ORACLE

JavaScript and Databases: a Perfect Match

Lucas Braun, Consulting Technical Program Manager
Noman Sheikh, Senior Member of Technical Staff

Oracle Labs Zurich

Lucas Braun

- Consulting Technical Program Manager @ Oracle Labs
- BSc, MSc and PhD in Computer Science from ETH
- Started at Oracle Labs in 2017
- Working on Oracle Database Multilingual Engine (MLE)

lucas-braun-277102153

@lucasbraun87

Noman Sheikh

- Senior Member of Technical Staff @ Oracle Labs
- MSc in Computer Science from ETH
- Started at Oracle Labs in 2021
- Working on Oracle Database Multilingual Engine (MLE)

noman-sheikh-aa797ab6

@NomanSheikh11

Agenda

1. Why JavaScript?
2. GraalVM
3. GraalVM in Oracle Database
4. Database Integration Deep Dive
5. Demo
6. Quick Glimpse: Oracle Labs Zurich

Why JavaScript?

Why JavaScript?

Popular

- Common purpose
- Widely used

Eco System

- Testing
- Deployment
- Debugging
- Monitoring

Open-Source

- 1,000,000+ libraries on NPM

Fast time to market

Why JavaScript in the Database?

Server-side development ("Smart-DB") offers many advantages

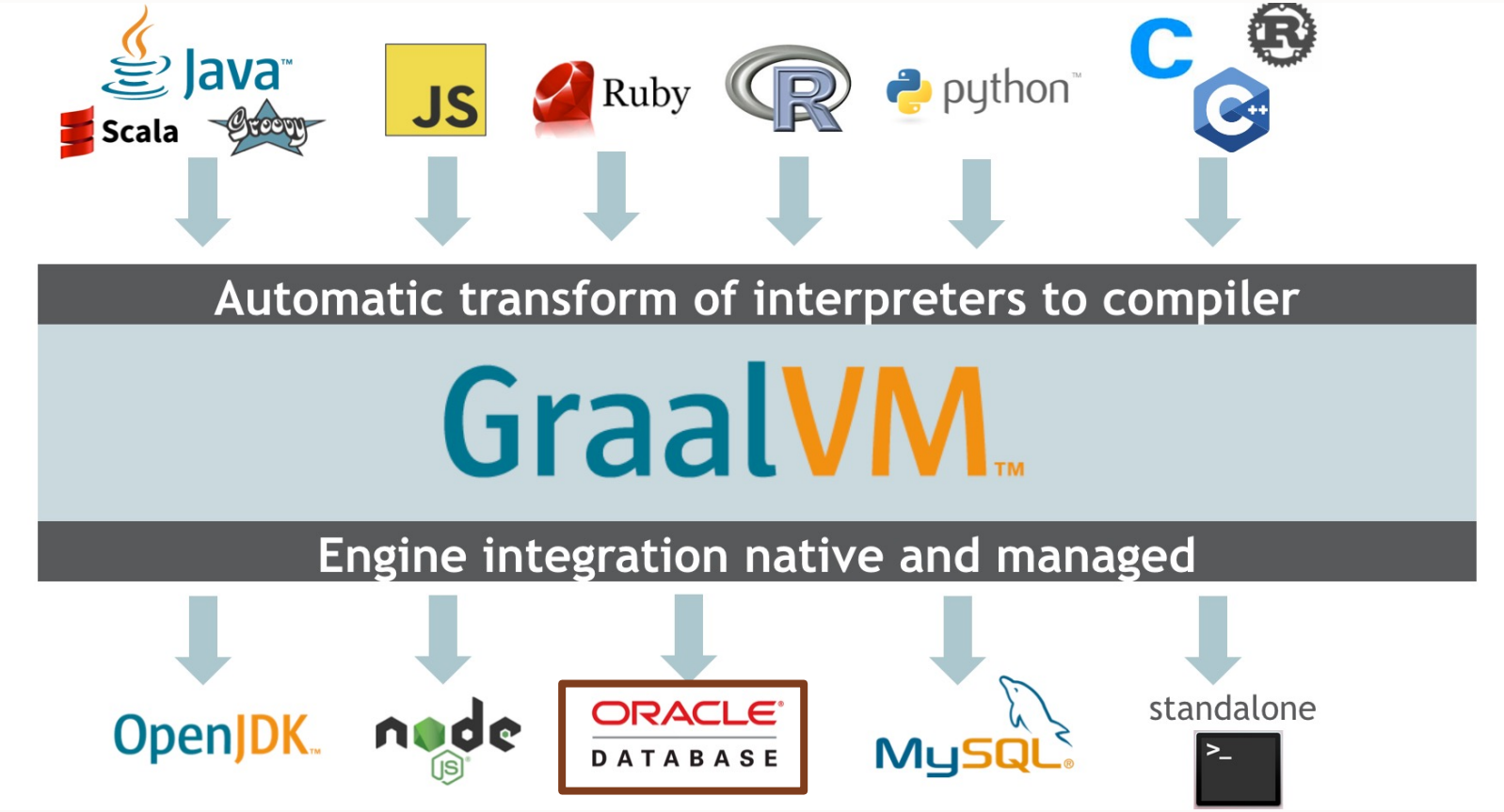
- Client systems are often constrained by lack of memory and CPU
- Processing and memory intensive operations are not completed at peak performance
- A single, central location simplifies code maintenance
- Processing data where they are stored eliminates costly network round-trips
- Adding set-based logic to JavaScript stored code should improve performance

Throughout its history Oracle has constantly improved the developer experience

- Adding JavaScript support to the core engine continues this tradition
- Enabling access to a huge ecosystem of existing module simplifies project delivery
- An additional talent pool becomes available

