

The background is a dark blue gradient. In the top-left corner, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram. In the top-right corner, there is a grey, 3D-rendered circuit board pattern. In the bottom-left corner, there is a circular inset showing a detailed image of a printed circuit board (PCB) with various electronic components.

Simple MLE Deployer

A Tool for deploying JavaScript
to Oracles Multilingual Engine

Bachelor thesis of Julian Weyermann
Supervised by Boris Spasojevic



Contents

1. What is MLE?
2. Original Goal & My issues
3. Rescoping & Solution: Simple MLE Deployer
4. Demo
5. Conclusion



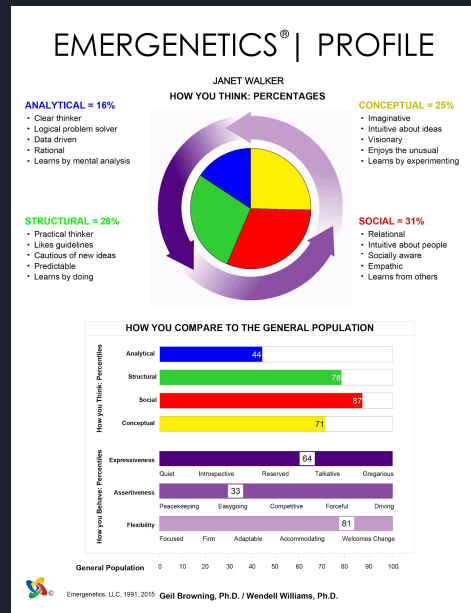
What is MLE? And what is it about?

- MLE = Multilingual Engine
- Extension of the Oracle Database
- Bring other languages to the Database (besides Java & C)
 - Language selection based on needs
 - Libraries
- Based on GraalVM
- Speed and resource benefits!

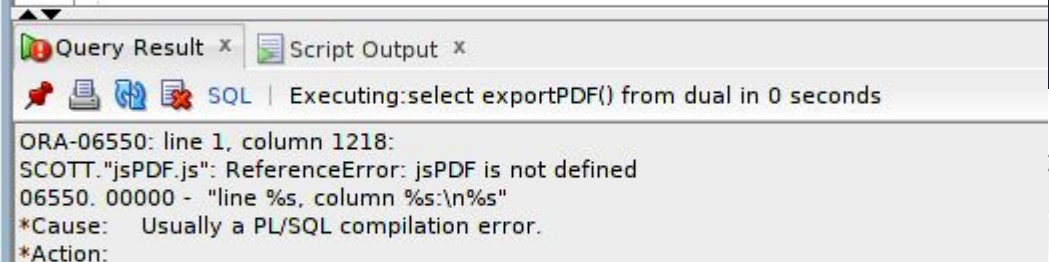
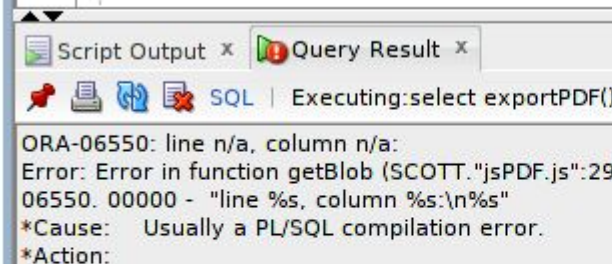
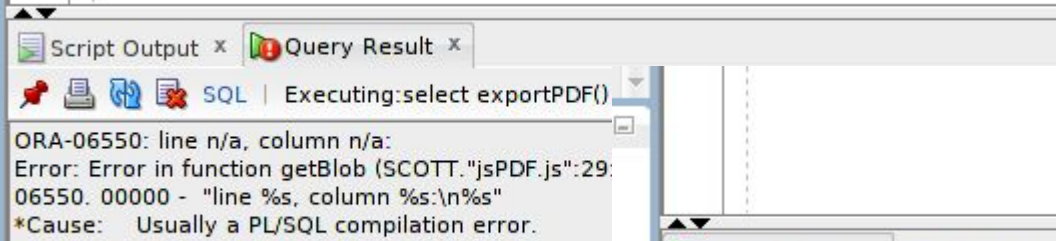
Original Goal

Generate reports directly on the Database

- JS libraries for generating PDFs available (for Node.js and Browser use)



Issues





Issues & Rescoping

- MLE behaves weird sometimes (unexpected behaviour)
- Deployment Process is Heinous



- Solution: Write tooling for deployment




Create Desired JavaScript functionality



```
module.exports.helloWorld = function (name) {  
  return "Hello " + name;  
}
```




Store Code in Database



```
INSERT INTO mylobs  
VALUES (:id, :data), {id: 1, data: data}
```




Create DB-Internal representation



```
CREATE OR REPLACE JAVASCRIPT SOURCE NAMED "mysource" USING CLOB
SELECT data
FROM mylobs
WHERE id = 1
```



Create Function



```
CREATE OR REPLACE FUNCTION helloWorld(name IN VARCHAR2)
RETURN STRING AS
LANGUAGE JAVASCRIPT NAME 'mysource\.js.helloWorld(name string) return string';
```



