VERTICA

Extending Vertica with the Latest Vertica Ecosystem and Open Source Initiatives

Tom Wall

vertica-opensrc@microfocus.com

MICRO FOCUS

Requirements

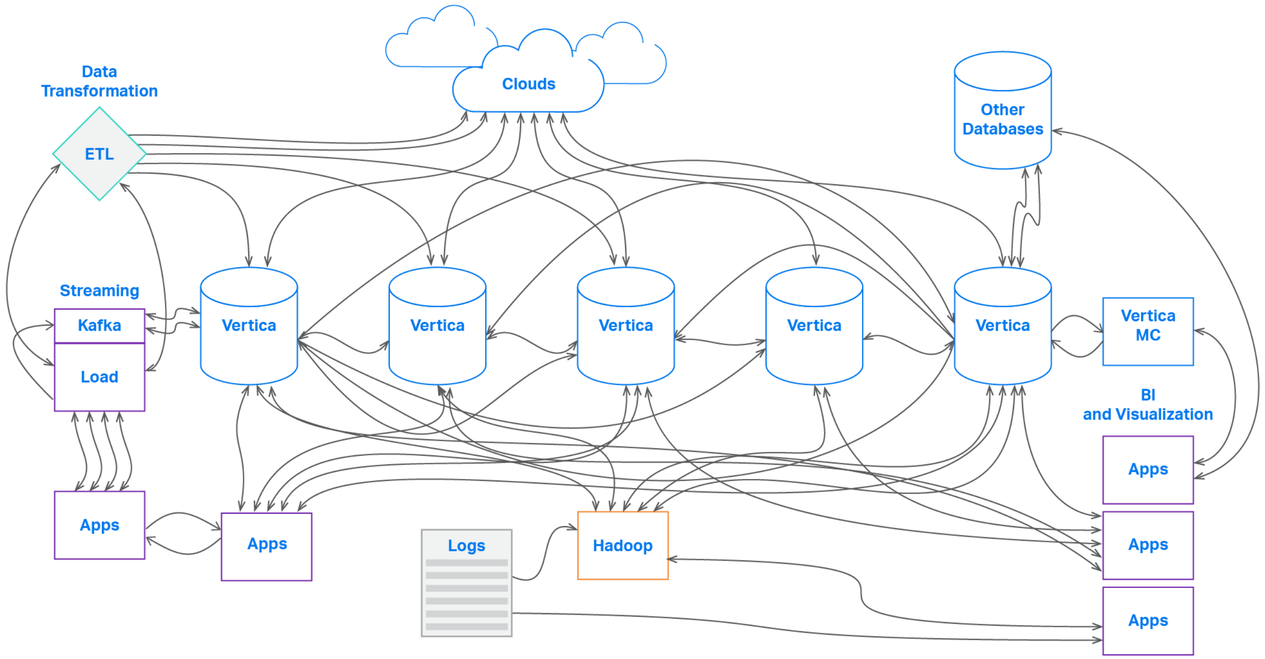
- Business Requirement: Build a fast dashboard to show our KPI’s
- Engineering Requirement: Do it in $LANGUAGE
  - Some are much easier than others
  - What if Vertica doesn’t support it?

PyODBC
• rpm –Uvh vertica-client*.rpm • find / -type f -name “odbc*.ini” • emacs odbc.ini • export ODBCINI= • pip install pyodbc • import pyodbc • <Unicode breaks> • <Play with settings> • <Recompile driver manager> • <Recompile python> • Goto 3

vertica-python
• pip install vertica-python • import vertica_python • Get to work!

Challenges of Big Data Engineering

- Every business is a software business
- Every business needs a data & analytics strategy
- Vertica users & use cases are diverse but we have a lot in common
  - Vertica is only one part of the puzzle
  - Technology landscape is constantly changing; we're all learning as we go
  - To be successful, developers need intuitive tools, libraries and examples



Vertica as a Programming Platform

- The SQL abstraction is great for “big data” workloads
  - Declarative syntax, well-known application semantics
  - We’ll be the experts in data management, you be the expert in your domain
- If you can’t do it with stock Vertica, there’s probably an extension point
  - Custom SQL & UDX Functions
  - Custom data formats & sources
  - Parallel loading & export
  - ML Models import & export

