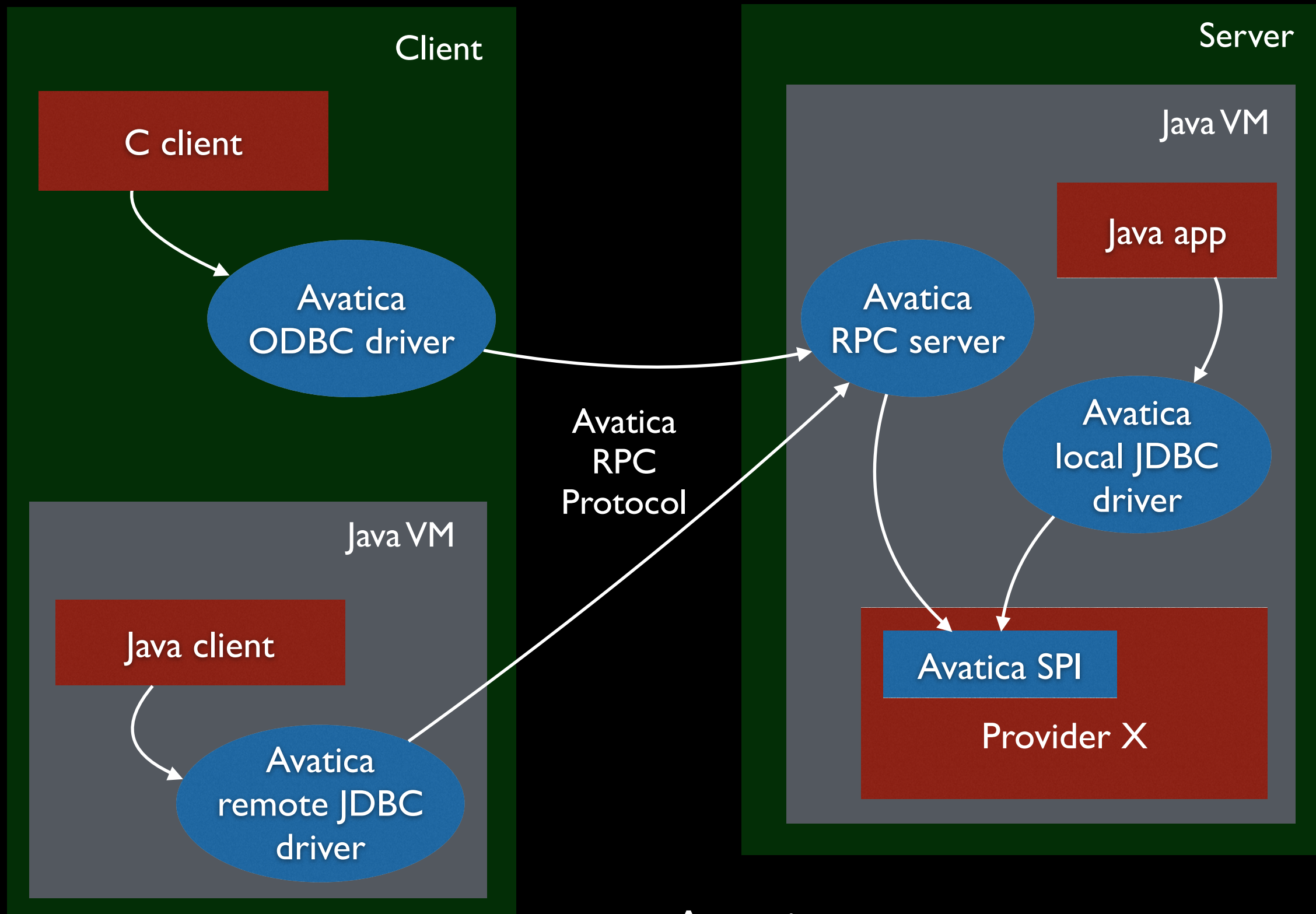# Avatica

An extensible, open source, ODBC and JDBC driver

Julian Hyde, 2013-12-19

Problem statement

- Today, many database engines and client applications are written in Java.

- But, some important client applications require ODBC.

- Maintaining a high-quality, portable ODBC driver is a lot of work (especially for projects whose developers mainly use Java).

Client

Server

C client

Avatica
ODBC driver

Java VM

Avatica
RPC
Protocol

Java client

Java VM

Avatica
remote JDBC
driver

Avatica
RPC server

Java app

Avatica
local JDBC
driver

Avatica SPI

Provider X

Avatica

Proposed solution: Avatica

- Database connectivity stack: ODBC driver, RPC protocol, remote JDBC driver, local JDBC driver.

- Start the project with a proven, portable ODBC driver (e.g. psqlODBC).

- Share the ongoing maintenance among projects. (Apache Phoenix, Apache Drill, Facebook Presto, Cascading & Optiq have expressed interest.)

- Currently just a proposal.

RPC protocol

- Language-independent

- Reasonably time- and space-efficient

- Easy to implement an RPC server

- Extensible (for example: a particular database might wish to send extra metadata to the client)

- Built from off-the-shelf components

- Options: Apache Thrift; Protobuf; Postgres protocol; ZeroMQ

ODBC driver

- Avatica ODBC driver speaks Avatica RPC protocol.

- Common customizations are made on the server: typical provider does not need to modify ODBC driver.

- Allow a provider to extend ODBC driver and extend wire protocol, if necessary.

- Proposal: Extend psqlODBC (Postgres ODBC driver), adding support for Avatica RPC protocol.

Avatica Server impl

- AvaticaServer implements RPC protocol in Java.

- To use it, each database must implement the provider SPI.

  - SPI is similar to JDBC (but simpler).

  - SPI also powers a local JDBC driver.

How to add a provider

- Suppose you have written a database, ProviderX.

- First, implement the provider SPI:

  - Specify provider name, version

  - Statement methods: prepare, execute and close

  - Implement calls for supported metadata (e.g. getTables)

- Next, run provider test and fix your SPI implementation.

- ODBC, remote JDBC, local JDBC should now work!

- Most projects will never need to touch the ODBC driver or protocol.

Status

- Local JDBC driver in use by Optiq and Apache Drill: github.com/julianhyde/optiq

- ODBC driver, RPC protocol, and RPC server are just proposals

- Project lead: @julianhyde (Optiq / Hortonworks)

- Other members:

  - Apache Drill

  - Apache Phoenix

  - Cascading

- Get involved on the optiq mailing list:
  https://groups.google.com/forum/#!forum/optiq-dev