

Transactional Dictionary in MySQL 8.0: An Internal Server Component That Matters

Dmitry Lenev, Senior Software Developer, Oracle
Amsterdam, Netherlands | October 3 – 5, 2016



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- What is a Data Dictionary?
- Data Dictionary before MySQL 8.0
- Transactional Data Dictionary in MySQL 8.0
- What's in it for you?

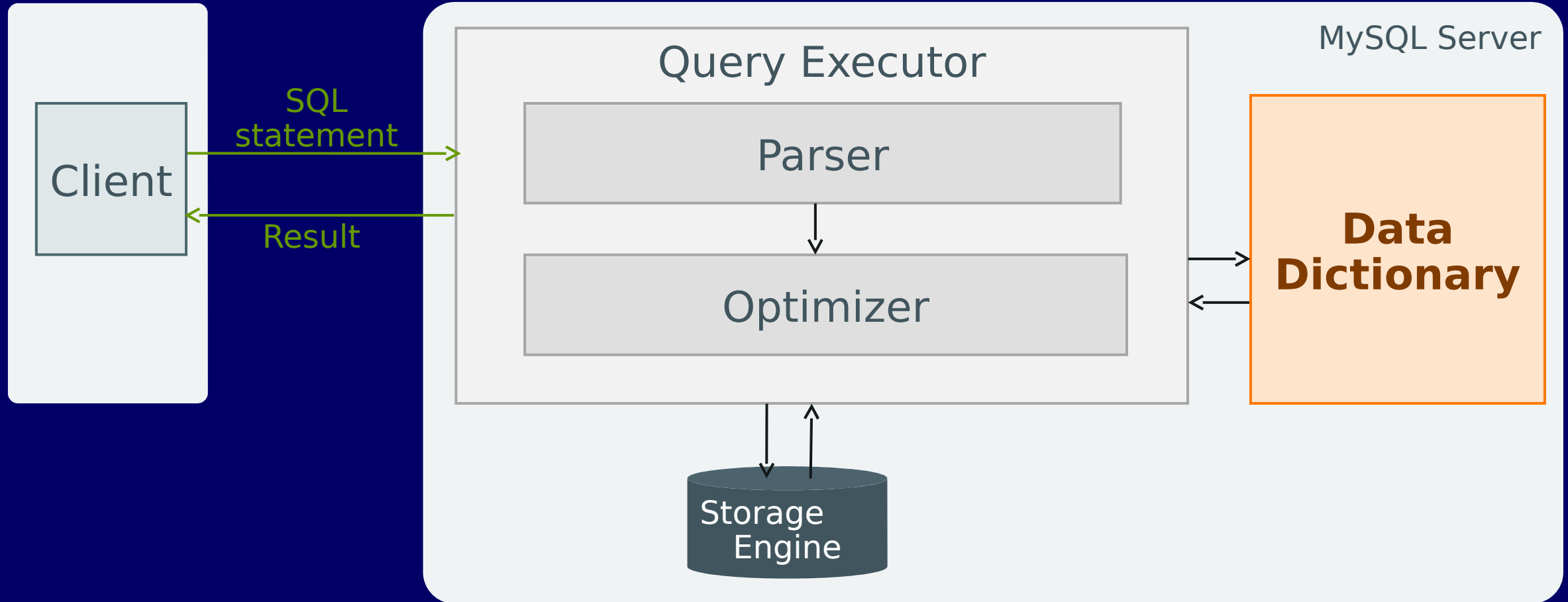
What is a Data Dictionary?

Data Dictionary

- **Metadata** is information about data in an RDBMS
 - Column Definitions, Index Definitions, Foreign Key Definitions,...
- **Data Dictionary** is a collection of metadata for all data in an RDBMS

Metadata →	ID	NAME	WEIGHT	HEIGHT	GENDER
Data →	3	Bob	80	185	M
	5	Liz	55	165	F

Data Dictionary: Role in Server



Data Dictionary: Types of Metadata

```
CREATE TABLE customers(  
  id INT AUTO_INCREMENT  
  ...  
  PRIMARY KEY (id),  
  INDEX ...  
  FOREIGN KEY ...  
)
```

```
CREATE PROCEDURE p1(v INT)  
SQL SECURITY INVOKER  
BEGIN  
  ...  
END
```