An Open Architecture with a Rich Ecosystem

Data Transformation

Apache Spark

Streaming

kafka

ETL

Informatica

ATTUNITY

User-Defined Loads

User-Defined Functions

R Java C++ Python

SQL

Geospatial | Real-time | Text Analytics

Event Series | VERTICA | Pattern Matching

Time Series | Machine Learning | Regression

User-Defined Storage

Security

ODBC JDBC

External Tables: Analyze in Place

Parquet hadoop Apache orc aws S3

BI & Visualization

looker

Qlik

+ableau SOFTWARE

Logi ANALYTICS

MicroStrategy

Microsoft Azure

aws

Google Cloud

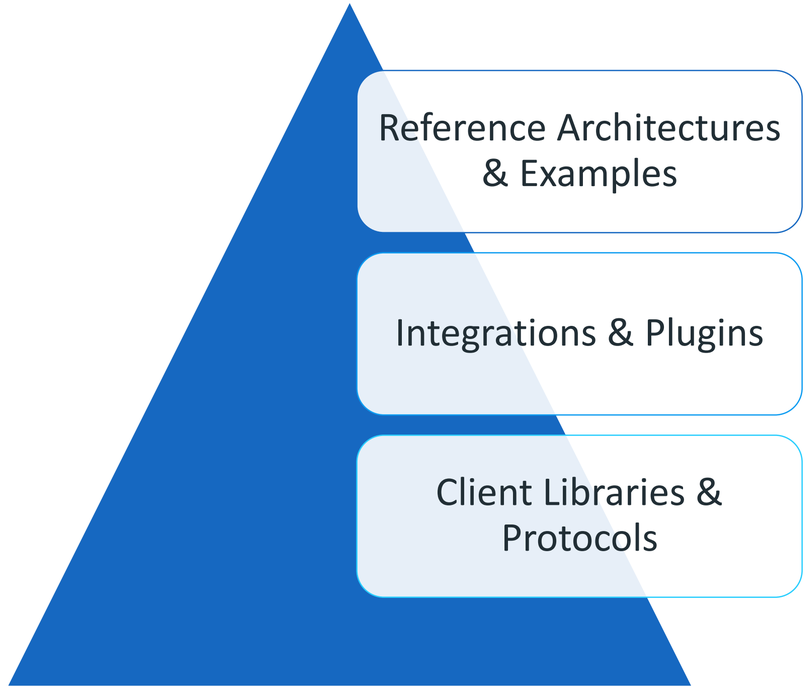
PURE STORAGE

hadoop

VERTICA

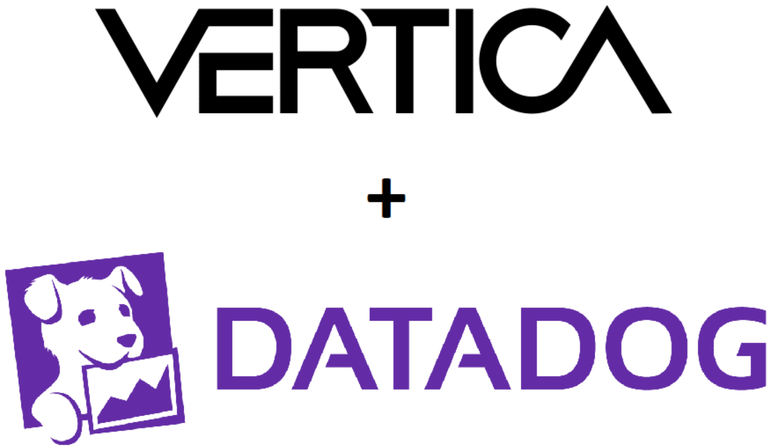
Open Technology at Vertica

- Open source technology plays a critical role in a modern data-driven enterprises
- The Vertica team is embracing an open source development model for new integration projects
  - Open source client libraries
  - Open source integrations
- We’re sharing more open source examples & references to make it easier to use Vertica the way you want



vertica-python

- Collaboration with Uber
- Use it to build your own Python apps
- Use it via tools written in Python
  - Vertica admintools (9.3.1)
  - DataDog integration
  - And more!
- Why Python?
  - DevOps, Data Science & Machine Learning
  - Internal needs: Python 2 EOL



vertica-python

- Prebuilt packages: pip install vertica-python
- Build it yourself: https://github.com/vertica/vertica-python

```
import vertica_python

# Initialize connection
conn_info = {'host': '127.0.0.1',
             'port': 5433,
             'user': 'some_user',
             'password': 'some_password',
             'database': 'vdb',
             'connection_load_balance': True}

# Open connection to the cluster
with vertica_python.connect(**conn_info) as conn:

    # Log the node we connected to
    cur = conn.cursor()
    cur.execute("SELECT NODE_NAME FROM V_MONITOR.CURRENT_SESSION")
    print("Client redirects to node:", cur.fetchone()[0])

    # Load some data
    with open("/tmp/binary_file.csv", "rb") as fs:
           cur.copy("COPY table(field1, field2) FROM STDIN DELIMITER ',' ENCLOSED BY '\"'",
                    fs, buffer_size=65536)
```

vertica-sql-go

- Collaboration with Micro Focus SecOps Group
- Use it to build your own go apps
- Use it via tools written in Go
  - Powers Grafana integration
- Why go?
  - Good balance of performance, concurrency, safety
  - Internal needs: Monitoring UI