GraalVM

GraalVM: Multi-Lingual, Embeddable Virtual Machine



Multi-Lingual

Embeddable

GraalVM is Open-source and Free to use

GraalVM in Oracle Database

Oracle Database Multilingual Engine: GraalVM in the Database

- Started as an Oracle Labs research project some time in 2013.
- Was released on December 8, 2020 as part of the Oracle Database 21c in the Oracle Cloud
- MLE comes with many new features in **latest Oracle Database 23c**. [1]

```
create mle module validator
language javascript as

function isEmail(email, options) {
  return …;
}

…

export { isEmail }
/

create function is_email (p_name varchar2)
return boolean
as mle module validator
signature 'default.isEmail(string)';
/

select ename, email, is_Email(email) as valid from emp;
```

[1] Oracle Database 23c FREE, check it out on oracle.com/database/free

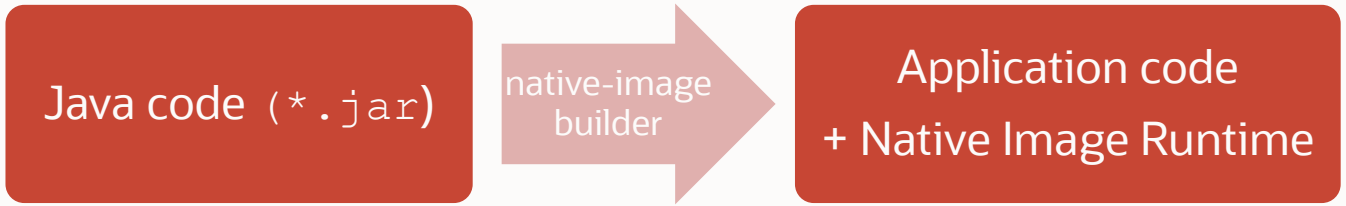
Native Image/Substrate VM: AOT Compilation & Application Snapshotting

Native Image is the key to embedding GraalVM into Oracle Database.

Initialize Once, Start Fast: Application Initialization at Build Time

CHRISTIAN WIMMER, Oracle Labs, USA
CODRUT STANCU, Oracle Labs, USA
PETER HOFER, Oracle Labs, Austria
VOJIN JOVANOVIC, Oracle Labs, Switzerland
PAUL WÖGERER, Oracle Labs, Austria
PETER B. KESSLER, Oracle Labs, USA
OLEG PLISS, Oracle Labs, USA
THOMAS WÜRTHINGER, Oracle Labs, Switzerland

AOT Compilation: compile Java code to native executable
Application Snapshotting: Perform initialization at build time

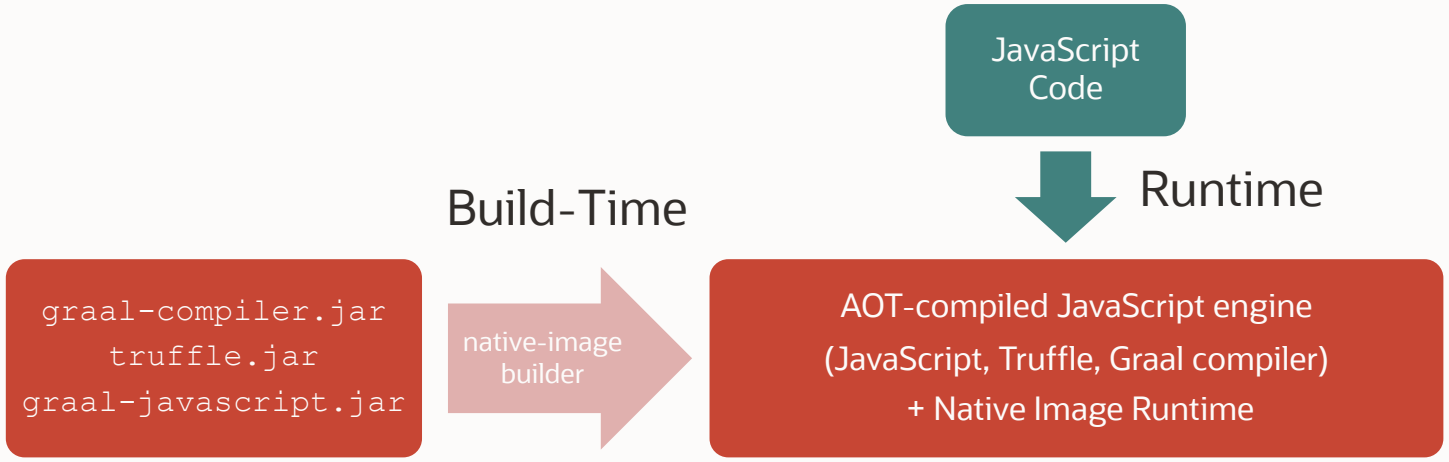


Native image: executable native code
No JVM required at run time

Native image at runtime: fast startup, low memory footprint (compared to JVM)
Embeddable: shared library can be loaded in native applications

Native Image and Truffle Languages

How can JavaScript code be executed by a Native Image?

