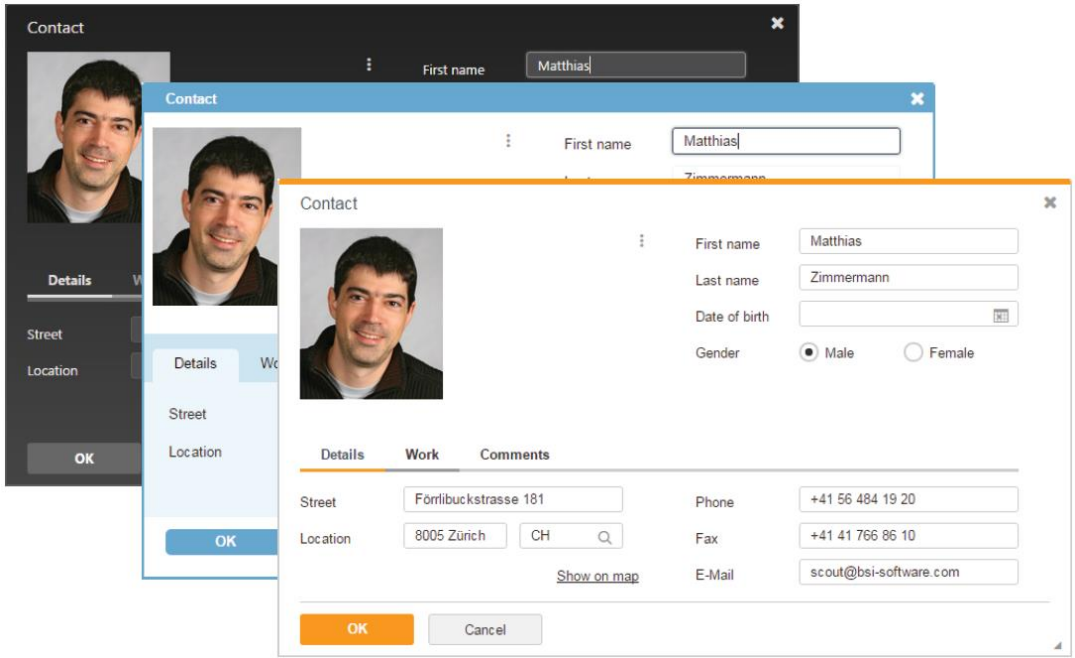
- ➔ (Very) Long Term
- ➔ Clean Business Code
- ➔ High Maintainability


```
@Order(10)
public class FirstNameField extends AbstractStringField {

    @Override
    protected String getConfiguredLabel() {
        return TEXTS.get("FirstName");
    }
}
```