Data Dictionary

Table Definitions

SP Definitions

ACL

View Definitions

Schemas

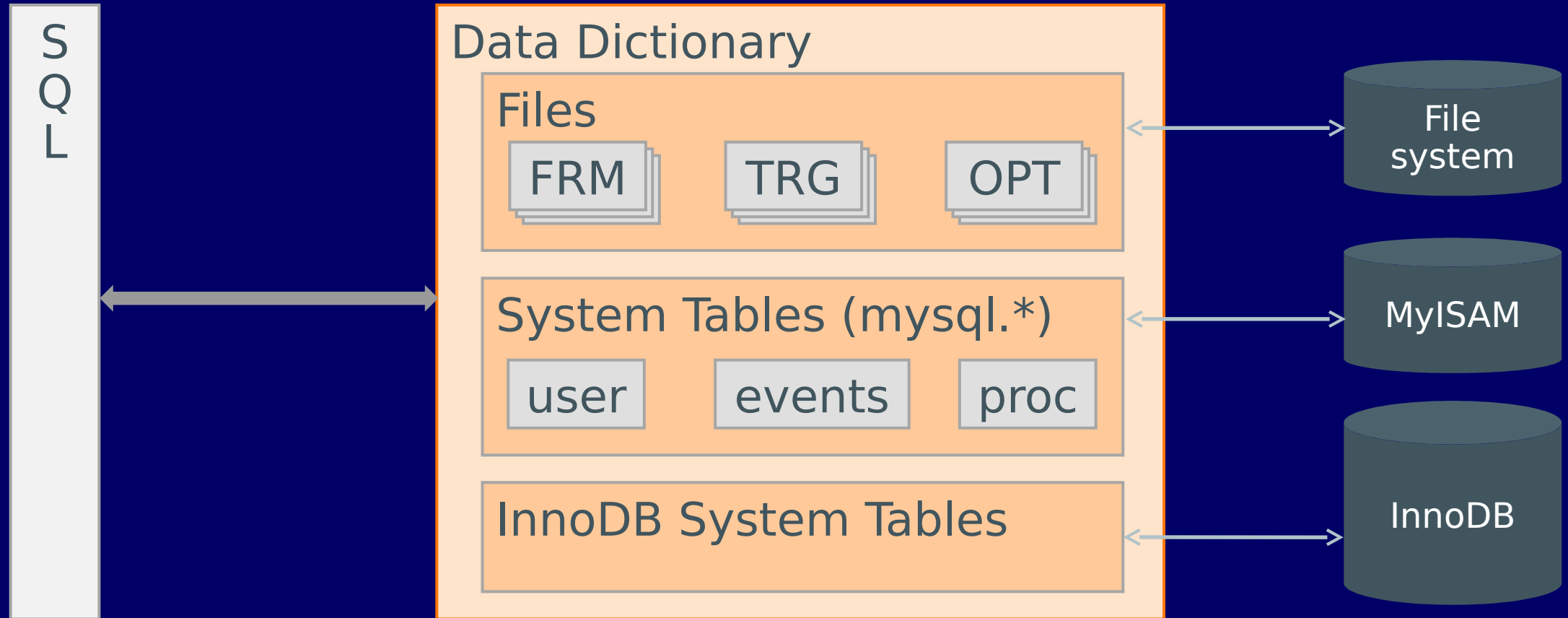
Data Dictionary before MySQL 8.0

Data Dictionary before MySQL 8.0

Metadata is stored in mix of files and tables:

- File based:
 - FRM, TRN, TRG, OPT, PAR
- Table based: non-transactional
 - mysql.proc, ...
- Table based: transactional
 - innodb.SYS*

