syntegris information solutions

PLCHESSQL

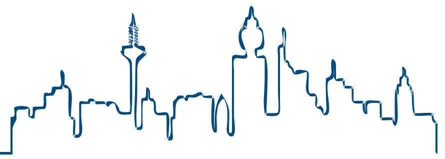
- SCHACH MIT APEX, NODEJS UND TWITTER

AUTHOR: MORITZ SCHENKEL

www.syntegris.de

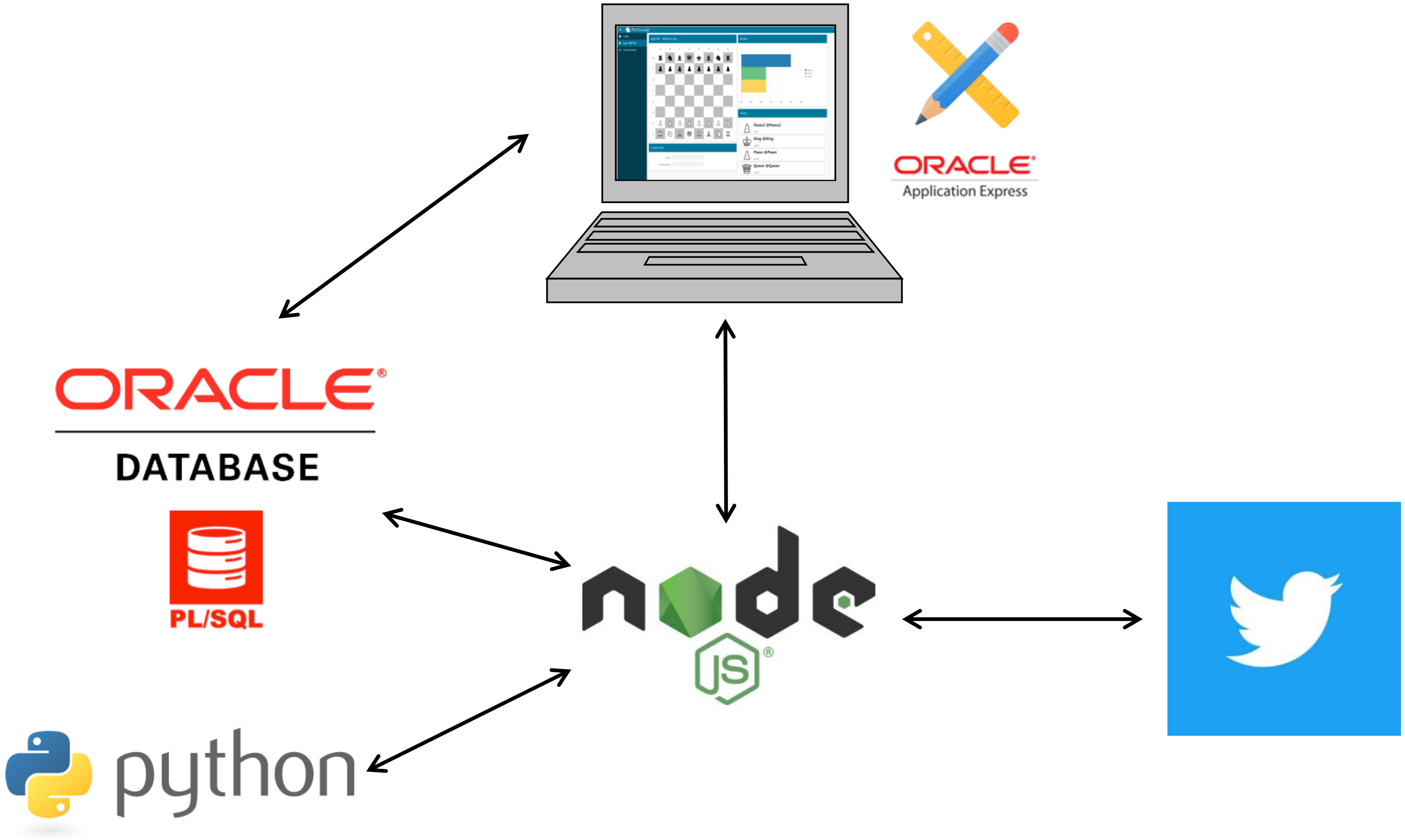
whoami

- 31 years old
- married
- 2 cats
- MSc Strategic Information Management
- Consultant at Syntegris information solutions GmbH
- Enthusiastic about new technologies



Demo !

