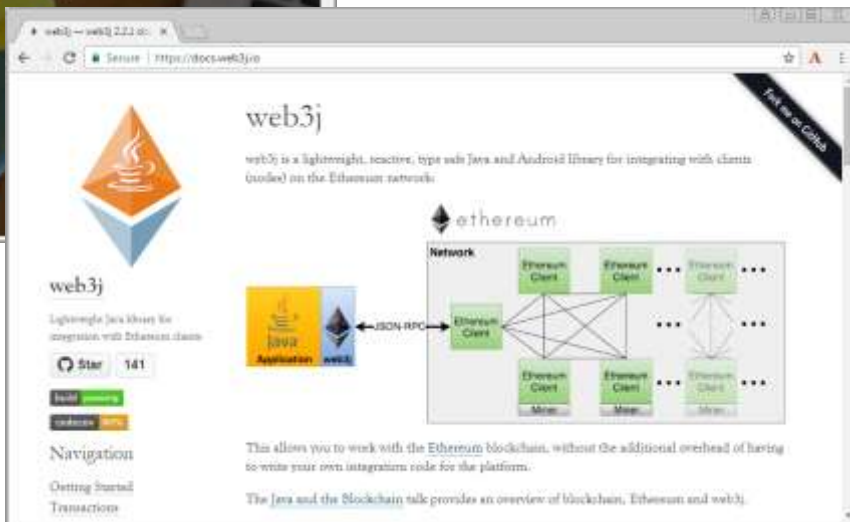


Machine Learning, Blockchain and Eclipse Scout

@EclipseScout by @ZimMatthias



Eclipse Scout

What is Eclipse Scout?

Business Application Framework

- Based on **Java** and **HTML5**
- **Multi Device** support, **Modular** Apps, ...

Framework Goals

- **Long Term** Strategy (enterprise apps live > 10 years)
- Boosts **Productivity** (producing software in Switzerland ...)
- **Easy** to learn (new team members productive in 1-2 weeks)