Data Dictionary before MySQL 8.0

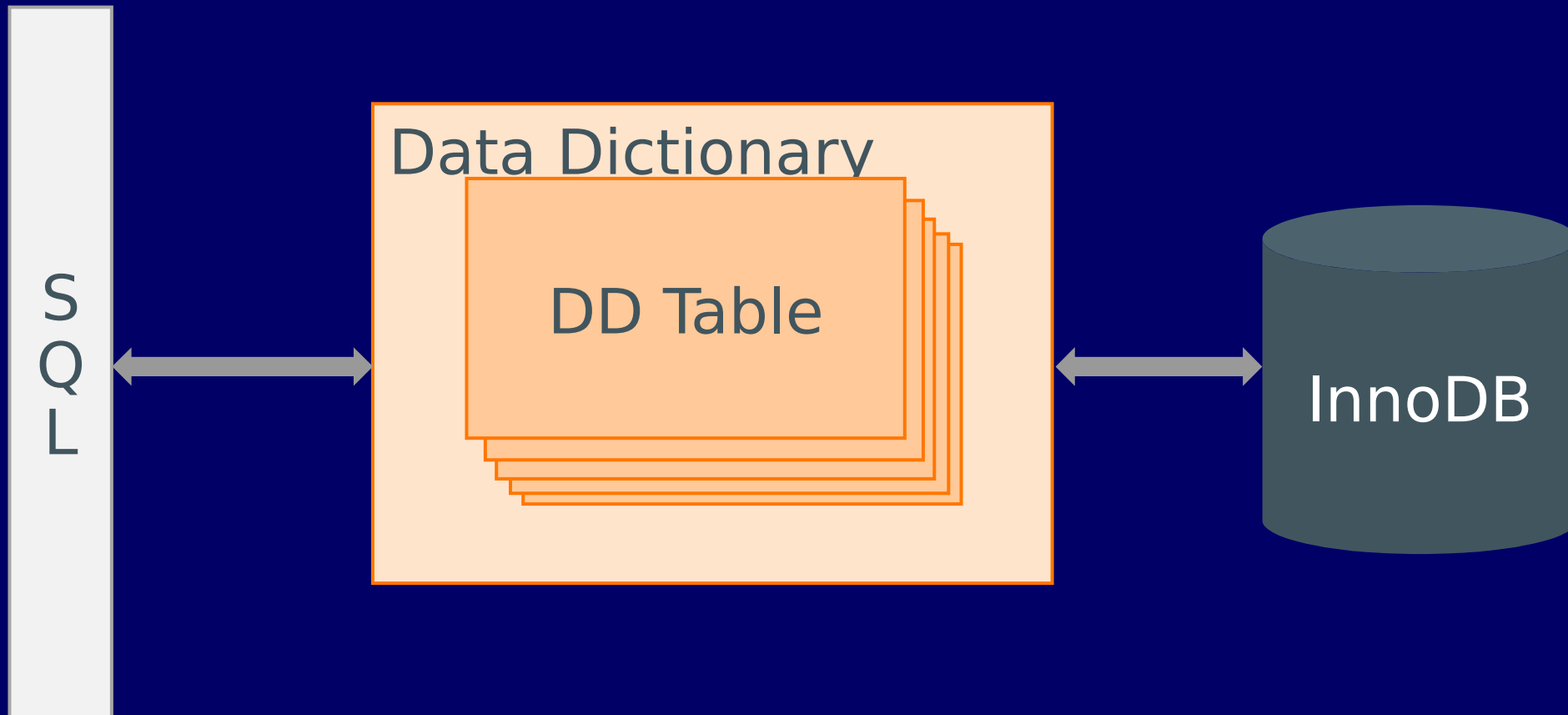


Data Dictionary before MySQL 8.0: Problems

- INFORMATION_SCHEMA queries are slow
- Inconsistencies due to non-transactional storage of metadata
- Inconsistencies between InnoDB metadata and Server metadata
- Show-stopper for crash-safe and transactional DDL
- Replication is challenging, as DDL are not atomic
- Difficult to extend
- No uniform API

Transactional Data Dictionary in MySQL 8.0

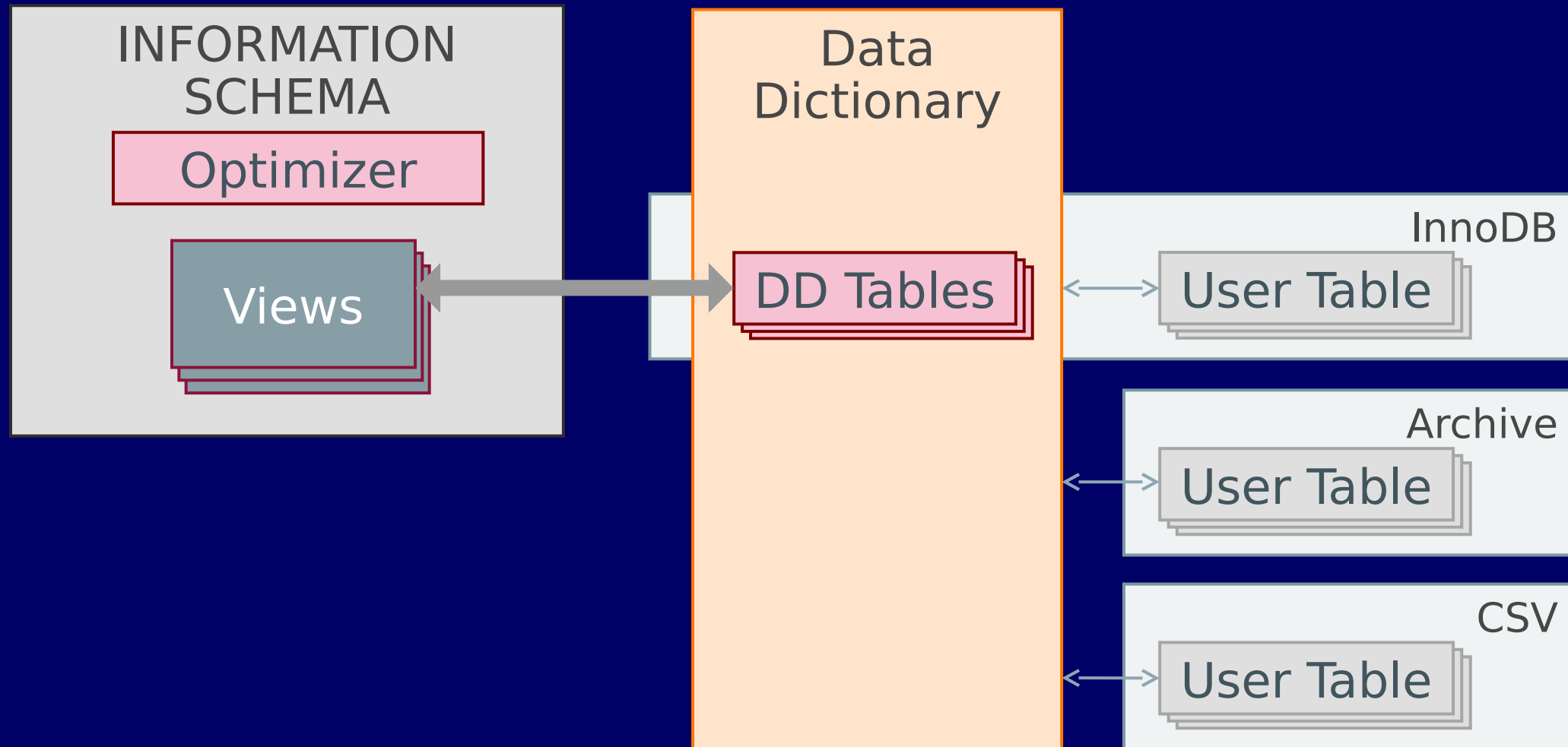
Transactional Data Dictionary in MySQL 8.0



Transactional Data Dictionary in MySQL 8.0: Main features

- All metadata stored in tables
- Single metadata repository for all MySQL Server components
- Reliable, crash-safe InnoDB tables
- INFORMATION_SCHEMA implemented as views over DD tables
 - Queries can be optimized
 - Improved performance
 - Simpler, uniform implementation, easier to maintain