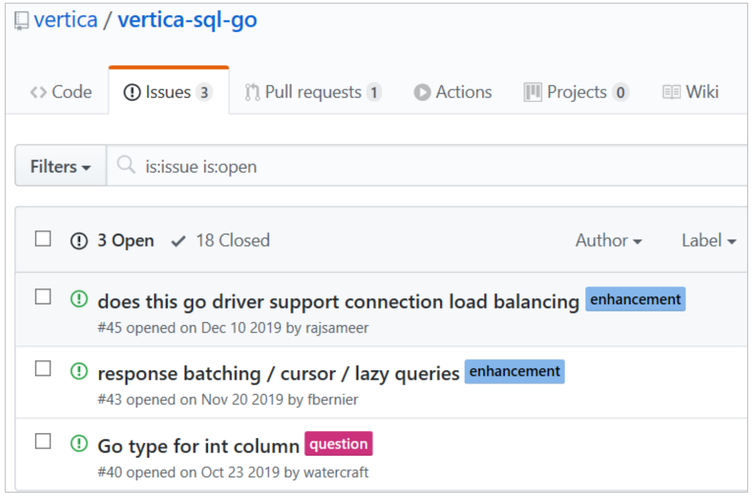
vertica-sql-go

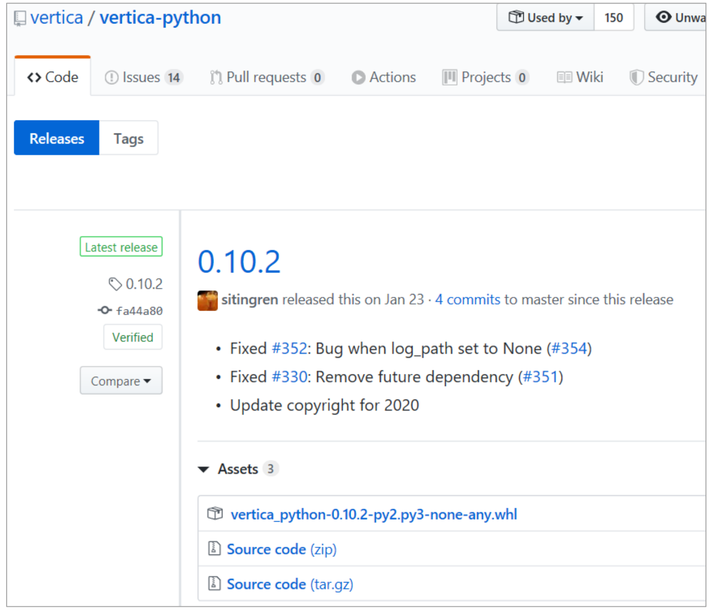
- go get github.com/vertica/vertica-sql-go

```
func TestBasicQuery(t *testing.T) {
        connDB := openConnection(t)
        defer closeConnection(t, connDB)

        rows, err := connDB.QueryContext(ctx, "SELECT * FROM v_monitor.cpu_usage LIMIT 5")
        assertNoErr(t, err)

        defer rows.Close()

        columnNames, _ := rows.Columns()
        for _, columnName := range columnNames {
                testLogger.Debug("%s", columnName)
        }

        for rows.Next() {
                var nodeName string
                var startTime string
                var endTime string
                var avgCPU float64

                assertNoErr(t, rows.Scan(&nodeName, &startTime, &endTime, &avgCPU))

                testLogger.Debug("%s\t%s\t%s\t%f", nodeName, startTime, endTime, avgCPU)
```

Open Source Development, Not Just Code

Ask questions, report issues, submit PR's and collaborate directly with the engineering team and other Vertica users on Github

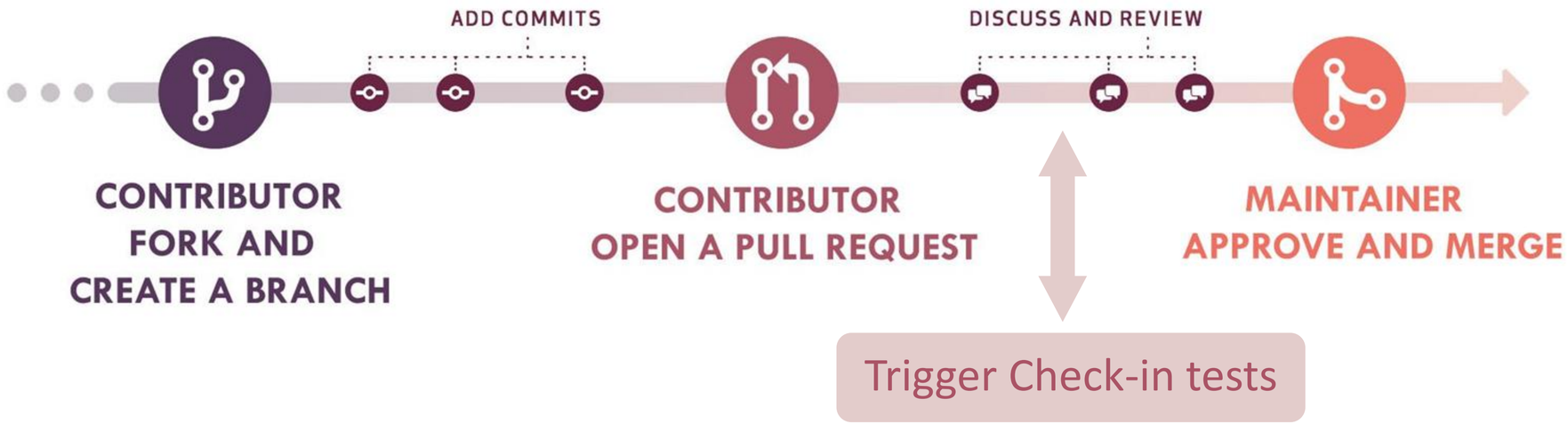




2019 vertica-python stats

- 11 Releases
- 40 Issues resolved
- 78 PR's merged to master
- 5k pypi downloads/day