Eclipse Scout Application Model

Java Application Model

- Clean Business Code
- High Maintainability

First name

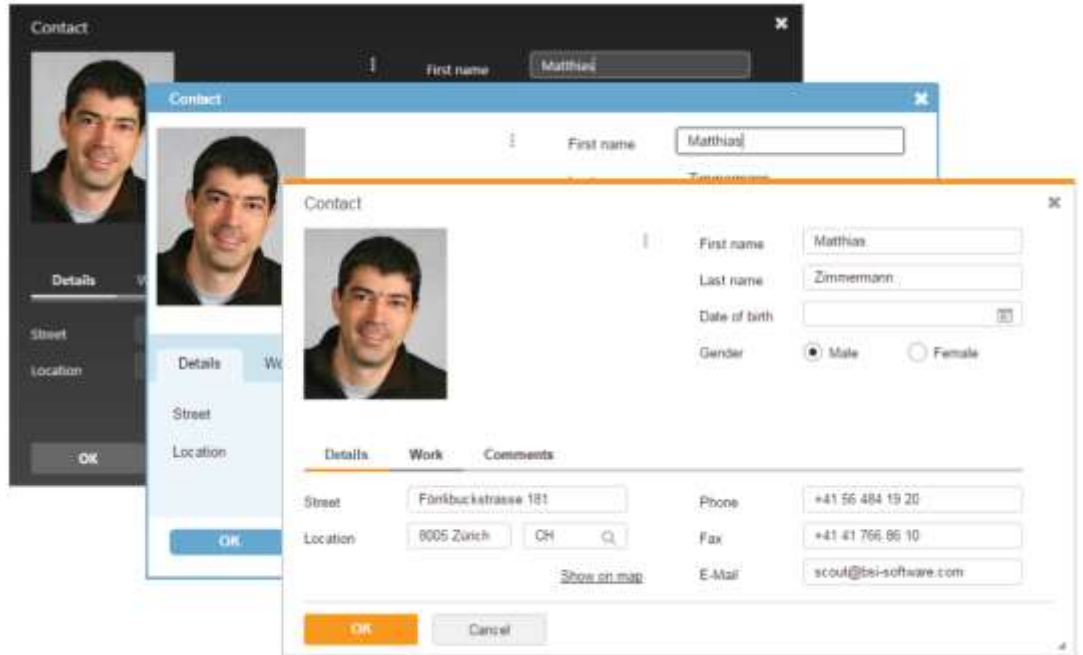
```
@Order(10)
public class FirstNameField extends AbstractStringField {

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

Eclipse Scout HTML5 Rendering

Current Standards

- HTML5, CSS3, JavaScript
- Flexible Styling
- Theming



Eclipse Scout Commercial Application

The screenshot displays the Eclipse Scout CRM interface. On the left, a search sidebar shows results for 'eclipse' and 'fabrizio', with categories like Companies (9), Persons (39), Business (20), Tickets (30), Communications (38), and Tasks (30). The main area shows a list of search results for 'fabrizio' with columns for Last name, First name, Company, Language, and Phone. A contact profile for 'Ralph Müller (ECLIPSE FOUNDATION EUROPE)' is shown, including a bar chart for 'Kommunikation pro Kanal' and a pie chart for 'Kommunikation