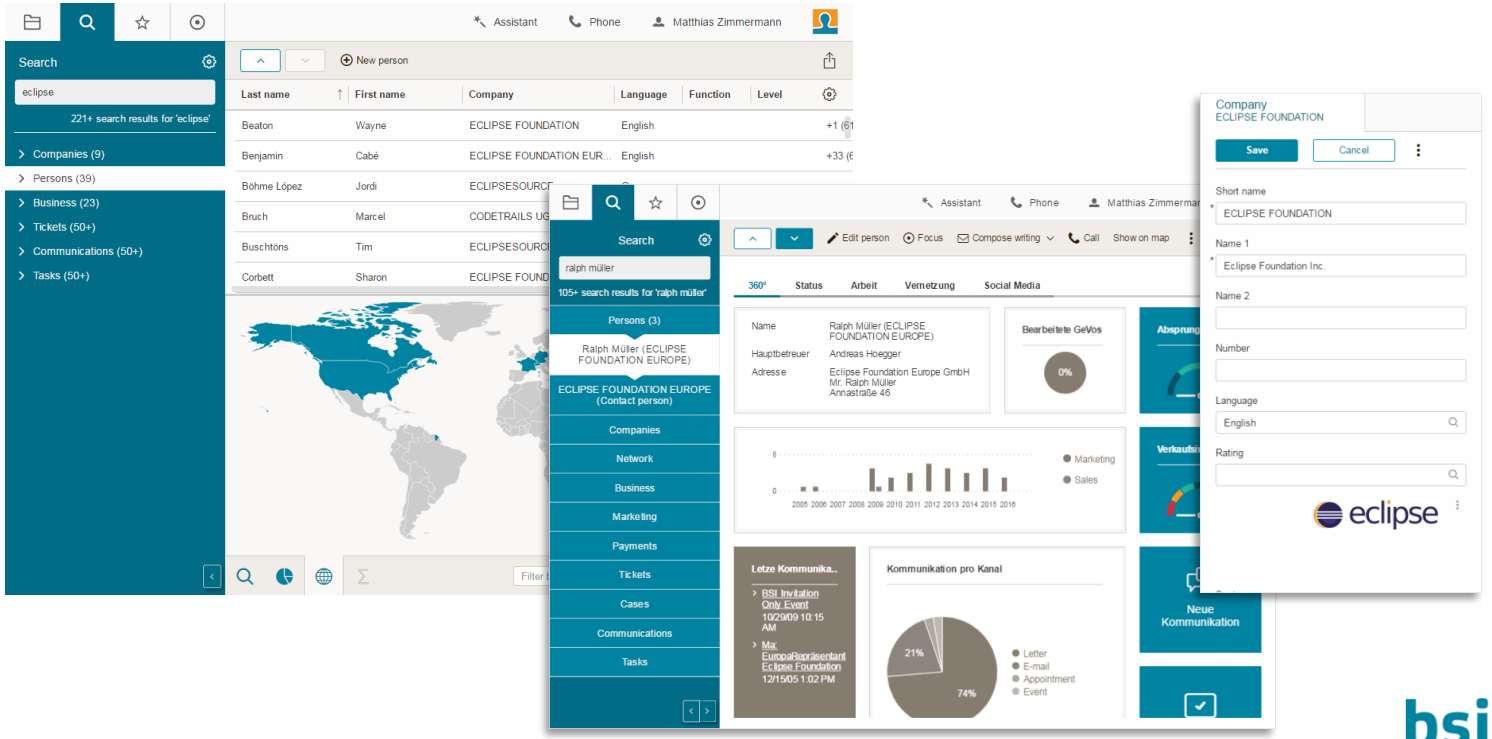
Eclipse Scout The HTML5 Story

Rendering

- ➔ HTML5, CSS3, JavaScript
- ➔ Styling & theming

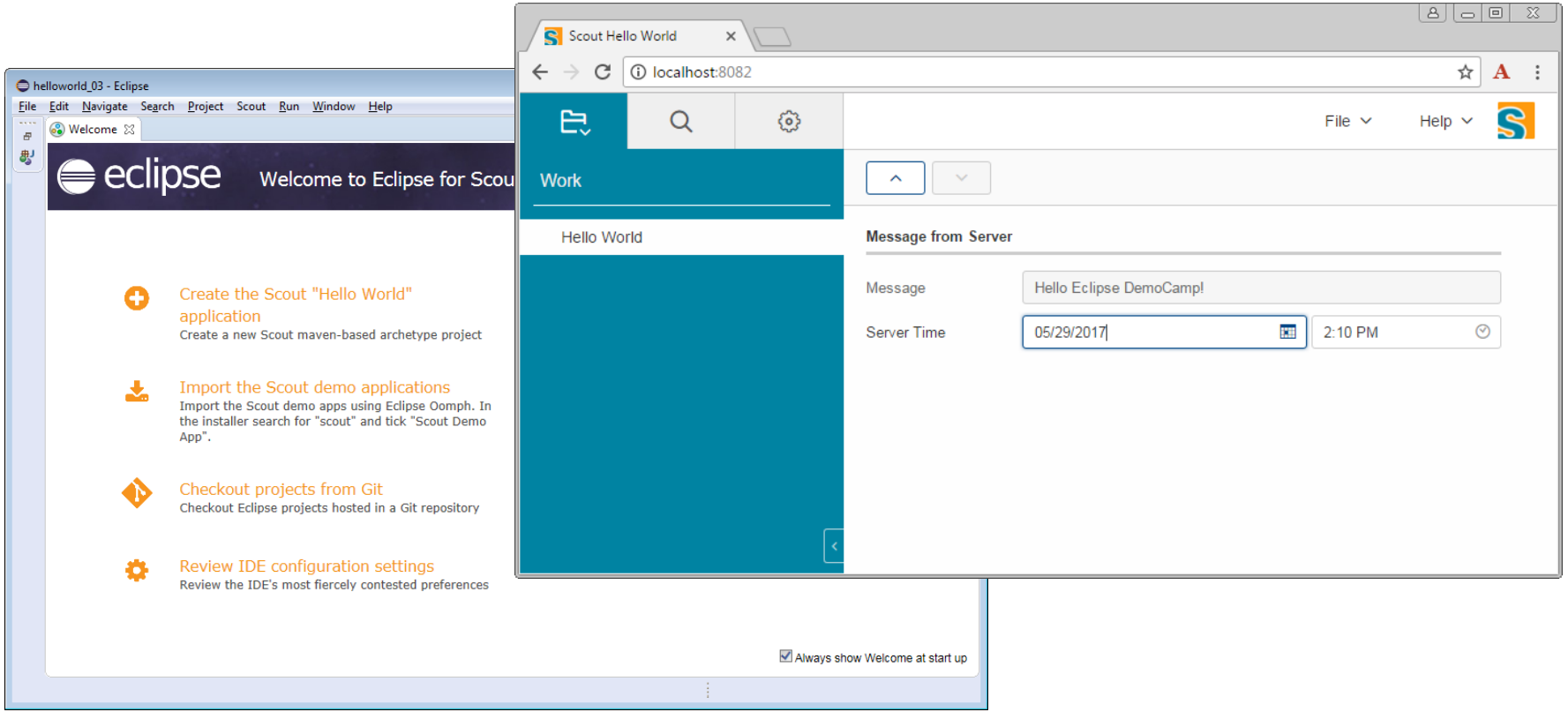


Eclipse Scout Commercial Application



bsi·crm

Eclipse Scout Hello World



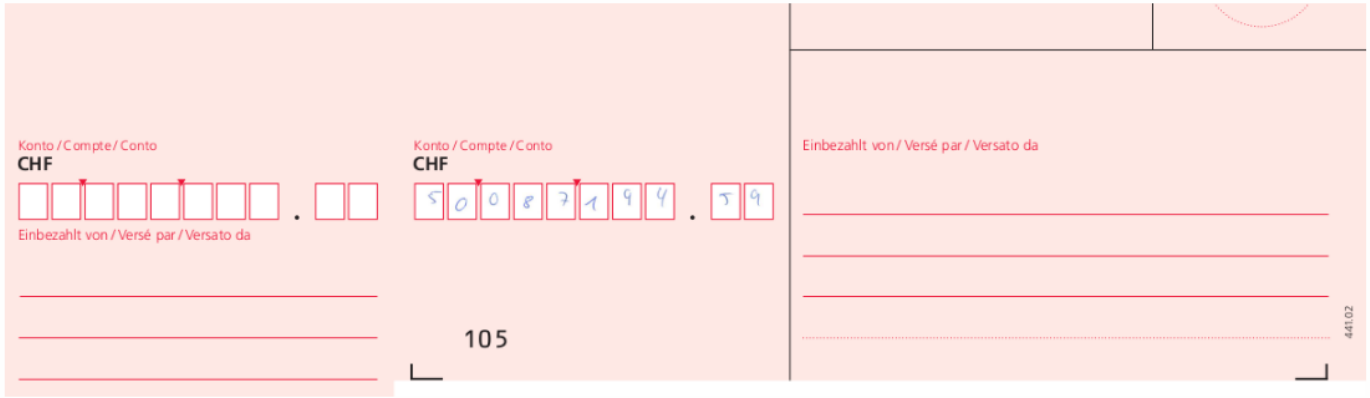
Machine Learning
Deeplearning4j

Anagnostes

localhost:8086

0021_CH4M

Next Scan



5	0	0	8	7	1	9	4	.	5	4
1.000	0.991	0.992	1.000	1.000	1.000	0.821	0.934	-	0.805	0.582