Open Source Development, Not Just Code



- All projects using the Apache 2.0 License
- Ongoing work will merge to master branch as it is ready and stable
- You'll always be able to build clients yourself directly from the latest source on GitHub
- Fully automated local & PR testing with Travis CI (vertica-python)

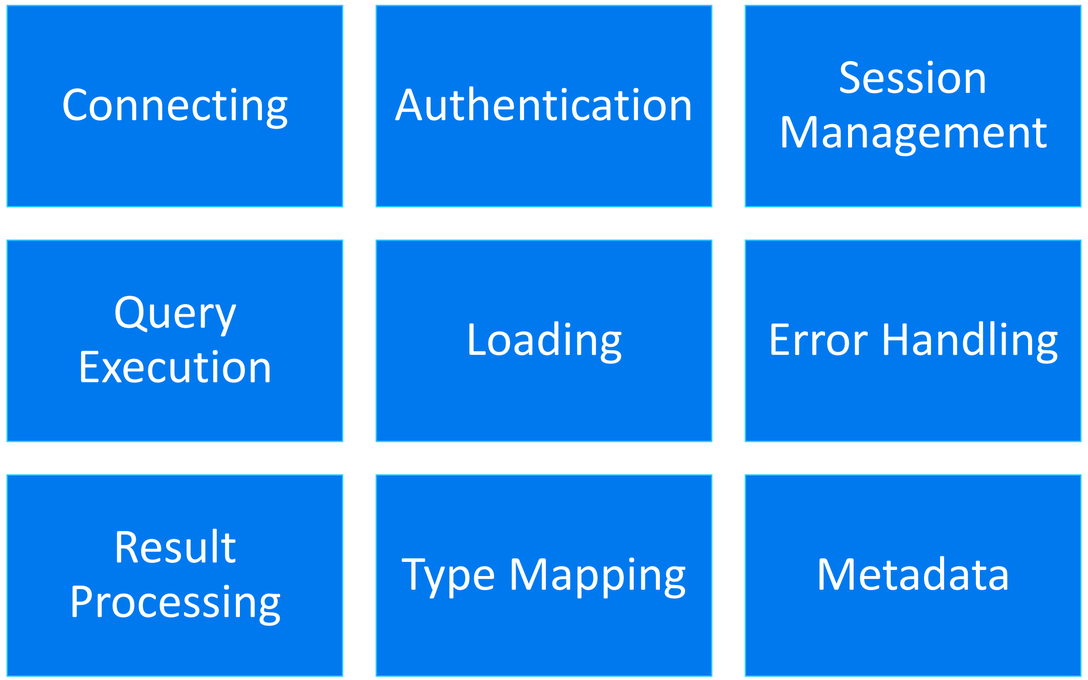
Working Better Together

- Vertica open source projects offer some exciting opportunities to work together
  - Collaborate directly with engineers
  - Contribute improvements and help guide the direction of the projects
  - Share knowledge, implementation details & best practices
- Don’t use Go or Python? It still matters!
  - There’s a lot of amazing Vertica developers out there who do
  - These clients are low-level building blocks for all kinds of exciting things
  - The discussions, implementations, examples & testing strategies generalize to many different use cases

Open Client Libraries

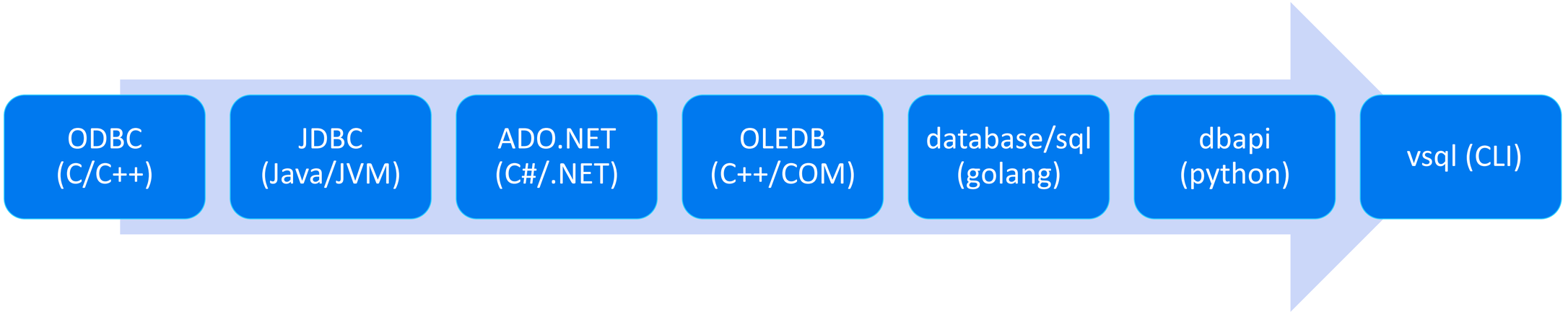
Database Client Interfaces

- SQL only specifies how to manipulate data in the database
- Database client interfaces address programmer's needs that surround the processing of SQL
- Specific to a language or technology stack, though use cases are often the same