pro Kanal'. A form for 'Company ECLIPSE FOUNDATION' is overlaid on the right, with fields for Short name, Name 1, Name 2, Number, Language, and Rating. The Eclipse logo is visible at the bottom of the form.

Last name	First name	Company	Language	Phone	Level
Bastian	Wolke	ECLIPSE FOUNDATION	English		<100
Dargatzis	Cobi	ECLIPSE FOUNDATION EUR...	English		<100
Bilvin Lopez	Javi	ECLIPSEFOUN...			
Beck	Marie	CODETRAIL U...			
Buchheit	Ty	ECLIPSEFOUN...			
Carbal	Shan	ECLIPSEFOUN...			

Company ECLIPSE FOUNDATION

Save Cancel

Short name
ECLIPSE FOUNDATION

Name 1
Eclipse Foundation Inc.

Name 2

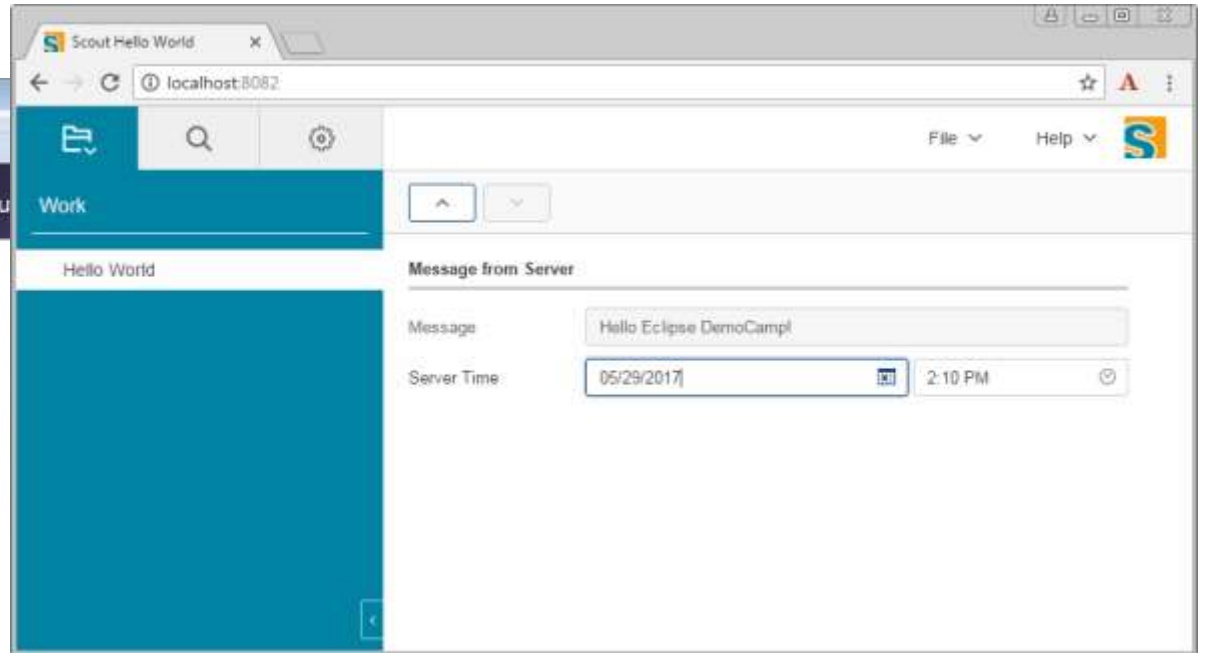
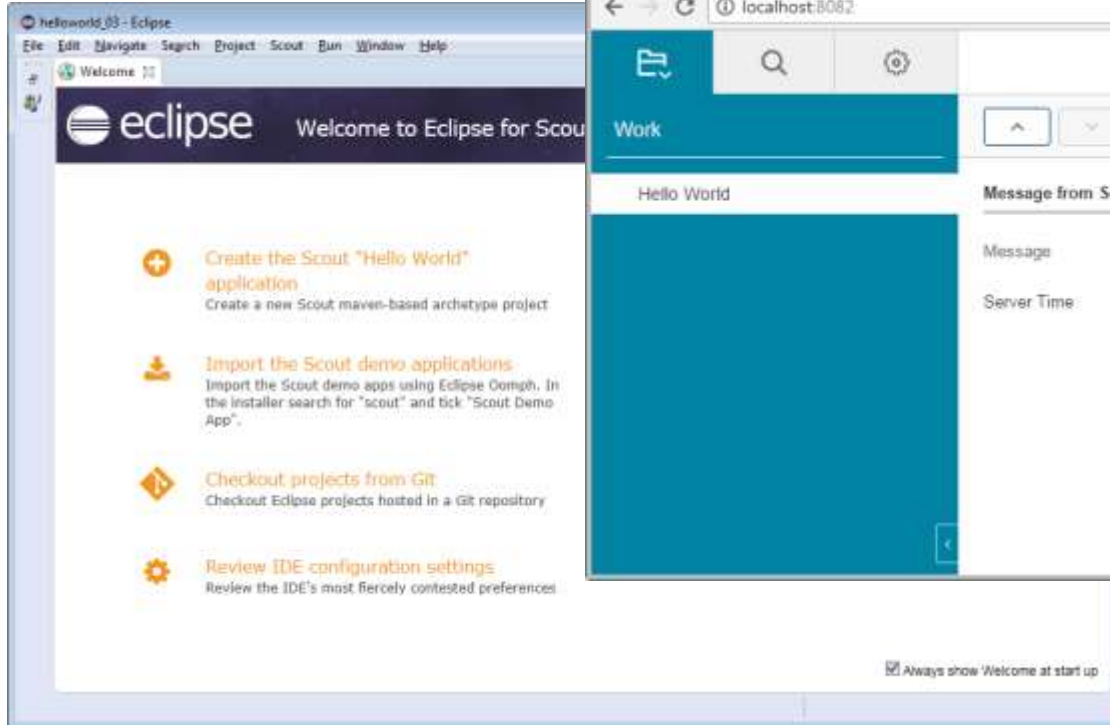
Number

Language
English

Rating

eclipse

Eclipse Scout Hello World



Machine Learning

Deeplearning4j demo

What's behind the Hype?

Machine learning (ML) actually starts to work ...

Many robust ML libraries / frameworks to choose from

Innovation driven by Open Source