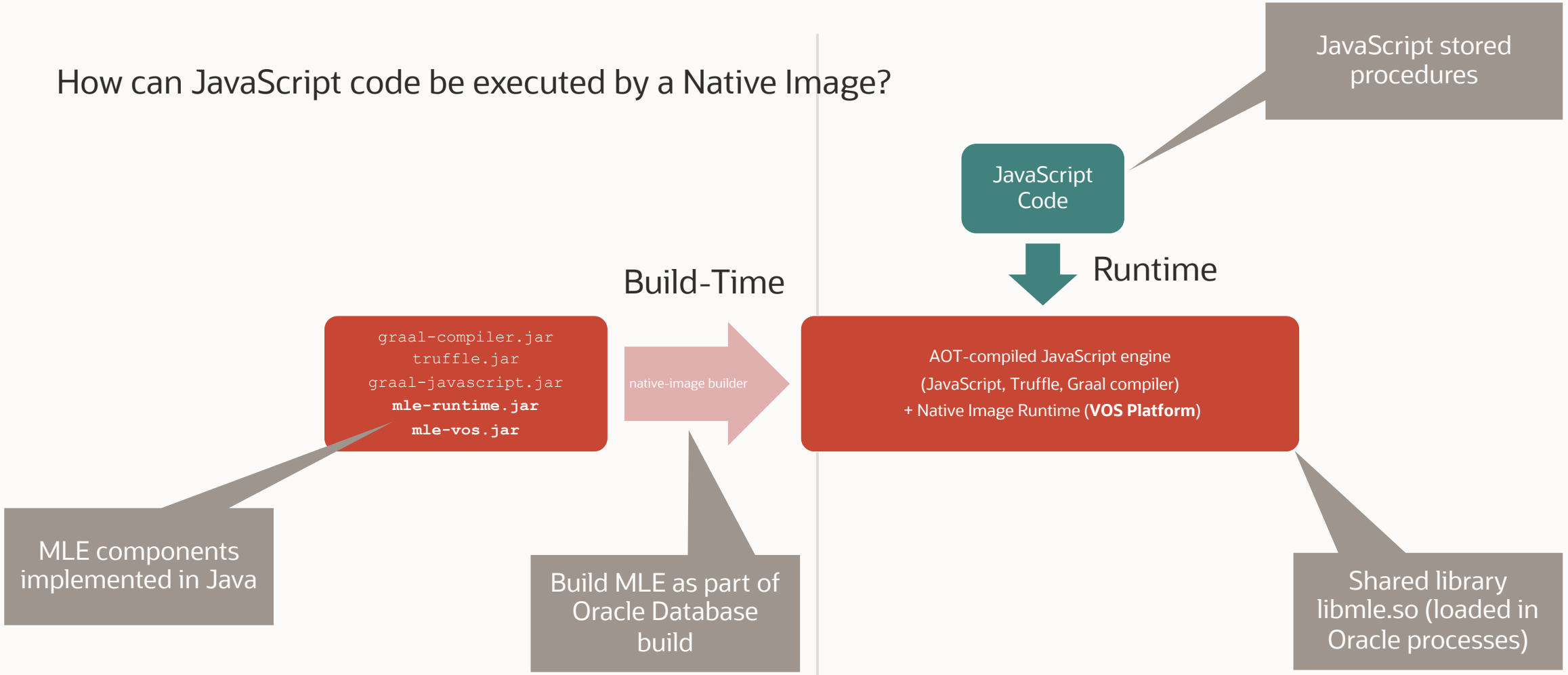


Host code is AOT-compiled and runs as machine code (no JVM).

Guest code (JavaScript code) is interpreted and JIT-compiled at runtime.

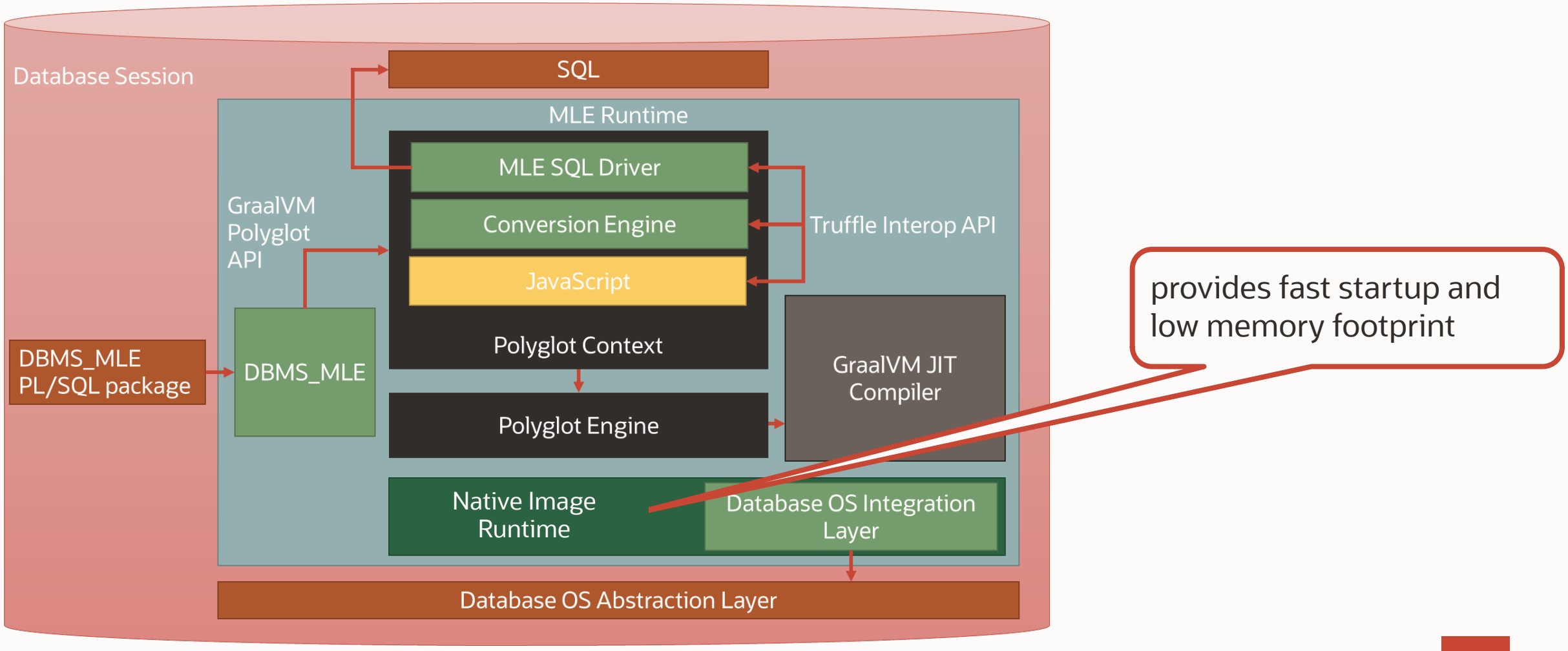
Native Image and Truffle Languages in the Database