Database Client Standards

- Some client interfaces are robust enough to support a pluggable "driver" model.
- Most aren’t, but that’s okay
  - Every database is unique so no standard will cover all use cases and features
  - De-facto standards arise via common conventions and design patterns
- To compensate, database tools invent extensible glue code & plugin layers for further customization



Semantics defined by standards

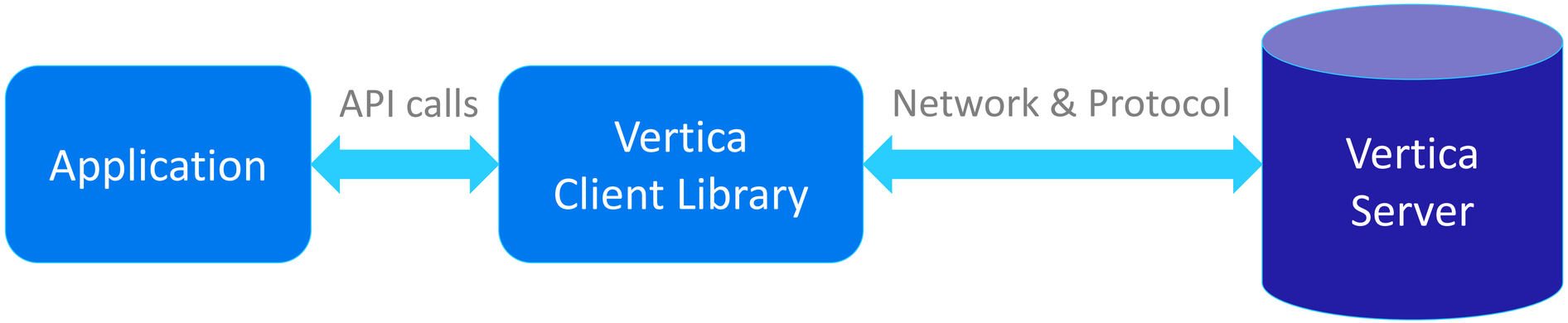
Semantics defined by implementation

Connecting to Vertica Database

- What happens under the covers?

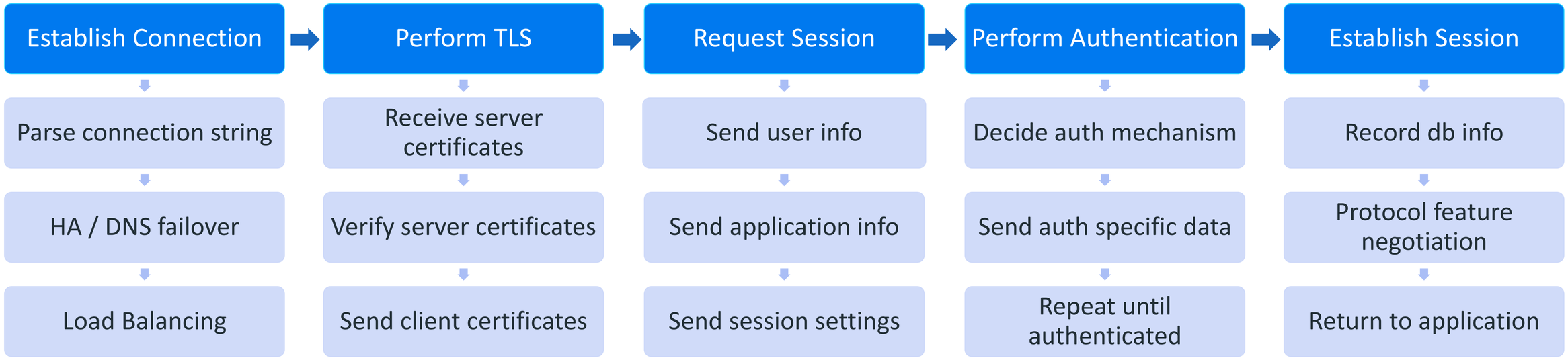
Connecting to Vertica Database

- Applications invoke API calls in a client library
- Library issues protocol operations to Vertica to implement request
- Typical APIs: Connect; Execute Query; Row Iterator, Commit Txn, etc.