- People able to replicate results
- Open Source software
- Open Data Collections
- Open Publications (arXiv)

ML performance \geq Human Levels (2017)

Games Backgammon 1979, chess 1997, Jeopardy! 2011,
Atari games 2014, Go 2016

Visual Face recognition 2007, traffic sign reading 2011,
lip-reading 2016

Other Personality judgement via social media 2014,
conversational speech recognition 2016

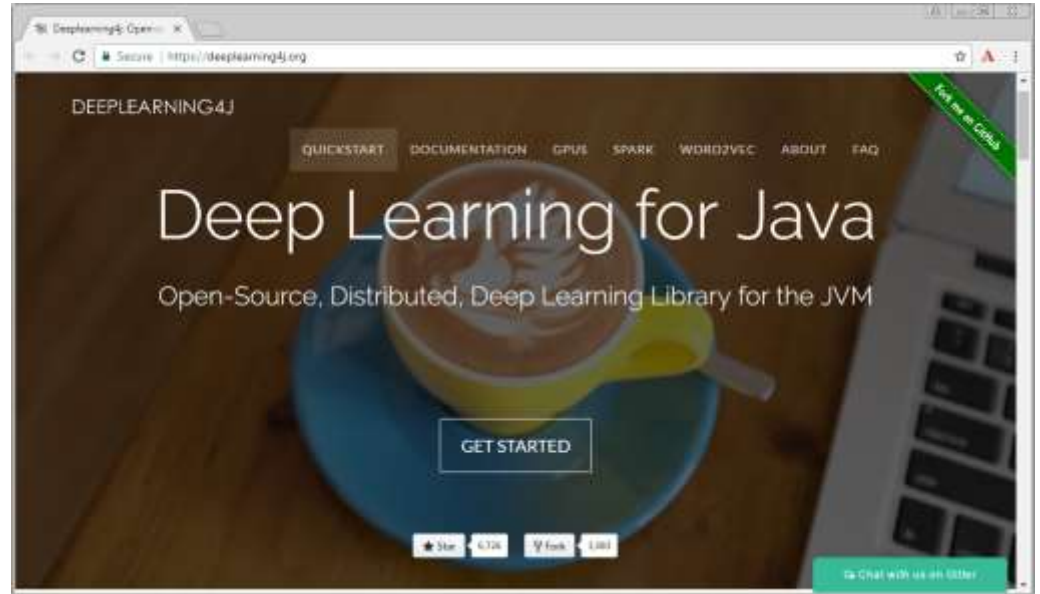
Deeplearning4j

Deep Learning Library

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

Features

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



<https://github.com/deeplearning4j/deeplearning4j>

The ML «Hello World» Recognition of handwritten digits

PROC. OF THE IEEE, NOVEMBER 1998

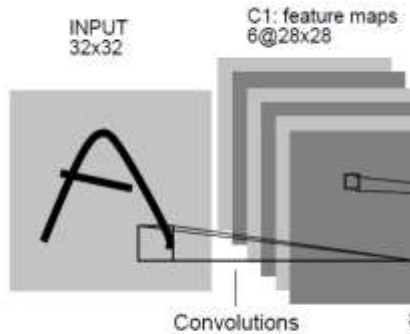
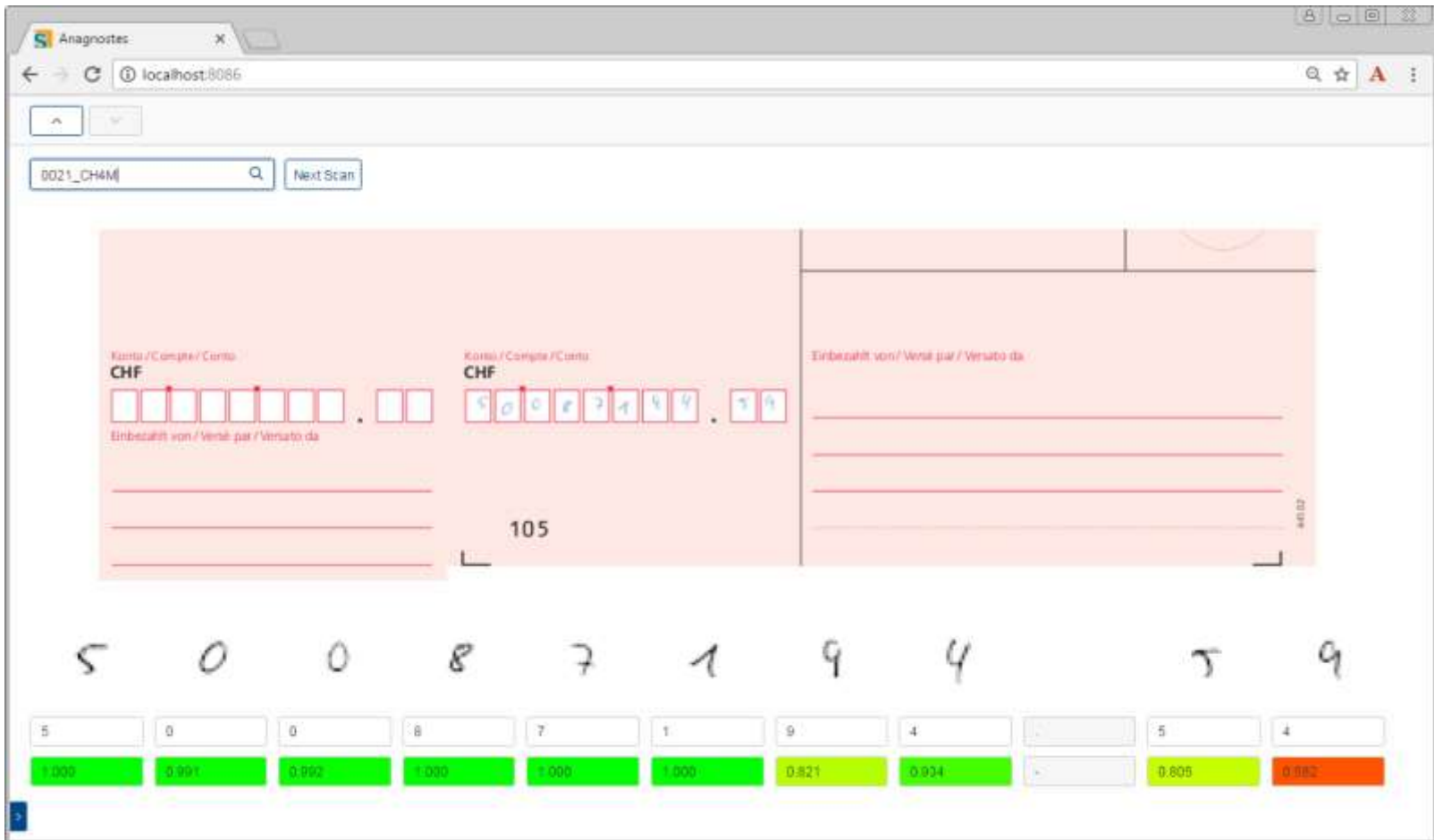


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network whose weights are constrained to be identical in a neighborhood.

1998 Gradient-based