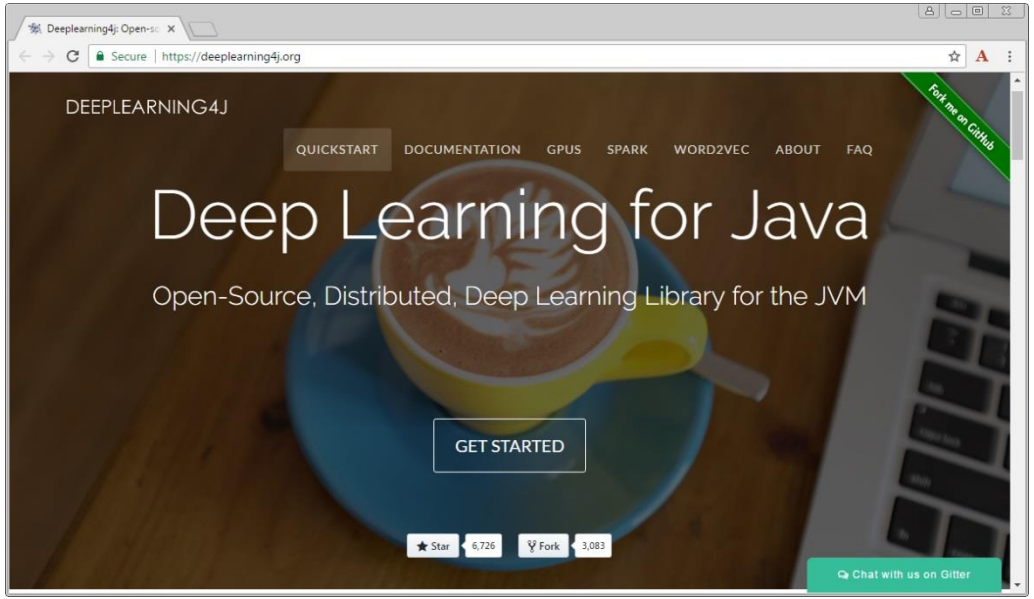
Deeplearning4j

Deep Learning Library

- Open Source
- **Java** (most are Python)
- Good documentation

Features

- Full GPU support
- Distributed deep learning
- Runs with Hadoop + Spark



`https://github.com/deeplearning4j/deeplearning4j`

The ML «Hello Wo
Recognition of han

PROC. OF THE IEEE, NOVEMBER 1998

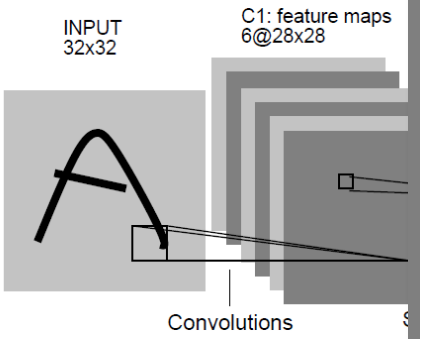


Fig. 2. Architecture of LeNet-5, a Convolut
whose weights are constrained to be ide