How can JavaScript code be executed by a Native Image?



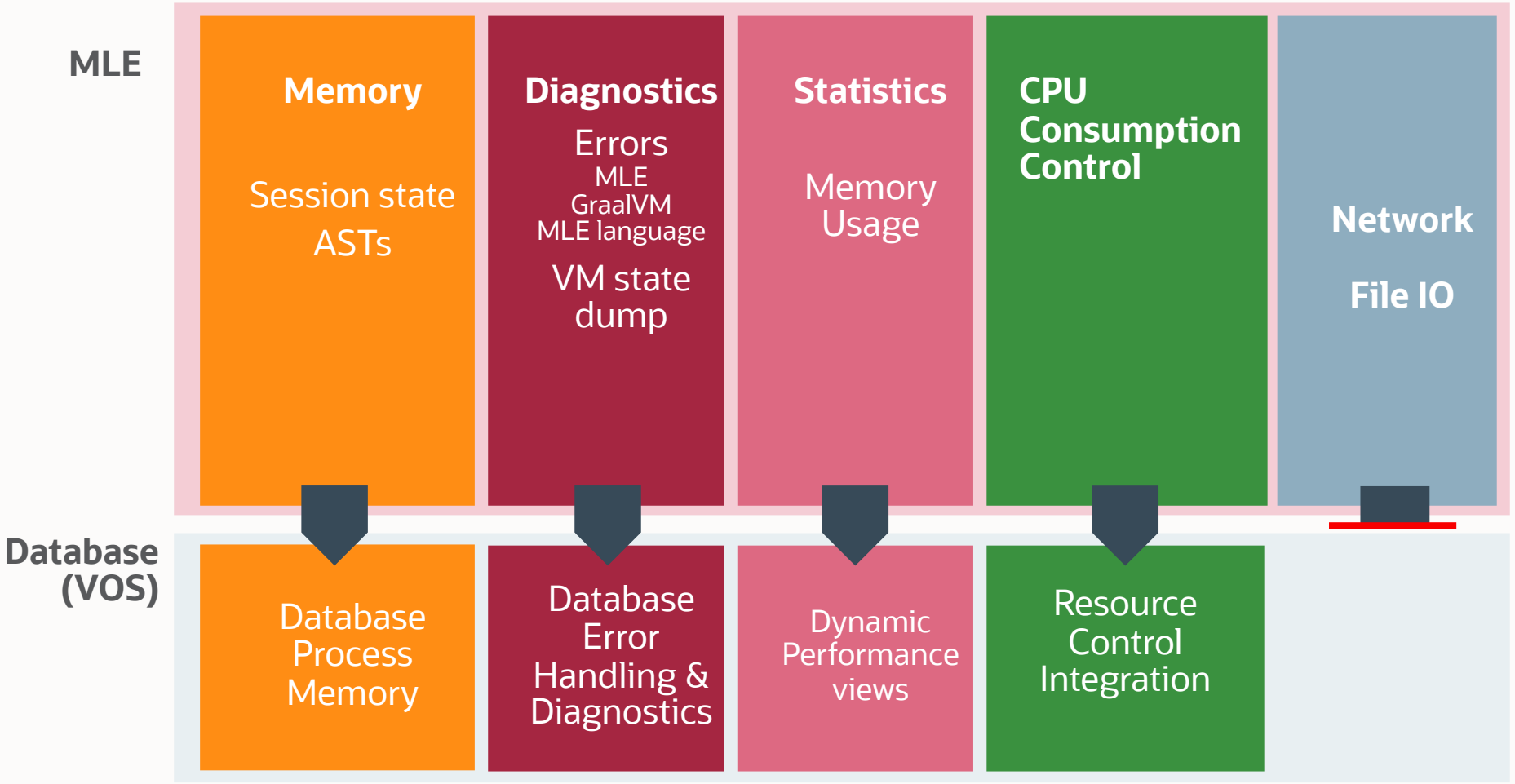
MLE Architecture

MLE runs a part of Oracle Database Server Process.



Database Integration Deep Dive

Integration with the Database Virtual OS & Resource Manager



Database Integration: Memory & Threading

Memory

- GraalVM needs to allocate memory for
  - Running user application code i.e. JavaScript code.
  - Storing ASTs of compiled code.
  - Allocations done by user code.
- MLE integrates with the memory management of Oracle Database.

Threading

- JavaScript code is just-in-time (JIT) compiled for better performance characteristics.
- MLE JIT compiles JavaScript code in a background thread.
- Oracle Database offers an internal framework for task-based parallel execution.
- MLE leverages this framework to execute compilation tasks asynchronously while the JS code in the DB session keeps running.