Infrastructure

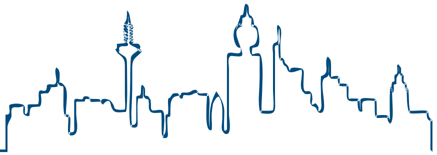


NodeJS / Code Review!

syntegris

NodeJS Code Review
- package.json

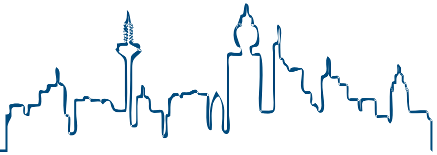
```
{
  "name": "chess_tweet",
  "version": "1.0.0",
  "description": "app to communicate with twitter and the apex application",
  "main": "tweet.js",
  "dependencies": {
    "moment": "^2.18.1",
    "node-cron": "^1.2.1",
    "oracledb": "1.13.1",
    "python-shell": "0.4.0",
    "twit": "2.2.5",
    "websocket": "1.0.24"
  },
  "author": "MSC",
  "license": "ISC"
}
```



NodeJS Code Review

declaration and used packages

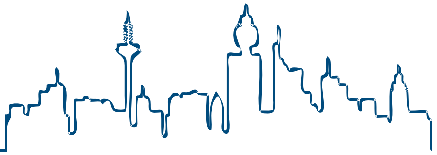
```
// Optional. You will see this name in eg. 'ps' or 'top' command
process.title = 'chess_tweet';

// websocket and http servers
var webSocketServer = require('websocket').server;
var http = require('http');

// wrapper for oracleDB
var database = require('./dbWrapper.js');

// twitter api
var twit = require('twit');
var twitterCredentials = require('./twitter.js')
var twitterAPI = new twit(twitterCredentials);

// Module which tweets the Board for the gameId provided
var tweetImage = require('./tweetImage.js');

// Module which gives cron support to nodeJS
var cron = require('cron');

// Module for managing time
var moment = require('moment');
```



NodeJS Code Review

websocket

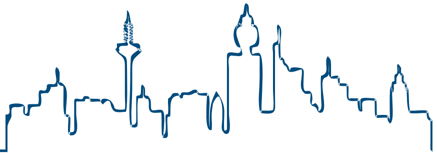
```
var server = http.createServer(function (request, response) {
    // Not that important for us. We're using a WebSocket server, not HTTP server
});
server.listen(webSocketsServerPort, function () {
    console.log((new Date()) + " Twitter Communication Server is listening”);
});
var wsServer = new webSocketServer({
    // WebSocket server is tied to a HTTP server. WebSocket request is just
    // an enhanced HTTP request. For more info http://tools.ietf.org/html/rfc6455#page-6
    httpServer: server
});
// This callback function is called every time someone
// tries to connect to the WebSocket server
wsServer.on('request', function (request) {
    // accept connection - you should check 'request.origin' to make sure that client is
    // connecting from your website - but i won't cause this isn't a productive application
    // (http://en.wikipedia.org/wiki/Same_origin_policy)
    var connection = request.accept(null, request.origin);
    // null in accept() means we don't care about the protocol
    connection.on('message', function (message) {
         // logic here
    }
}
```



NodeJS Code Review

oracleDB - a node-oracledb add-on

```
var oracledb = require('oracledb');
var dbConfig = require('./dbconfig.js');

// callback hell!
// imagine connection pool is already defined
function execute(sql, bindParams, callback) {
    connectionPool.getConnection(function (err, connection) {
        if (err) {
            callback(err);
        } else {
            connection.execute(sql, bindParams, function (err, result) {
                if (err) {
                    connection.release(...)
                    callback(err);
                } else {
                    connection.commit(function (err) {
                            // omitted some code here
                            connection.release(...)
                    })
                }
            })
        }
    })
}
```