1998 **Gradient-base**

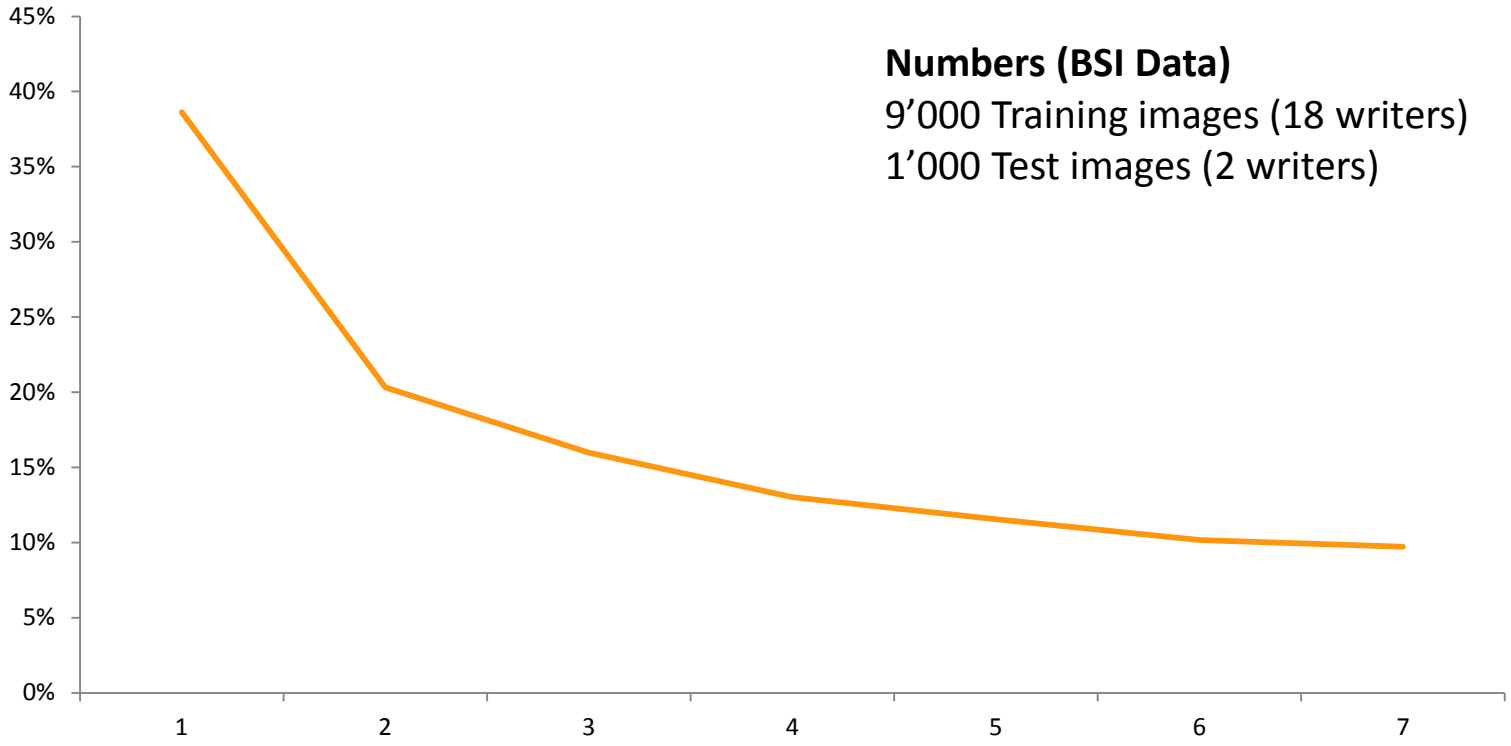
```
public static MultiLayerConfiguration configuration() {
    return new NeuralNetConfiguration.Builder()
            .seed(SEED).weightInit(WeightInit.XAVIER)
            .iterations(NUM_ITERATIONS)
            .regularization(true).l2(0.0005).learningRate(.01)
            .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
            .updater(Updater.NESTEROVS).momentum(0.9)
            .list()
            .layer(0, new ConvolutionLayer.Builder(5, 5)
                    .stride(1, 1)
                    .nIn(NUM_CHANNELS)
                    .nOut(20)
                    .activation(Activation.IDENTITY)
                    .build())
            .layer(1, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX)
                    .kernelSize(2, 2)
                    .stride(2, 2)
                    .build())
            .layer(2, new ConvolutionLayer.Builder(5, 5).stride(1, 1)
                    .nOut(50)
                    .activation(Activation.IDENTITY)
                    .build())
            .layer(3, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX)
                    .kernelSize(2, 2)
                    .stride(2, 2)
                    .build())
            .layer(4, new DenseLayer.Builder()
                    .activation(Activation.RELU)
                    .nOut(500)
                    .build())
            .layer(5, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
                    .activation(Activation.SOFTMAX)
                    .nOut(NUM_OUTPUTS)
                    .build())
            .setInputType(InputType.convolutionalFlat(28, 28, 1))
            .backprop(true)
            .pretrain(false).build();
}
```