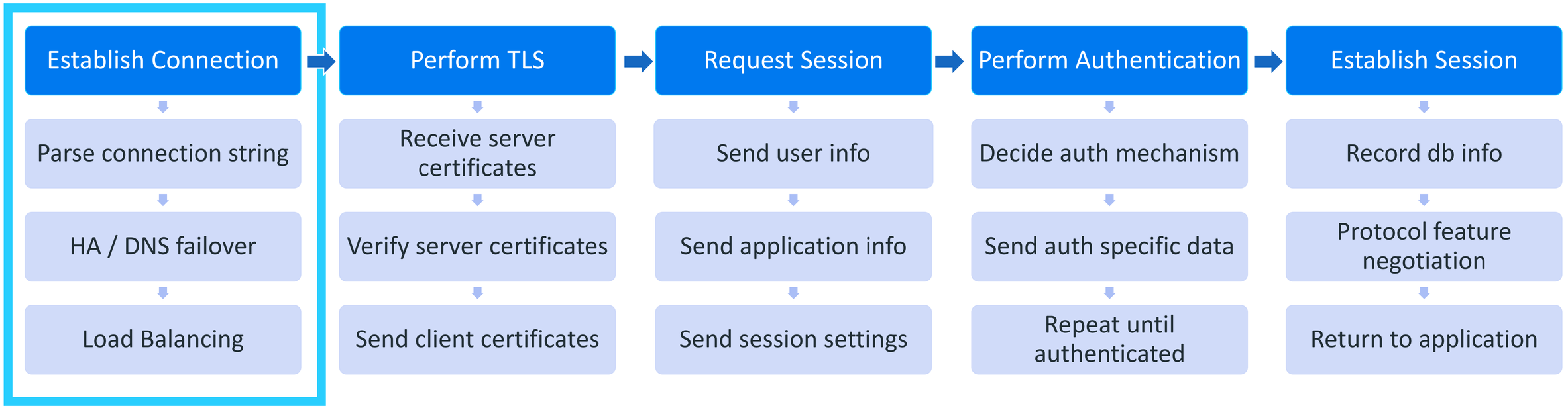
Establishing a Database Connection

vertica_python.connect(“host=vcluster.company.com; user=twall; password=hunter2; …”)



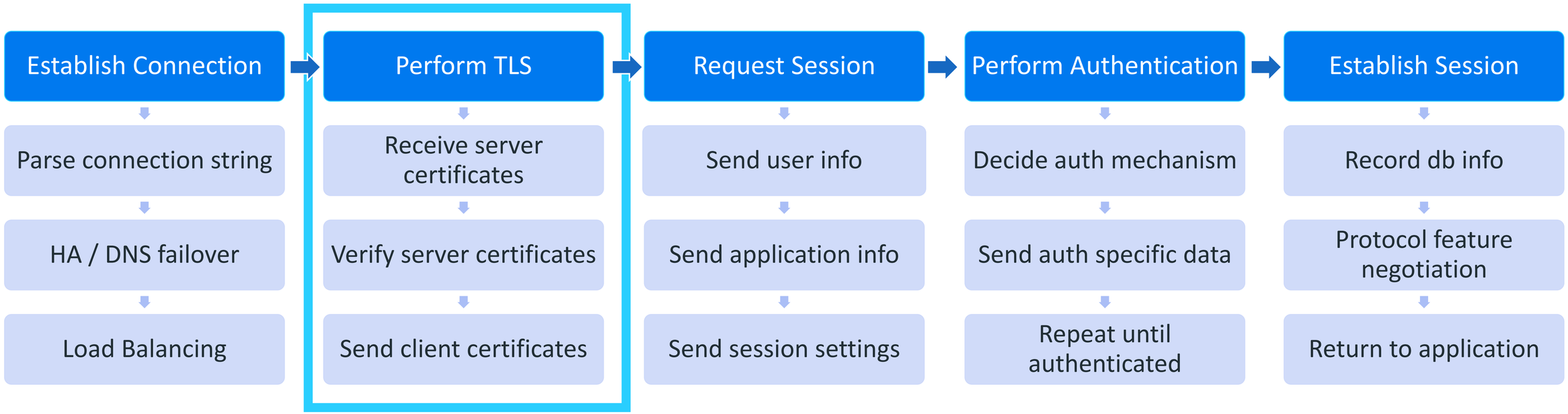
Establishing a Database Connection

vertica_python.connect(“host=vcluster.company.com; user=twall; password=hunter2; …”)