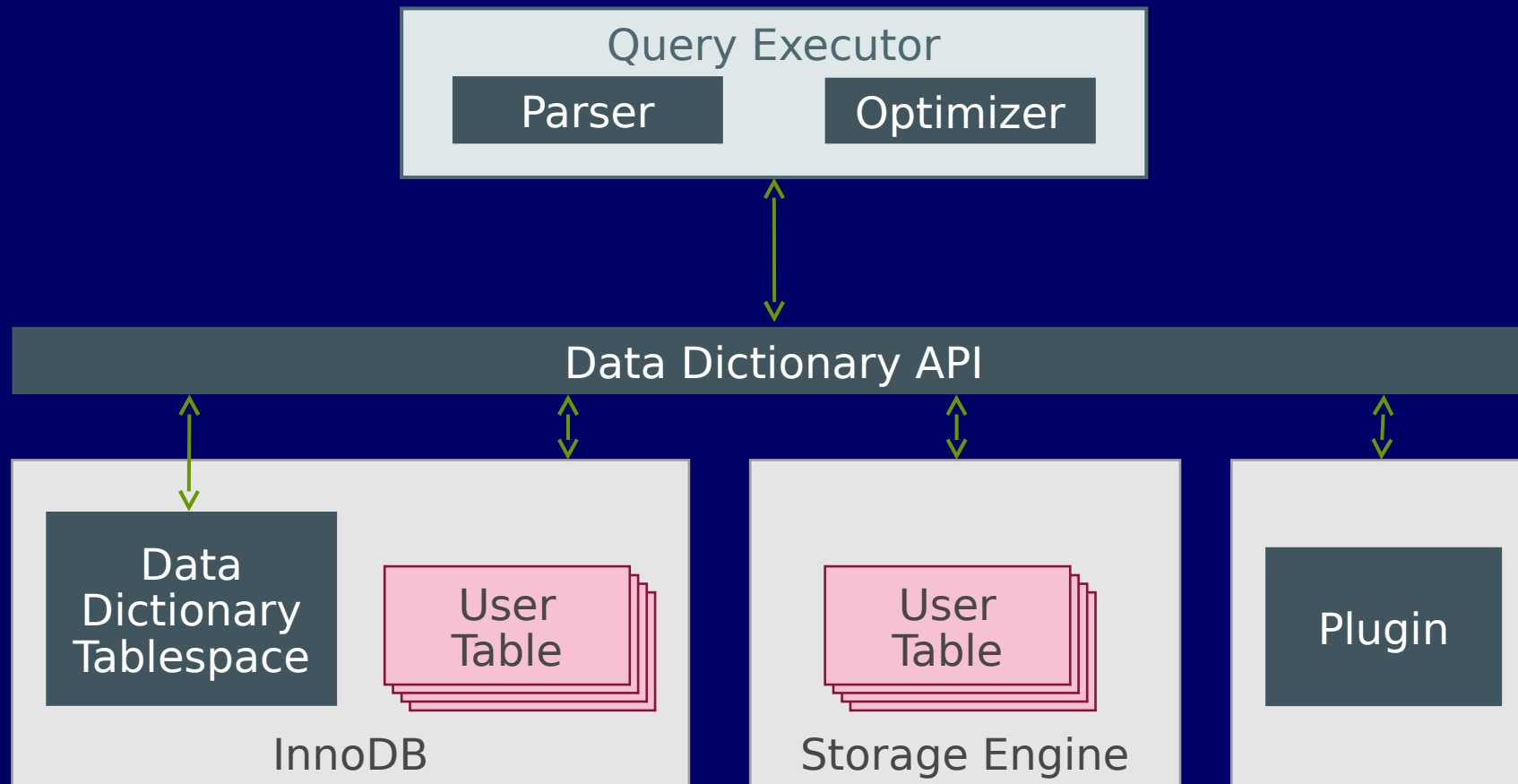
Transactional Data Dictionary in MySQL 8.0: Overview



Transactional Data Dictionary in MySQL 8.0: Main features, cont'ed

- Extendable:
 - The data dictionary schema is based on SQL standard definitions
 - The data dictionary is designed to be easily extended for new requirements
 - Designed for automated upgrade of metadata
 - Designed to allow plugins to add and extend system tables, INFORMATION_SCHEMA views, PERFORMANCE_SCHEMA tables

Transactional Data Dictionary in MySQL 8.0: Architecture



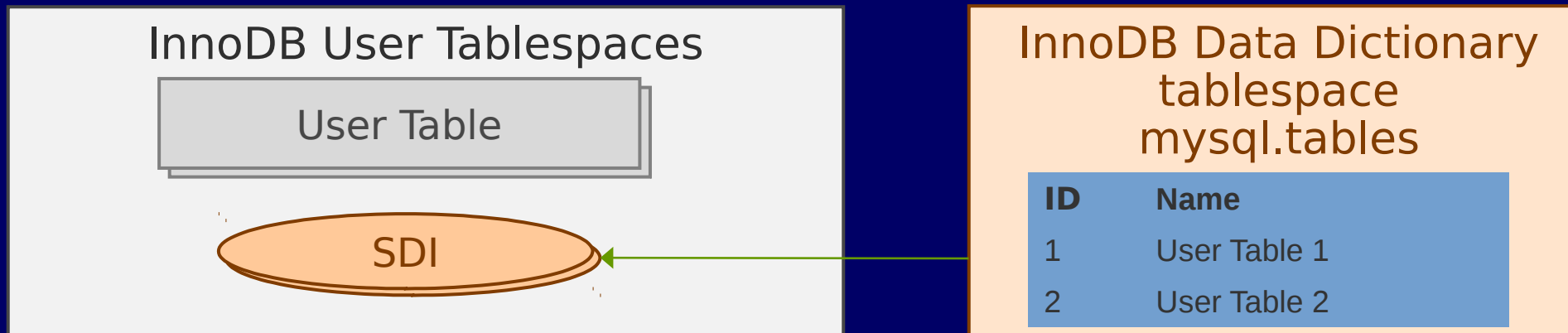
Transactional Data Dictionary in MySQL 8.0: API design goals