```
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();  
}
```



Blockchain

web3j demo with Ethereum

Blockchain «Micro-Intro»

Blockchain

- **Bitcoin** here since 2009
- Cheap, fast, reliable, efficient (compare with T+3 and high fees)
- **Ethereum** adds smart contracts
- Increasing amounts of money ...

Main Challenges

- Scalability
- Privacy

Bitcoin



USD

CNY

1h 12 3m 1y All

Jul 18, 2010 to May 29, 2017 Export

\$2000
\$1500
\$1000



Bitcoin (24h)

USD ▲ 1.33%

\$2,237.04

EUR €1,999.69

CNY ¥14,529.23

GBP £1,741.60

\$2272



Ethereum

USD

1d 5w 1m 3m 1y JUL

-Sep 1, 2015 to May 28, 2017 Export

\$1200
\$1000
\$500
\$0



Sep 15 Jan 16 May 16 Sep 16 Jan 17 May 17

Bitcoin ▲ 1.29%

\$2,236.03

Ethereum (24h)

USD ▲ 7.91%

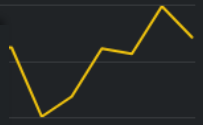
\$171.38

EUR €153.26

CNY ¥1,174.53

GBP £133.71

\$211
\$176
\$140



\$171.38

Smart Contracts

What is a Ethereum Smart Contract?

- Piece of (byte) code
- Is executed by the Ethereum Virtual Machine (EVM)
- Has an owner
- Has a life cycle

Example: «Truly» autonomous cars

«Truly» Autonomous Cars

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

Tempe or bust

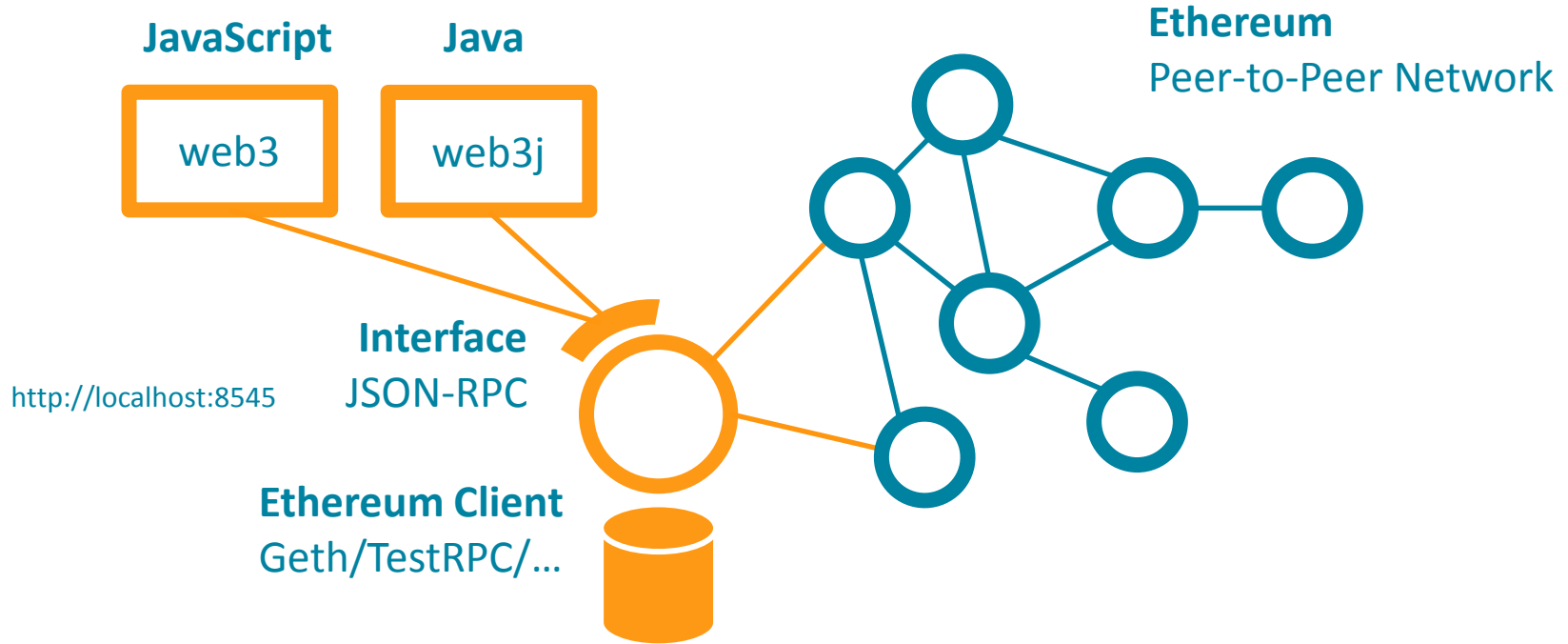
by Andrew J. Hawkins | @andjehawk | Feb 21, 2017, 1:55pm CST



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)

- **Java**
- Good documentation
- Open Source

Features

- JSON-RPC client API implementation
- Command line tools to generate **Java contract wrappers**



<https://github.com/web3j/web3j>

Smart Contracts Life Cycle

Deploying and using Smart Contracts

1. Write contract in high level language (eg. **Solidity**)
2. Compile contract to **EVM byte-code**
3. Pack byte code into a **contract creation TX** and sent to the network
4. The TX gets its own contract address
5. Invoke contract methods

«Hello World» (greeter.sol)