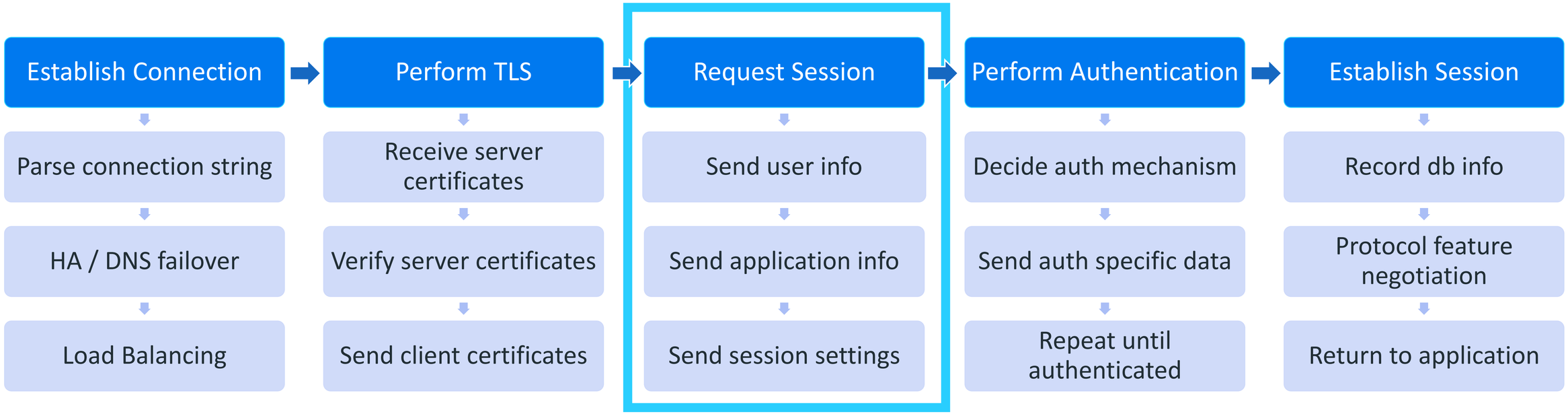
Establishing a Database Connection

vertica_python.connect(“host=vcluster.company.com; user=twall; password=hunter2; …”)



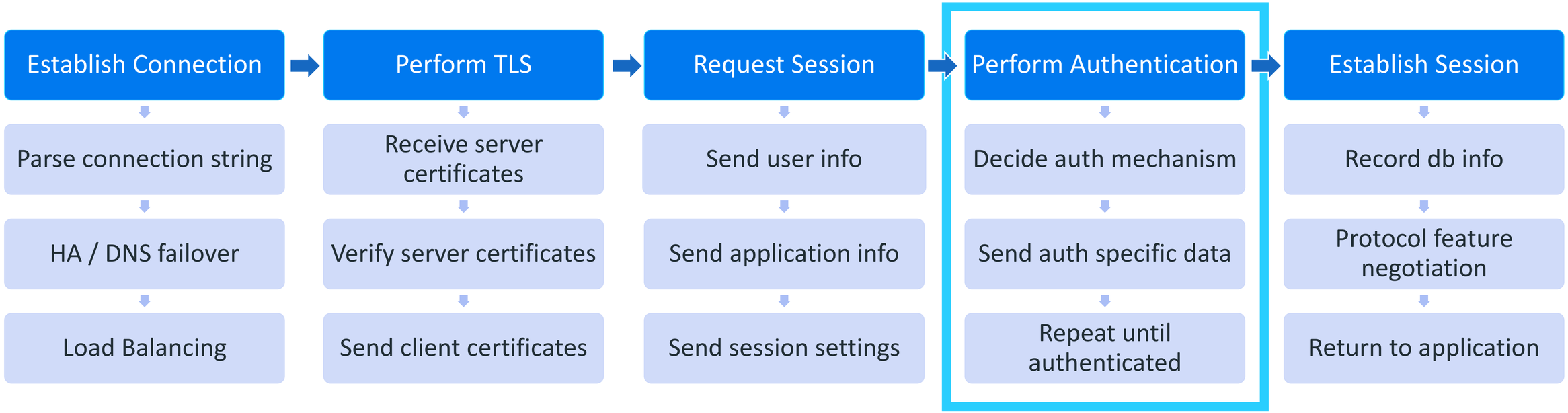
Establishing a Database Connection

vertica_python.connect(“host=vcluster.company.com; user=twall; password=hunter2; …”)



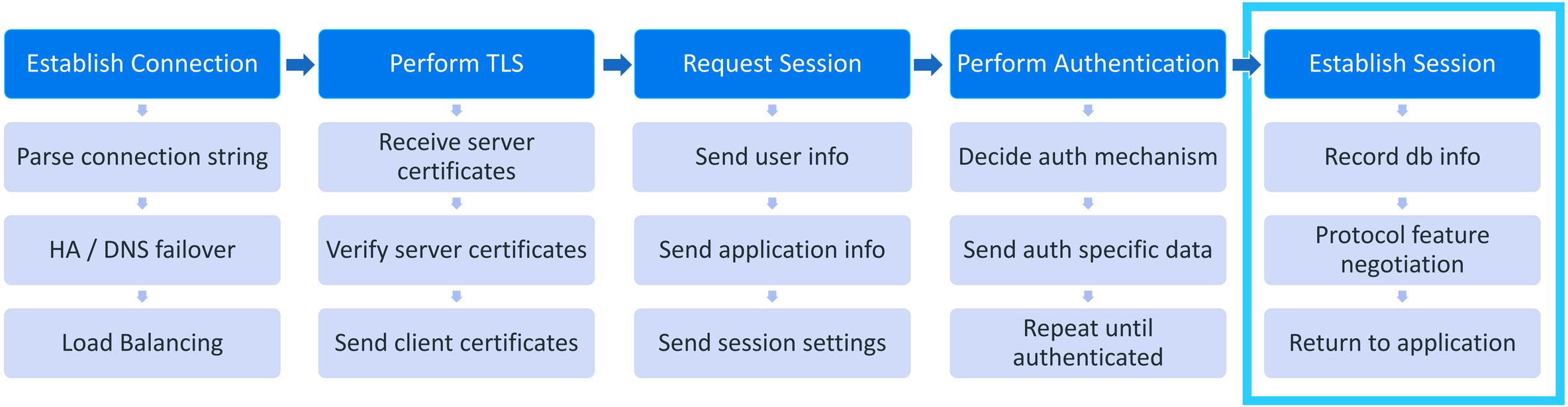
Establishing a Database Connection

vertica_python.connect(“host=vcluster.company.com; user=twall; password=hunter2; …”)