- A single way to deal with Data Dictionary
 - For the server core
 - For Storage Engines
 - For plugins
- A uniform API for all types of metadata
- Provide a way to handle Storage Engine private data
 - Storage Engines no longer have to store their own metadata
 - The API provides a blob to be used by SEs

Transactional Data Dictionary in MySQL 8.0: Reliability and redundancy

- SDI: Serialized Dictionary Information
 - Copy of metadata from the Data Dictionary in JSON format
 - Stored in data tablespaces (InnoDB) or .SDI files (MyISAM)
 - Used for data migration and redundancy

The InnoDB Data Dictionary tablespace is the **metadata storage**

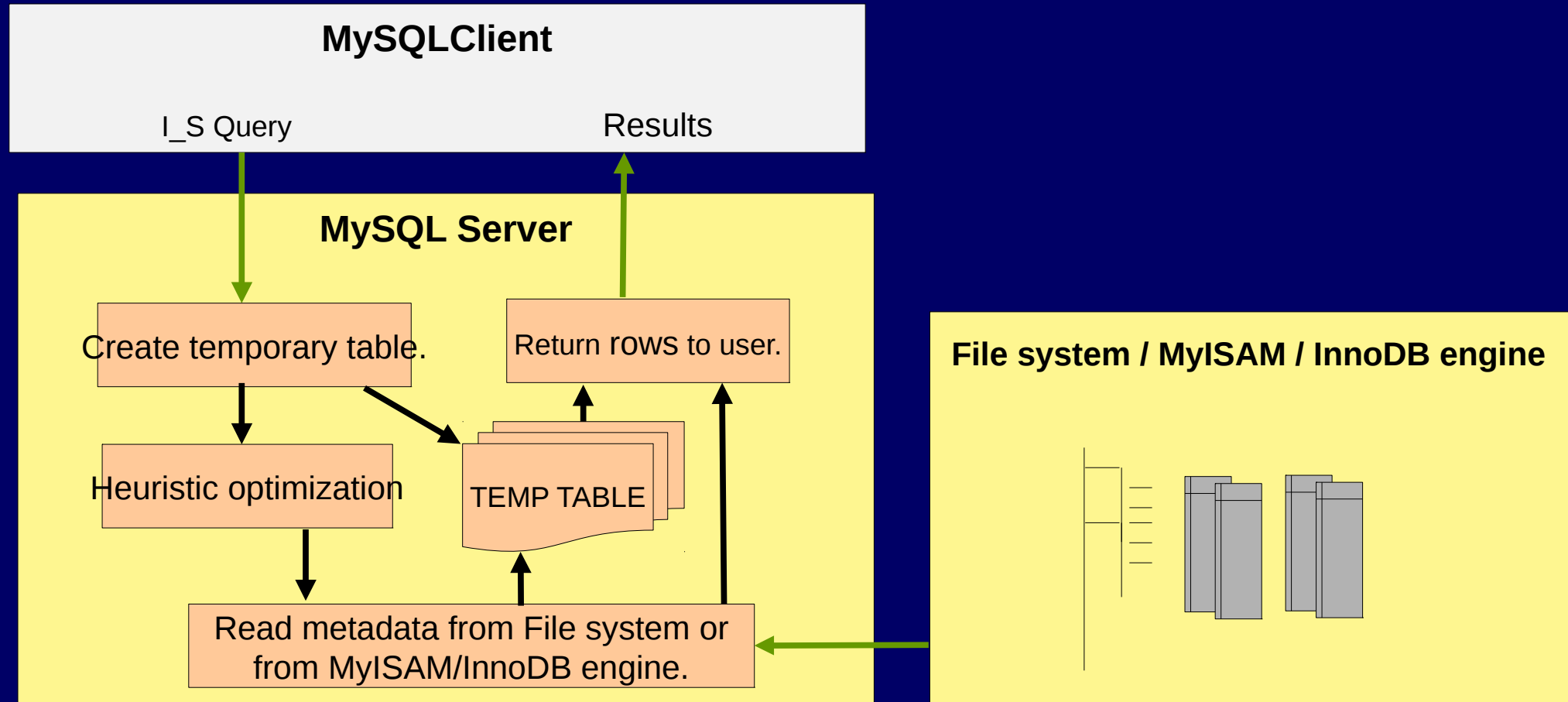


Transactional Data Dictionary in MySQL 8.0: What's in it for you?