Database Integration: Resource Management

- What do you do?
  - If user stop execution by pressing Ctrl+C on SQL console
  - When user code starts consuming too much memory
  - When user session exhaust its share of CPU

- MLE implements regular callback mechanisms
  - Periodically checks for interrupts
  - Communicates with Oracle Database Resource Manager
  - Dynamically controls memory footprint by implementing its own Garbage Collection policy.

- Callbacks implemented using Java safe points
  - Inserted by compiler when building MLE.

Database Integration: Errors and Diagnostics

2 way integration

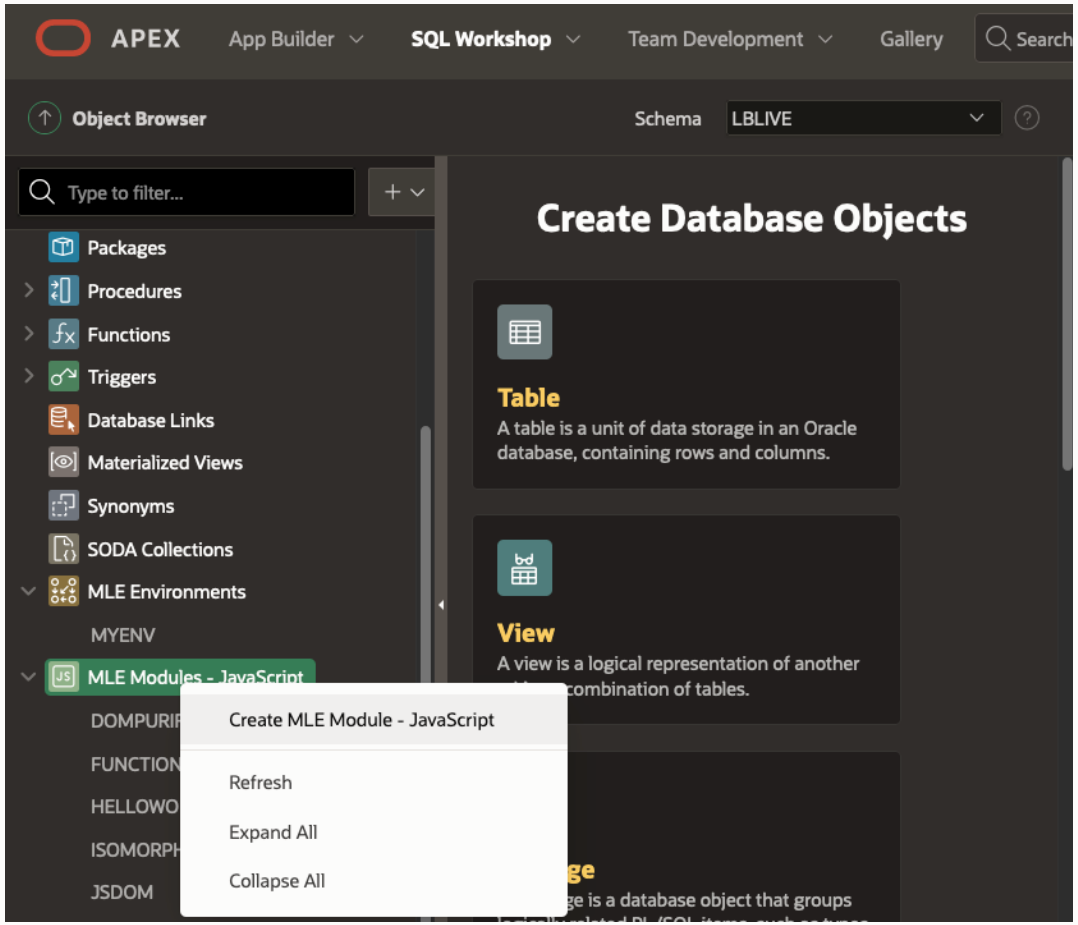
- Database → Java: When there is an error on Database calls
  - Out of Memory
  - Permission Errors
- Java → Database: When a Java exception is “escaping”
  - Java Exceptions that are not caught.
  - Escaping Java Exception are caught at top level.
  - Converted into Database error messages.