NodeJS Code Review

twit – a Twitter API Client for node

```
// initialize streaming API
var stream = twitterAPI.stream('user');

// reaction when getting tweets
stream.on('tweet', function (tweet) {
  // handle tweet json
}
stream.on('direct_message', function (directMsg) {
  // handle DM json
}

// if not handled, server quits on connection loss
stream.on('error', function(err) {
  // handle disconnects
});

// twitterAPI.post() & twitterAPI.get() also possible methods
```

NodeJS Code Review

pythonShell - run python scripts from node

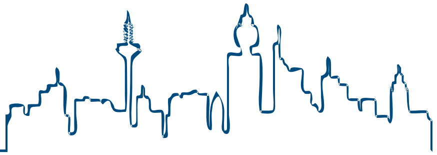
```
var pythonShell = require('python-shell');

// options can be passed on as an object
var options = {mode: 'text',
                    pythonPath: '/usr/bin/python',
                    scriptPath: '/home/oracle/python/fen/',
                    args: ['"' + result + '"', '--outfile=chessboard.png']
                };

pythonShell.run('fen.py', options, function (err, results) {
  // do error handling / stuff with the result
}
```

Apex / Code Review!

APEX Code Review

apex - ajax process

```
// process returns 2dimensional boolean array
apex.server.process('AJAX_POSS_MOVES'
          , { pageItems: "#P10_GAME_ID,#P10_COORD_FROM" }
          , { dataType: "json" // by defining datatype it gets parsed
            , success: function (possMoves) {
                   $('td.t-Report-cell').each(function () {
                     // don‘t color first column
                     if (this.cellIndex > 0) {
                            // get coord x & y
                            var x = this.cellIndex;
                            var y = this.parentNode.innerText.replace(/[\n\r]+/g, '').trim();
                            // if the boolean array says it‘s true – color it
                            if (possMoves['row' + y]['val' + x]) {
                              $(this).addClass("possible-move");
                            }
                     }
                   });
            }
          }
);
```



APEX Code Review

websockets

```
window.WebSocket = window.WebSocket || window.MozWebSocket;

var connection;
connection = new WebSocket('ws://[URL]:[Port]');

connection.onopen = function () {
  // do whatever you want when establishing connection
}

connection.onerror = function (error) {
  // do stuff like error handling, reconnecting, etc.
}

connection.onmessage = function (message) {
  // most important - do stuff with the data you get
}

// send data on your connection
connection.send(JSON.stringify(object));
```

APEX Code Review

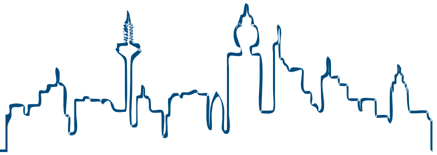
knockout / oracleJet – part 1

```
<!-- HTML in static content region -->
<div id="chart">
  <div id="funnelChart" data-bind="ojComponent: {
    component: 'ojChart',
    type: 'bar',
    series: funnelSeriesVal,
    groups: [''],
    orientation: 'horizontal',
    animationOnDisplay: 'auto',
    animationOnDataChange: 'auto',
    styleDefaults: {threeDEffect: 'on'}
  }"
  style="max-width:500px;width:100%;height:350px;">
  </div>
</div>
```

APEX Code Review

knockout / oracleJet – part 2

```
//create ViewModel
function chartModel() {
  var self = this;

  self.funnelSeriesVal = ko.observableArray();
  self.addVal = function (val) {
         // add value to observable array
         self.funnelSeriesVal.push({ name: val, items: [1] });
         // sort the array
         self.funnelSeriesVal.sort(function (left, right) {
           return left.items[0] == right.items[0] ? 0 : (left.items[0] < right.items[0] ? 1 : -1)
         });
  }
}

// initialize the model
var chart = new chartModel();

// apply bindings to auto-update on change
ko.applyBindings(chart, document.getElementById('chart'));

// add a value (this will be populated by the websocket)
chart.addVal("f7-f5");
```



PLSQL / Code Review!

Questions?

APEX connect

by DOAG

24. - 26. April 2018 in Düsseldorf

apex.doag.org

Thank You!

SYNTEGRIS INFORMATION SOLUTIONS GMBH
HERMANNSTRASSE 54-56
63263 NEU-ISENBURG

FON	(06102) 29 86 68
FAX	(06102) 55 88 06

INFO@SYNTEGRIS.DE
SYNTEGRIS.DE