```
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);  
    }  
}
```

Solidity Compiler (online)

The screenshot displays the online Solidity IDE interface. On the left, the Solidity source code is visible, defining a contract named `greeter` with a constructor, a `payable` function, and a `greet` function. On the right, the compiler output shows the generated `Bytecode` (hexadecimal string) and the `Web3 deploy` script (JavaScript code). An orange callout box points to the `Bytecode` section with the text "byte code (EVM) to deploy contract". Another orange callout box points to the `Web3 deploy` section with the text "Deploy script (JS)".

```
1 pragma solidity ^0.4.6;
2
3 contract greeter {
4
5     /* Owner of this contract */
6     address owner;
7
8     /* Counter for deposits calls */
9     uint public deposits;
10
11    /* Configurable greeting */
12    string greeting;
13
14    /* Constructor runs when contract is deployed */
15    function greeter(string _greeting) public {
16        owner = msg.sender;
17        greeting = _greeting;
18        deposits = 0;
19    }
20
21    /*
22     * Default function.
23     * 'payable': Allows to move funds to contract.
24     * Changes state: Costs gas and needs contract transaction.
25     */
26    function() payable {
27        deposits += 1;
28    }
29
30    /* Main function */
31    function greet() public {
32        return greeting;
33    }
34
35    /* Function to rec
36    function kill() {
37        if (msg.sender
38            selfdestru
39
40 }
```

Bytecode: 6060604052341561000c57fe5b604051610407380380610

Interface: [{"constant":true,"inputs":[],"name":"deposits"}]

Web3 deploy: var _greeting = /* var of type string here */ ;
var undefined_greeterContract = web3.eth.contra
var undefined_greeter = undefined_greeterContra
_greeting,
{
 from: web3.eth.accounts[0],
 data: '0x6060604052341561000c57fe5b6040516
 gas: '4700000'
}, function (e, contract){
 console.log(e, contract);
 if (typeof contract.address !== 'undefined'
 console.log('contract mined! address:

web3j: greeter.sol → Greeter.java

From Solidity to Java Contract Class

1. Compile `greeter.sol` (e.g using online compiler)
→ `greeter.bin`, `greeter.abi`
2. Create contract wrapper class (use Web3j command line tool)
→ `Greeter.java`
1. Use `Greeter.java` in your Java code

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 = "0x606060405234610000576040516102e33803806102e3833981016040528051015b60008054600160a060020a0319166c

    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);
    }
}
```