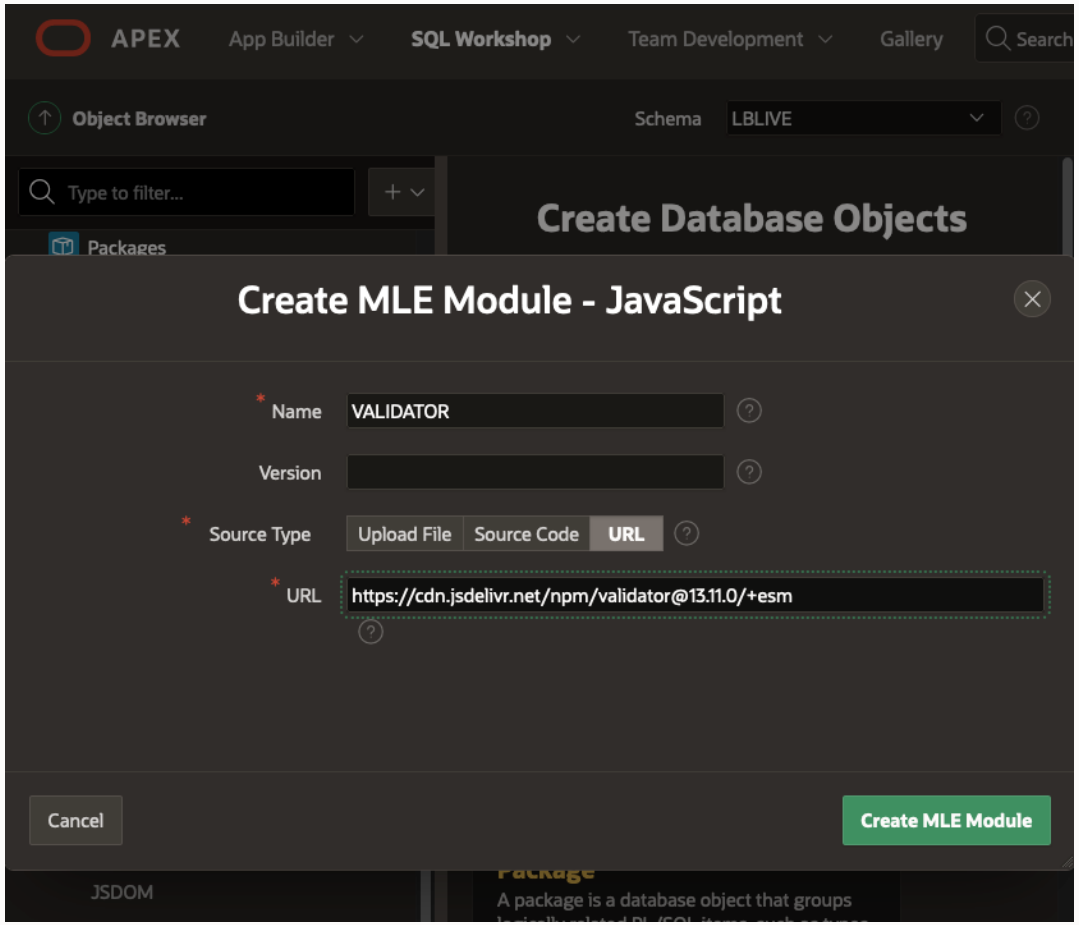
Reporting and Diagnostics

- MLE integrates with Oracle Database incident reporting framework.
- This allows Oracle DB incident reporting framework to dump relevant information about internal errors and exceptions.