Handwritten Data

Training of a Neural Network Model

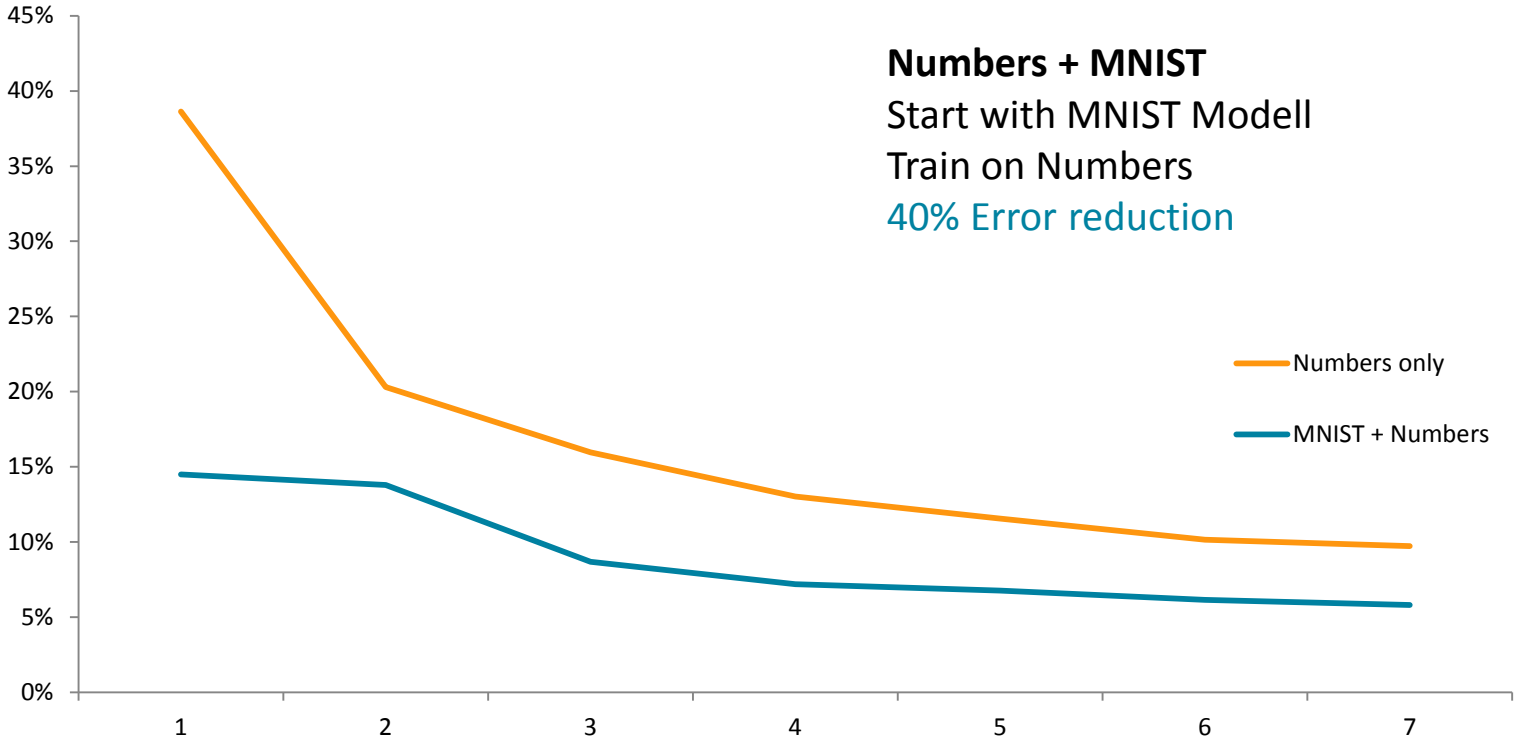
```
/**
 * Train the network for the specified number of epochs.
 */
public void train(DataSetIterator trainData, DataSetIterator validationData, int epochs) {
    for(int epoch = 1; epoch <= epochs; epoch++) {

        // train the network using training data
        log.info("Starting epoch {}, samples: {}", epoch, trainData.numExamples());
        trainData.reset();
        m_network.fit(trainData);

        // evaluate performance using validation data
        validationData.reset();
        evaluate(validationData);
    }
}
```

Error Rate and Training Epochs



Numbers (BSI Data)
9’000 Training images (18 writers)
1’000 Test images (2 writers)

Error Rate and Training Epochs



Numbers + MNIST
Start with MNIST Modell
Train on Numbers
40% Error reduction

Blockchain

web3j/Ethereum

Blockchain «Micro-Intro»

Blockchain

- **Bitcoin** started the field in 2009
- Main Features: Cheap, fast, efficient (traditional setup: T+3 and high fees)
- **Ethereum** adds smart contracts
- Remarkable achievements and much hype

Main Challenges

- Scalability
- Privacy
- Regulatory & legal

Ethereum Smart Contracts

What is it?