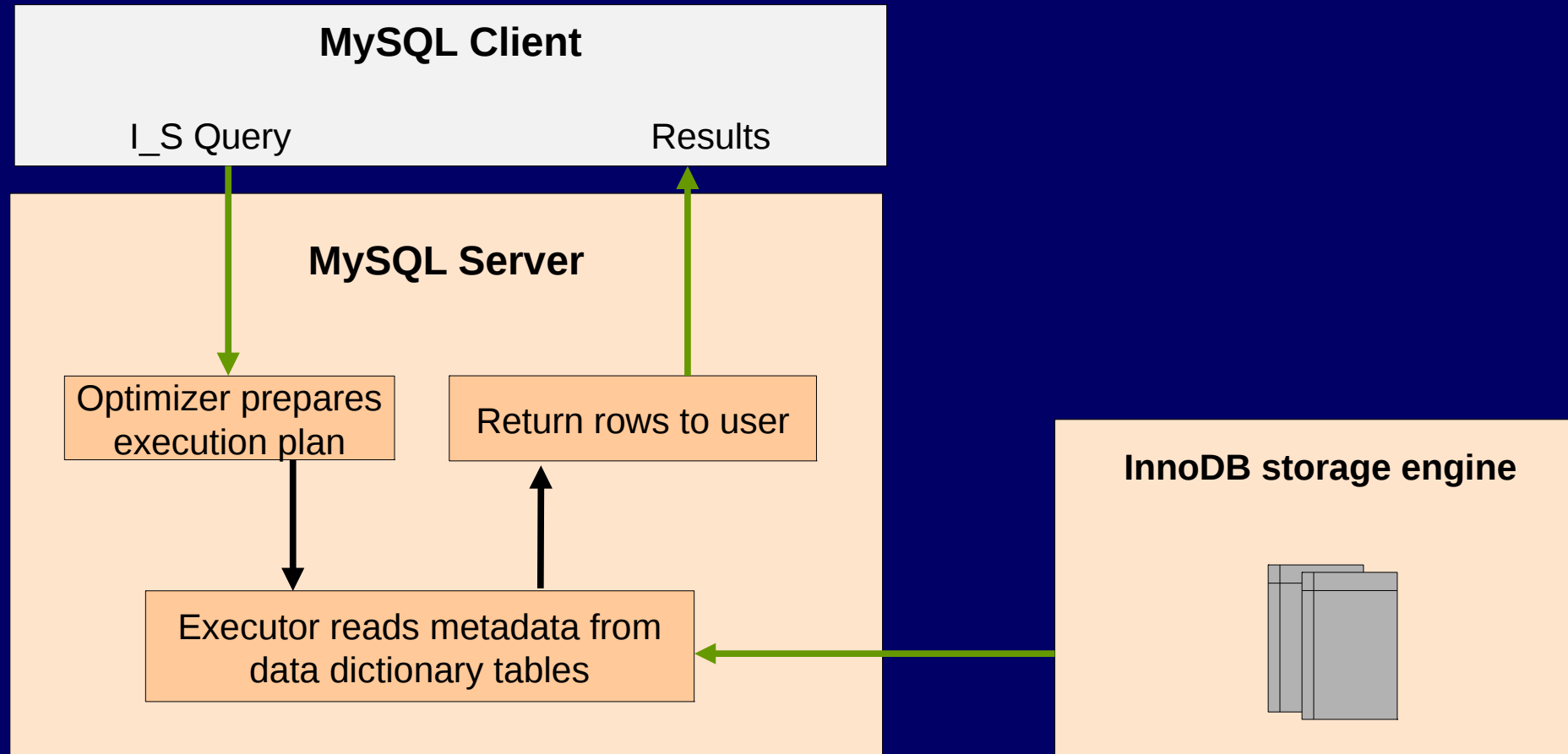
Transactional Data Dictionary Benefits

- INFORMATION_SCHEMA
 - Improved performance and scalability
 - Solves longstanding issues
- Reliability and Ease of Use
 - Atomic DDL
 - Serialized Dictionary Information
 - Automated Data Dictionary upgrade
- ...

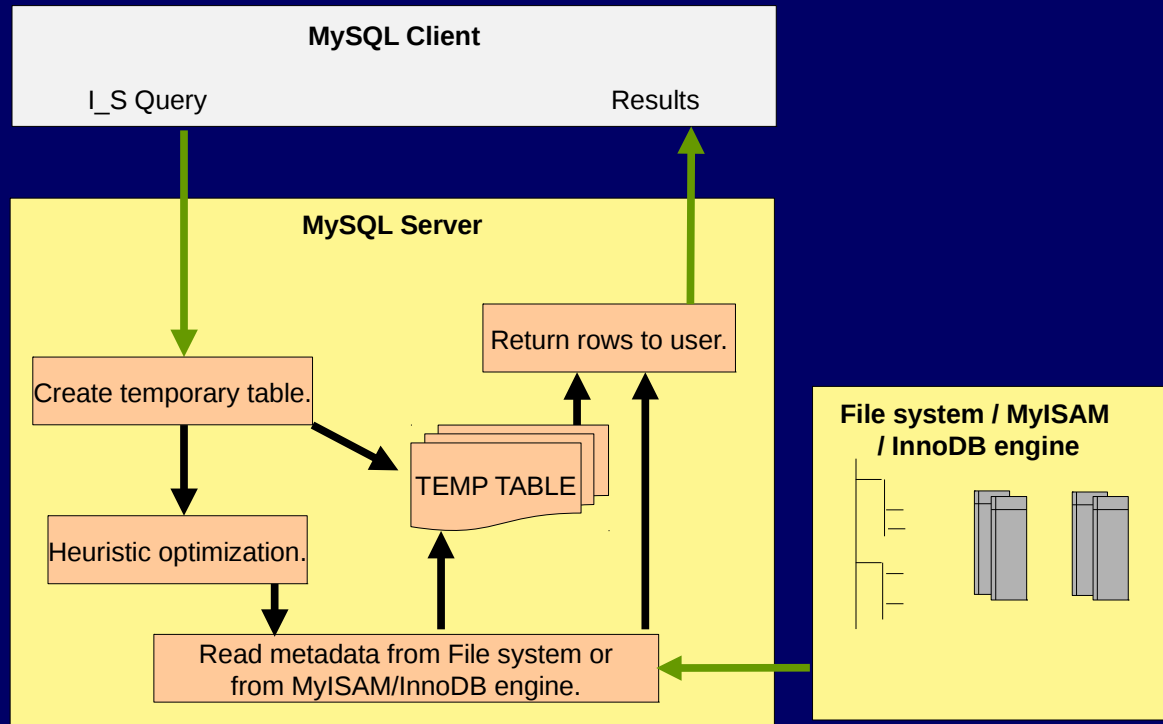
INFORMATION_SCHEMA in MySQL 5.7: Multiple sources, heuristics, creation of temp tables