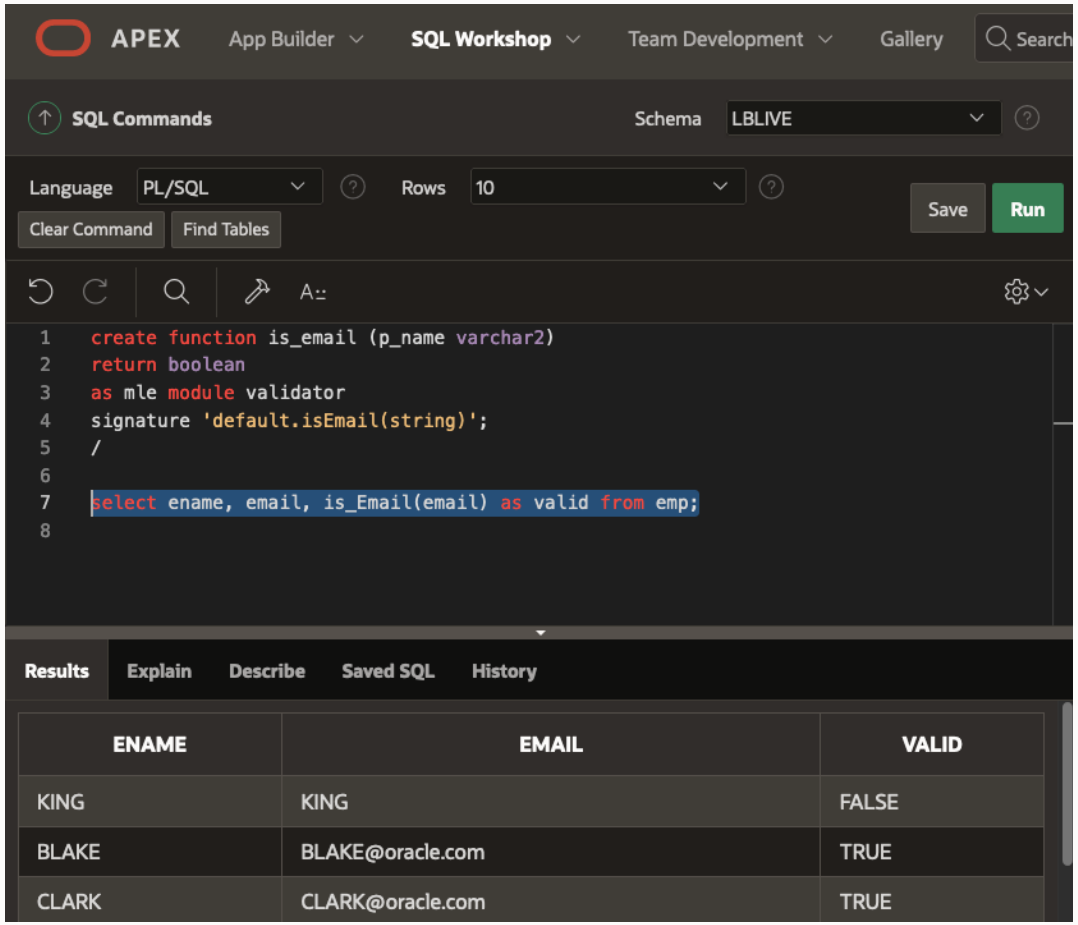
Demo

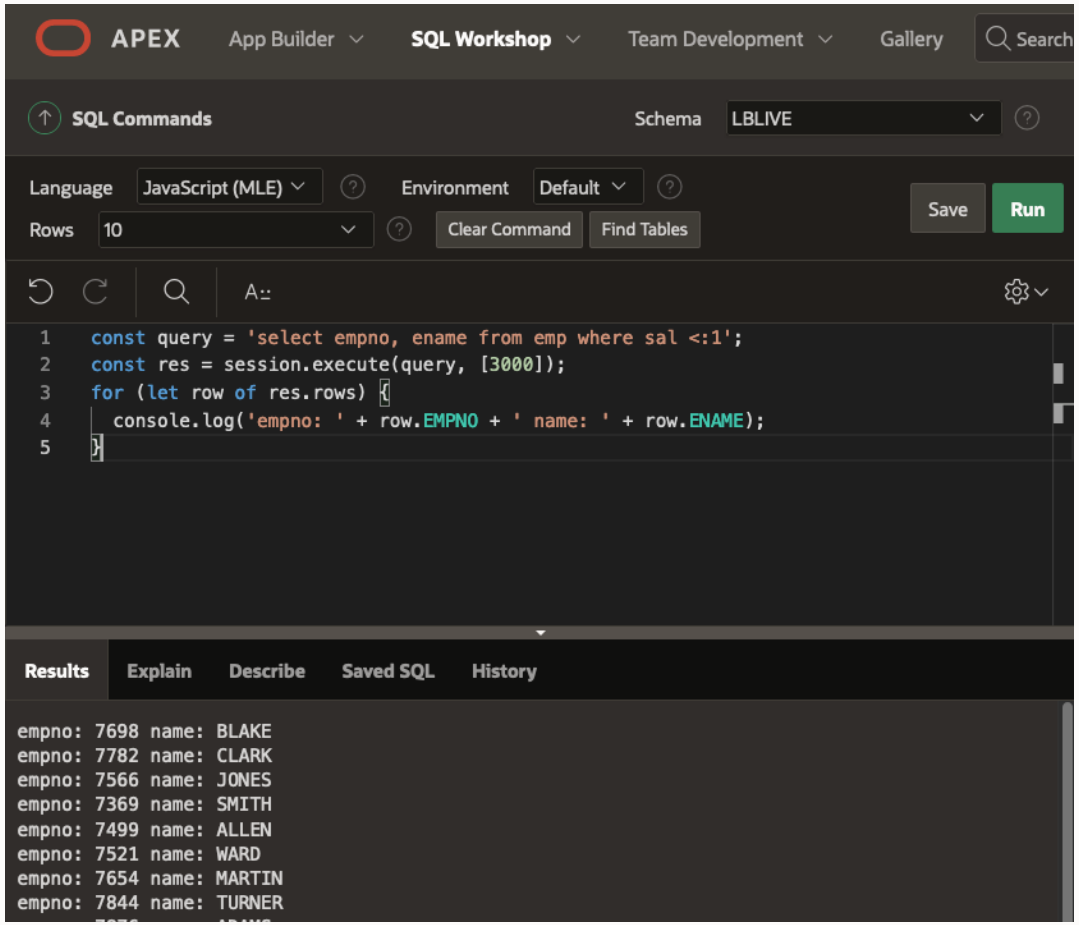
Adding a JavaScript module to Oracle 23c





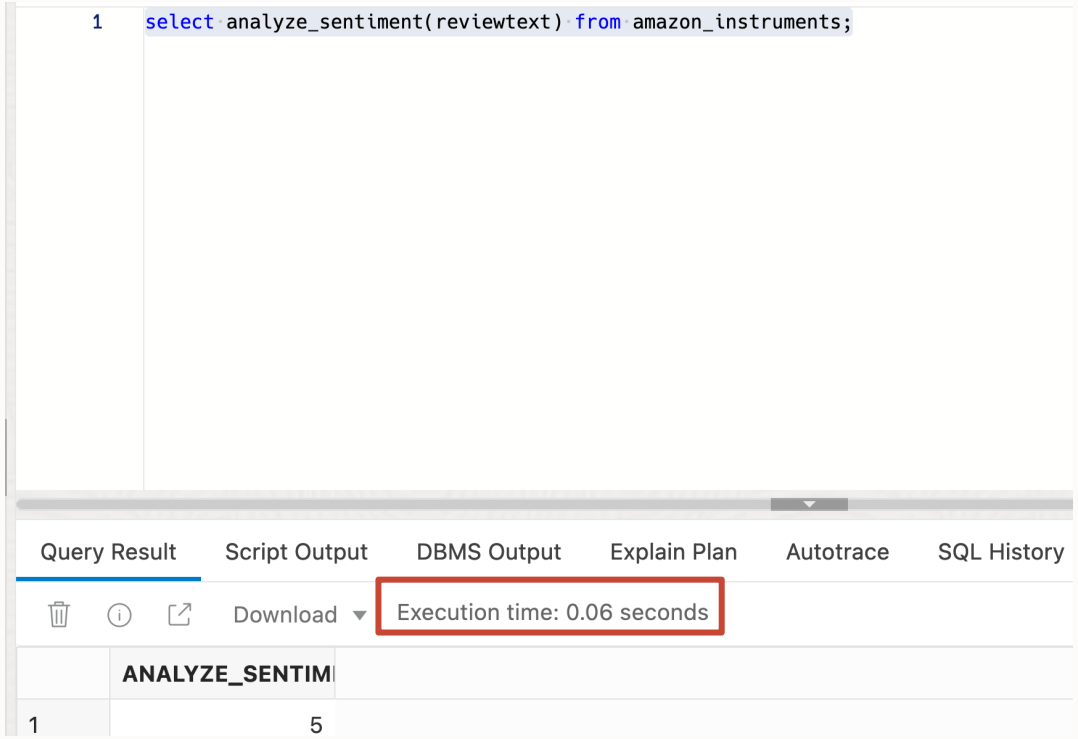
JavaScript in Oracle 23c: JavaScript from SQL and SQL from JavaScript





In-Database JavaScript can be much faster than NodeJS in mid Tier

Sentiment Analysis of 10,000 Amazon reviews about musical instruments [1]



[1] cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews

Quick Glimpse: Oracle Labs Zurich

Oracle Labs' Four Approaches to a Balanced Research Portfolio

Exploratory Research

- Pursue new ideas within domains relevant to Oracle

Directed Research

- In collaboration with product teams
- Difficult, future-looking problems
- Driven by product requirements

Consulting

- Provide unique expertise
- Small engagement across product organizations

Product Incubation

- Grow new products from Oracle Labs research

A global research team

Hundreds of researchers worldwide

Zurich: The biggest location with two floors at the Prime Tower (and growing!)