- ➔ Piece of byte code (usually written in Solidity)
- ➔ Is executed by the Ethereum Virtual Machine (EVM)
- ➔ Has an owner

Examples

- ➔ Greeter (the Ethereum «Hello World»)
- ➔ «Truly» autonomous cars

Ethereum «Hello World»

```
contract greeter {

    /* Owner of this contract */
    address owner;

    /* Configurable greeting */
    string greeting;

    /* Constructor runs when contract is deployed */
    function greeter(string _greeting) public {
        owner = msg.sender;
        greeting = _greeting;
    }

    /* Main function */
    function greet() constant returns (string) {
        return greeting;
    }

    /* Function to recover the funds on the contract */
    function kill() {
        if (msg.sender == owner)
            selfdestruct(owner);
    }
}
```

«Truly» Autonomous Cars

Uber's self-driving cars are now picking up passengers in Arizona

Tempe or bust

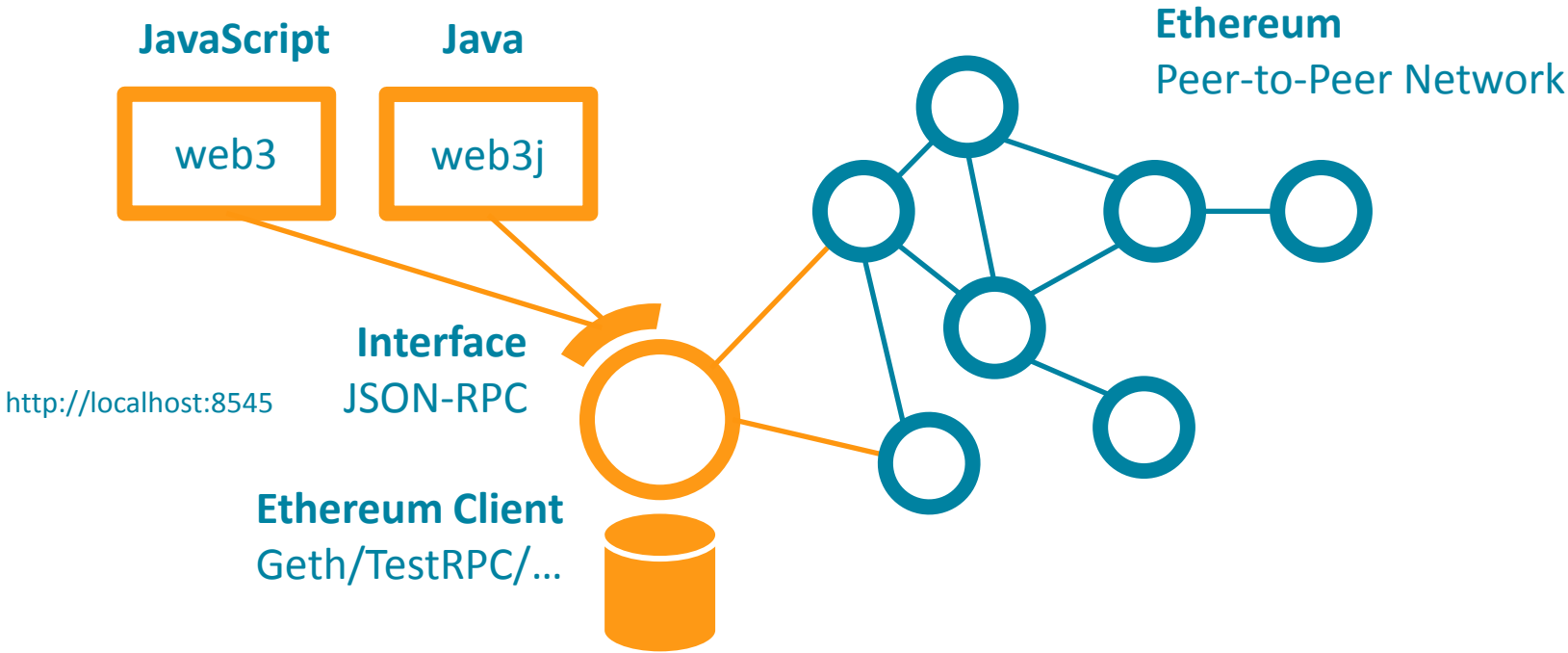
by Andrew J. Hawkins | @andyjayhawk | Feb 21, 2017, 1:55pm EST

SHARE TWEET LINKEDIN

A subsidiary of RWE, one of Germany's biggest energy and gas provider with 30 million customers and billions of revenue, has launched 100s of electronic vehicles (EV) charging stations all over Germany, connected to ethereum's public blockchain.