Ethereum, web3j + Eclipse Scout

Trading Network Demo

- **Currency Hedging**: Buy orders and Sell orders (€ / US\$)
- **Classical Business App**
 - **Identity Management** for mapping real persons \leftrightarrow Blockchain addresses
 - **User Interface**
- **Blockchain Benefits**
 - **Efficiency**: No central organization/infrastructure
 - **Trust**: Tampering-proof ledger, trust by blockchain technology



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

```

> var USDEUR = OrderBook.at('0xb7fc371bedaa57b0fb79a596aff6ef8a019c6441')
undefined
>
> USDEUR.symbol()
'USDEUR'
> USDEUR.getNumberOfBuyOrders()
rOfSellOrders()
USDEUR.matchExists()
USDEUR.topBuyOrderId() [ IString: '2' s: 1, e: 0, c: [ 2 ] ]
> USDEUR.getNumberOfSellOrders()
[ IString: '3' s: 1, e: 0, c: [ 3 ] ]
> USDEUR.matchExists()
true
> USDEUR.topBuyOrderId()
[ IString: '1' s: 1, e: 0, c: [ 1 ] ]
> USDEUR.topSellOrderId()
[ IString: '5' s: 1, e: 0, c: [ 5 ] ]
> USDEUR.matchExists()
false
>
  
```

USD / EUR	Sell	100,000	0.82	Pending
-----------	------	---------	------	---------

3 rows loaded | One row selected

Reload data | Select all

Thanks!

@ZimMatthias