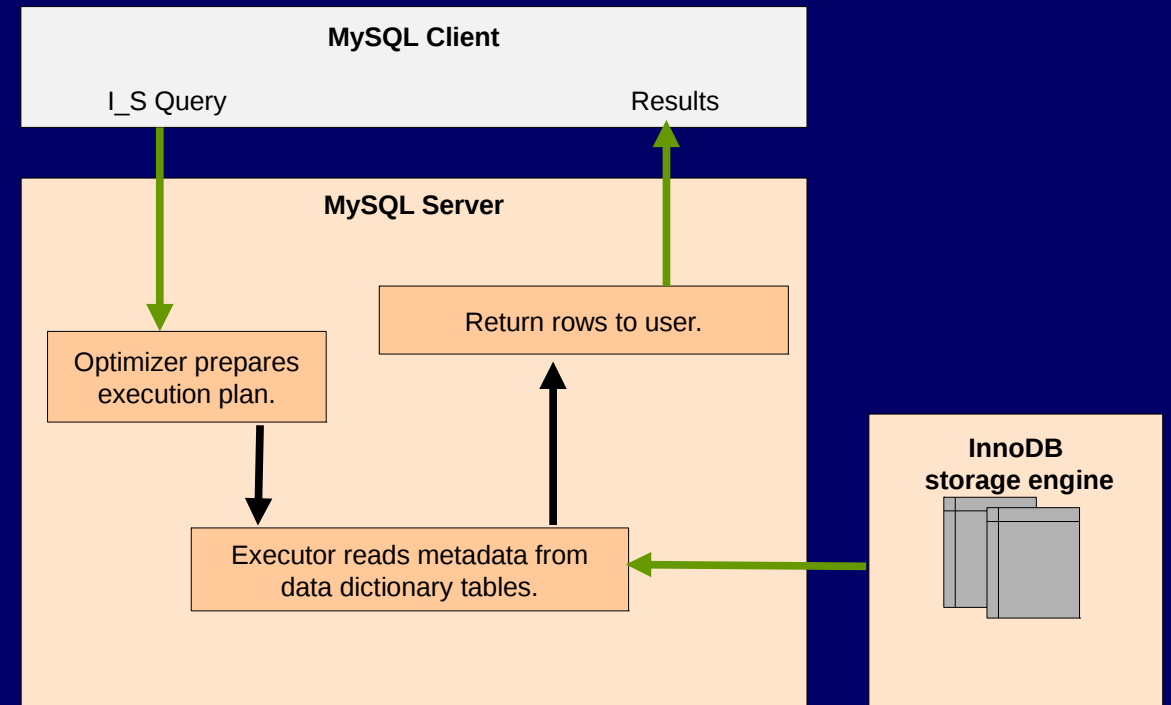
INFORMATION_SCHEMA in MySQL 8.0: Uniform, simple and using server standard features



INFORMATION_SCHEMA Performance and Scalability: Uniform, simpler implementation makes it a lot faster



INFORMATION_SCHEMA in 5.7



INFORMATION_SCHEMA in 8.0

INFORMATION_SCHEMA Performance and Scalability: Some results

I_S queries scale, both with database size and query load:

- Typically 30X performance improvements over MySQL 5.7
- More than 100X for some queries like: List all InnoDB table columns