The geographic spread allows Oracle Labs to take advantage of a **tremendous pool of scientific and engineering talent** and enables Labs researchers to **collaborate with colleagues** from a **wide range of industries and universities**.

Oracle Labs locations

- Zurich, Switzerland
- Prague & Brno, Czech Republic
- Casablanca, Morocco
- Linz, Austria
- Redwood Shores, California, USA
- Austin, Texas, USA
- Belgrade, Serbia
- Brisbane, Australia
- … and more!

Selection of projects with involvement of the Zurich Lab

- **Parallel Graph AnalytiX (PGX)** – High-performance graph toolkit (single machine, distributed, in DB)
- **Data Studio (DS)** – Notebook technology for visualizing graphs and more
- **GraalVM** – A universal, polyglot VM environment
- **MultiLingual Engine (MLE)** – Bringing modern languages into the Oracle DB
- **Oracle Labs Apps (Apps)** – designing, building and operating apps using the principles of modern app development
- **Application Dependency Management (ADM) and App Platform** – develop an app platform built around Oracle's opinionated view of how cloud-native apps should be architected

Several other topics across the other offices

- (Generative) ML / AI applications, code analysis and security, concurrent programming, …

If you are interested in Computer or Data Science, we have a great topic for you!

Check them out on labs.oracle.com/pls/apex/labs/r/labs/internships.

What former interns say…

I initially joined Oracle Labs for a short internship where I was working on a distributed graph processing engine. I **designed and implemented major components** for the system in collaboration with well established Oracle Labs members and got the opportunity to **learn from very skilled people**. While I did enjoy the task, the highlight for me were the people. They were very **welcoming and helpful from the beginning** to the end of the internship. I ended up extending my internship and accepting a full time offer afterwards.

Irfan Bunjaku
ETH student, 6-month intern + MSc thesis with Oracle Labs in 2022
Today, regular team member of the MLE team at Oracle Labs

Further Information

[MLE/JavaScript Live Lab]

Try it out, play around with it.

[MLE Architecture Blog]

More details about the architecture we just presented

[Oracle Database JavaScript Developers Guide]

Plenty of nice examples ☺

[Oracle Labs Homepage]

Find out more about Oracle Labs and its internship program.

Further Information

MLE/JavaScript Live Lab
Try it out, play around with it.

MLE Architecture Blog
More details about the architecture we just presented

Oracle Database JavaScript Developers Guide
Plenty of nice examples ☺

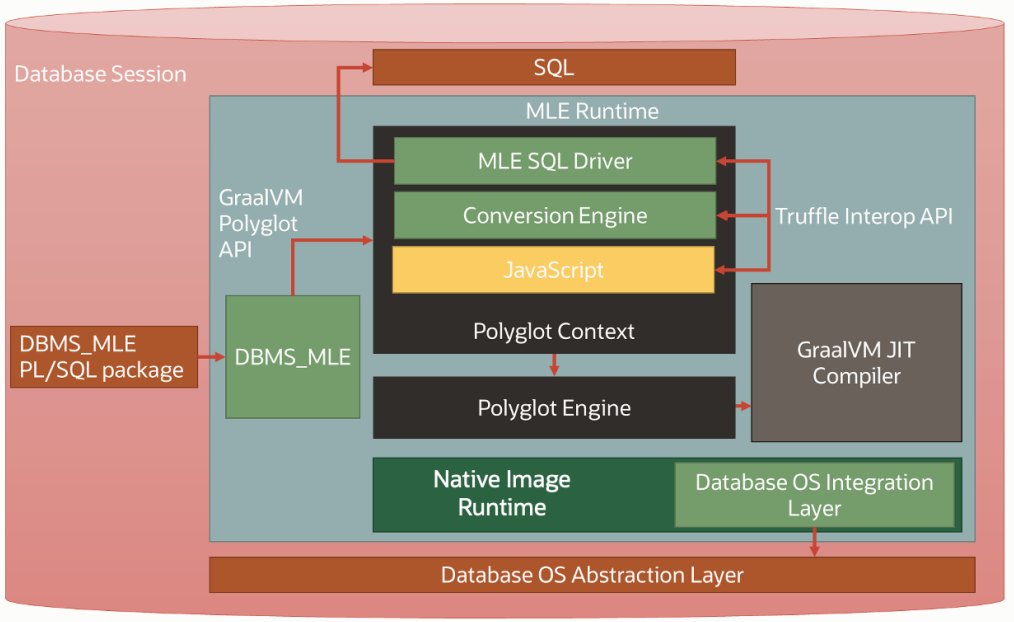
Oracle Labs Homepage
Find out more about Oracle Labs and its internship program.

Wanna know more?

lucas.braun@oracle.com
noman.ahmed.sheikh@oracle.com

MLE Architecture

MLE runs a part of Oracle Database Server Process.




ORACLE