Call Function



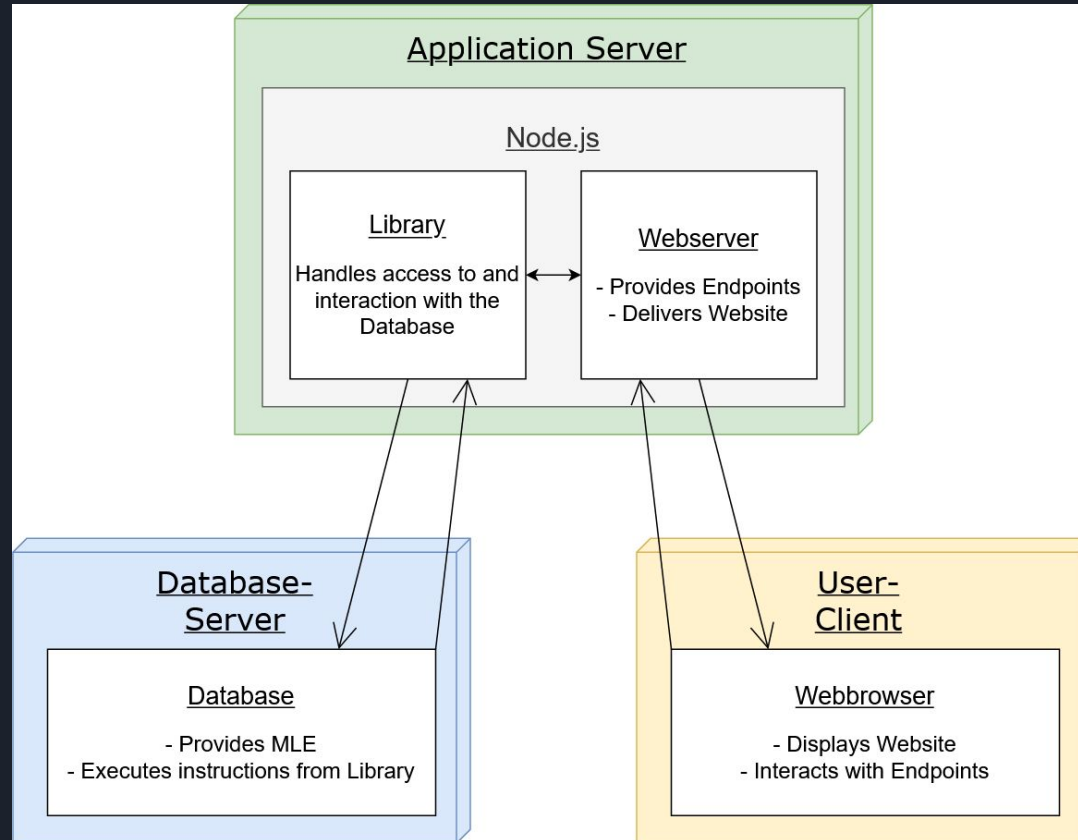
```
SELECT helloWorld('Rainer') from dual
```



Result

```
{"metaData":[{"name":"HELLOWORLD('RAINER')"}], "rows":[[{"Hello Rainer"}]}
```

Solution: Simple MLE Deployer





DEMO



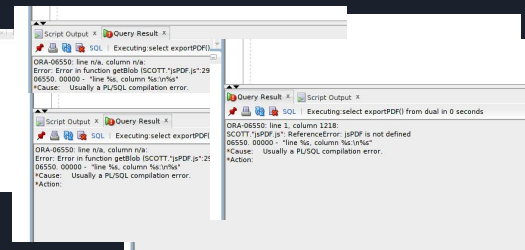
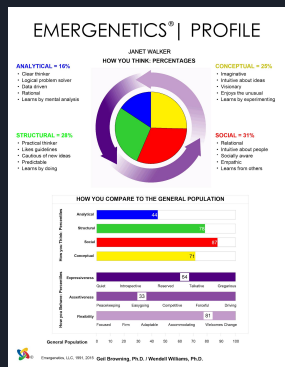
Conclusion

- Everyone with JS knowledge now can try MLE
- Wide field of future work:
 - Type conversions
 - Multiple PL/SQL functions at once
 - IDE integration
 - ...

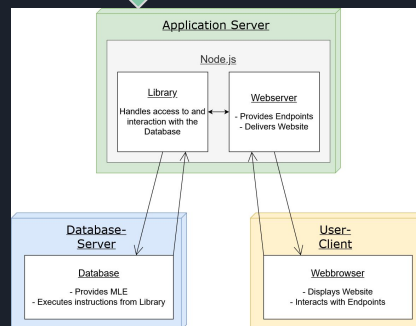
Thanks for your Attention!

Issues

Original Idea



Solution: Architecture



```
functionName
functionName without
bracket() or arguments()
mul

Arguments
a 0 STRING NUMBER
b 0 STRING NUMBER

return type
0 STRING NUMBER
0 No return value

module name
math

connection string
140.238.173.160/ORCLPDB1

database user
scott

password
*****

Deploy Run SQL
```

```
module.exports.mul = function(a, b) {
  return a * b;
};

select mul(4, 5)
from dual
```

Simple MLE Deployer



Go to simplemledeployer.uk to try it yourself!