- ➔ Smart contract: To order car to transport people (by paying to contract)
- ➔ Smart contract: Car pays for energy/services

Ethereum and Application Integration



web3j

Library to interact with Ethereum (its peer-to-peer clients)

- ➔ Open Source
- ➔ **Java** (default is JavaScript)
- ➔ Good documentation

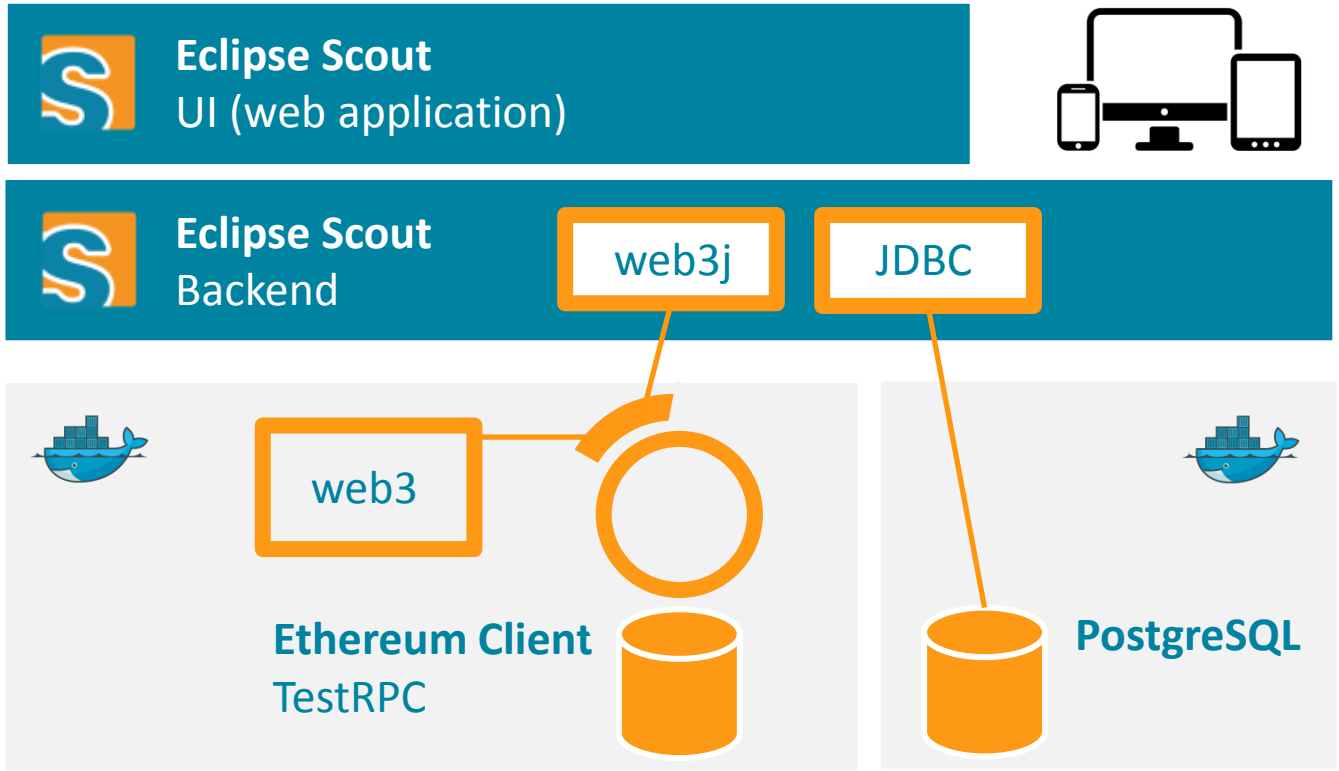
Features

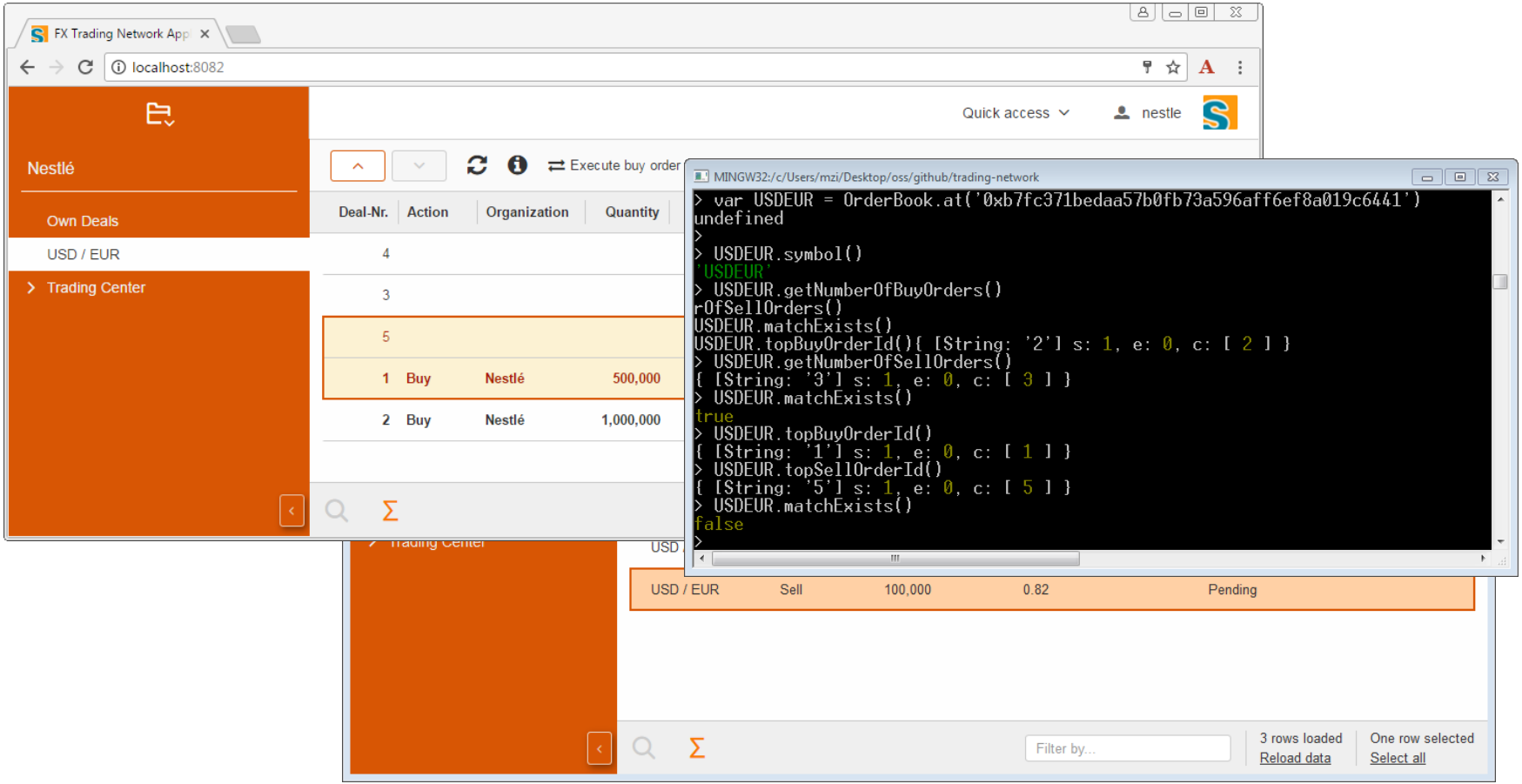
- ➔ JSON-RPC client API implementation
- ➔ Tool to generate **Contract Wrappers** in **Java**

`https://github.com/web3j/web3j`

Web3j: Generated Contract Wrapper

```
/**
 * <p>Auto generated code.<br>
 * <strong>Do not modify!</strong><br>
 * Please use {@link org.web3j.codegen.SolidityFunctionWrapperGenerator} to update.
 *
 * <p>Generated with web3j version 2.1.0.
 */
public final class Greeter extends Contract {
    private static final String BINARY = "0x60606040523461000057604051610e33803806102e383398101604052805101 5b60008054600160a060020a0319166c

    private Greeter(String contractAddress, Web3j web3j, Credentials credentials, BigInteger gasPrice, BigInteger gasLimit) {
        super(contractAddress, web3j, credentials, gasPrice, gasLimit);
    }

    private Greeter(String contractAddress, Web3j web3j, TransactionManager transactionManager, BigInteger gasPrice, BigInteger gasLimit) {