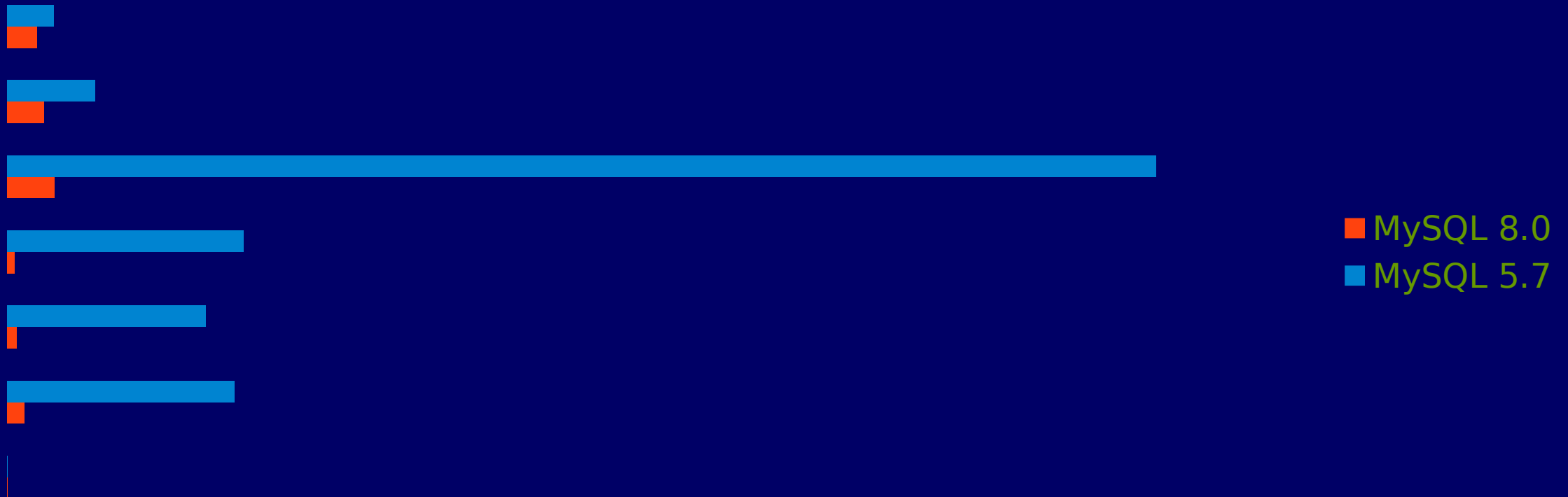


Time in Seconds (Lower is better)

INFORMATION_SCHEMA Performance and Scalability: More results

- 100 schemas times 50 tables (5000 tables)

Already faster at
7/10 queries in our
test suite!



Time in Seconds (Lower is better)

INFORMATION_SCHEMA in MySQL 8.0: Dynamic values

- TABLES.TABLE_ROWS, TABLES.DATA_FREE, ...
 - Handled by opening table and retrieving data from SE in MySQL 5.7 – expensive!!!
- Two approaches in MySQL 8.0
 - information_schema_stats=cached
 - Default
 - Stored in auxiliary tables
 - ANALYZE TABLE
 - information_schema_stats=latest
 - Native functions in the I_S views
 - Uses special SE API (InnoDB) or falls back to opening tables (MyISAM)



INFORMATION_SCHEMA in MySQL 8.0: Longstanding issues solved

- Bug#34921 “Comparison with information schema tables don't honor collation”
- Bug#48445 “Inconsistency with SHOW and SELECT FROM I_S”
- Bug#61846 “SHOW FULL COLUMNS displays incorrect privileges for table”
- Bug#65121 “Inconsistent result for select on INFORMATION_SCHEMA.STATISTICS”
- Bug#75532 “Join between I_S schema tables is case insensitive/returns wrong value”
- Bug#81347 “Unnecessary scanned all databases for information schema”
- ...



Reliability – Atomic DDL

Work in progress!

- All metadata is stored in InnoDB (transactional SE)
- DDL code changed to avoid intermediate commits:
 - At Server level
 - At InnoDB level
 - At Replication level

Reliability – Atomic DDL: Use case – DROP DATABASE

Work in progress!

MySQL 5.7

- Delete tables
 - Metadata, TRN/TRG/FRM files
 - Data, InnoDB tables
- Delete stored programs
 - Metadata, rows in MyISAM (non-transactional)
- Delete schema
 - Metadata, db.opt file

**Mix of filesystem, non-transactional/
transactional storage updates and
multiple commits**

MySQL 8.0

- Delete tables
 - Metadata rows in InnoDB
 - Data, InnoDB tables
- Delete stored programs
 - Metadata, rows in InnoDB
- Delete schema
 - Metadata, rows in InnoDB

**Updates to transactional storage,
single commit**



Reliability and Ease of Use:

Use case – disaster recovery / import (InnoDB)

- MySQL 8.0: metadata and data stored in InnoDB (transactional SE)
 - Metadata copy is stored in SDI embedded in .ibd
 - Data is stored in .ibd
- MySQL before 8.0: metadata in files, data in InnoDB
 - Metadata is stored in .FRM
 - Data stored in .ibd

Reliability and Ease of Use:

Use case – disaster recovery / import, con't ed

MySQL 5.7

- Moving .FRM and data files around (MyISAM)
- Create tables and do import using SQL (InnoDB)

```
CREATE TABLE t1 (...)
```

```
ALTER TABLE t1 DISCARD TABLESPACE
```

```
ALTER TABLE t1 IMPORT TABLESPACE
```

Easy to do mistakes!

MySQL 8.0

- Self-descriptive tablespaces (InnoDB/ embedded SDI, MyISAM needs .SDI file)
- New IMPORT statement will import the tablespace:
 - Read metadata from SDI and create tables
 - Import data into these tables

Worst case, if SDI is corrupt, you can extract it in JSON format and edit



Reliability and Ease of Use: Automated dictionary upgrade

Work in progress!

- The Data Dictionary is versioned
- The MySQL Server will support upgrading dictionary tables automatically
 - Done in executable and not in script
 - Reduces user and privilege issues for the process doing the upgrade
- The Data Dictionary will be updated atomically

Extending Data Dictionary from Plugins

Work in progress!

- API for plugins to extend INFORMATION_SCHEMA and PERFORMANCE_SCHEMA
 - Add new virtual tables/views
- API for plugins to store their specific data in the Data Dictionary



Try for yourself!

- Downloadable 8.0.0 DMR
 - dev.mysql.com
- Enjoy and give us your feedback!

Thank you!