        super(contractAddress, web3j, transactionManager, gasPrice, gasLimit);
    }

    public Future<Uint256> deposits() {
        Function function = new Function("deposits",
                Arrays.<Type>asList(),
                Arrays.<TypeReference<?>>asList(new TypeReference<Uint256>() {}));
        return executeCallSingleValueReturnAsync(function);
    }
```


Eclipse Scout
UI (web application)
Eclipse Scout
Backend
web3j
JDBC
web3
Ethereum Client
TestRPC
PostgreSQL


FX Trading Network App
localhost:8082
Quick access
nestle
Nestlé
Own Deals
USD / EUR
Trading Center
Execute buy order
Deal-Nr.
Action
Organization
Quantity
4
3
5
1 Buy Nestlé 500,000
2 Buy Nestlé 1,000,000
MINGW32:/c/Users/mzi/Desktop/oss/github/trading-network
> var USDEUR = OrderBook.at('0xb7fc371bedaa57b0fb73a596aff6ef8a019c6441')
undefined
> USDEUR.symbol()
'USDEUR'
> USDEUR.getNumberOfBuyOrders()
rOfSellOrders()
USDEUR.matchExists()
USDEUR.topBuyOrderId(){ [String: '2'] s: 1, e: 0, c: [2] }
> USDEUR.getNumberOfSellOrders()
{ [String: '3'] s: 1, e: 0, c: [3] }
> USDEUR.matchExists()
true
> USDEUR.topBuyOrderId()
{ [String: '1'] s: 1, e: 0, c: [1] }
> USDEUR.topSellOrderId()
{ [String: '5'] s: 1, e: 0, c: [5] }
> USDEUR.matchExists()
false
>
USD / EUR Sell 100,000 0.82 Pending
Filter by...
3 rows loaded
Reload data
One row selected
Select all

Thanks!

@ZimMatthias