Establishing a Database Connection

vertica_python.connect(“host=vcluster.company.com; user=twall; password=hunter2; …”)



Opening Up The Vertica Client Wire Protocol

- The Vertica client protocol is derived from Postgres, but it differs in many ways
  - Not all Postgres protocol features work with Vertica (e.g. large objects, advanced cursors)
  - The Vertica protocol supports features not present in Postgres (e.g. load balancing)
- Vertica-python serves as an open reference implementation of the Vertica client protocol

Connection Management

- Connection load balancing
- HA Connections & DNS failover
- Backwards compatibility

Query Dispatch

- Prepared statements
- Streaming Results
- Query Cancellation

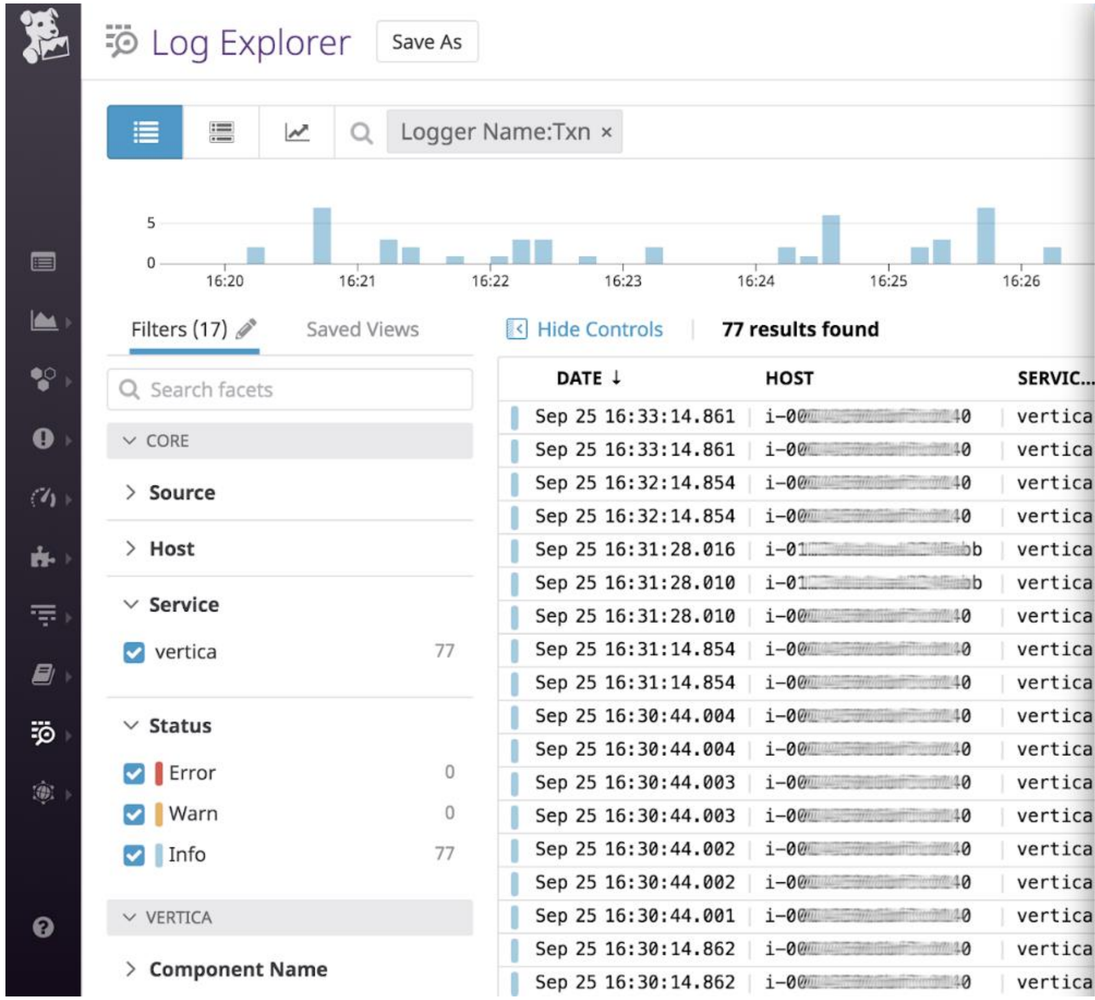
Application Support

- Data Types (long strings, UUID)
- Authentication methods (SHA, Kerberos)
- Metadata

vertica-python protocol features introduced since August 2018
vertica-sql-go implementation in progress

Building Protocol-Aware Applications & Tools

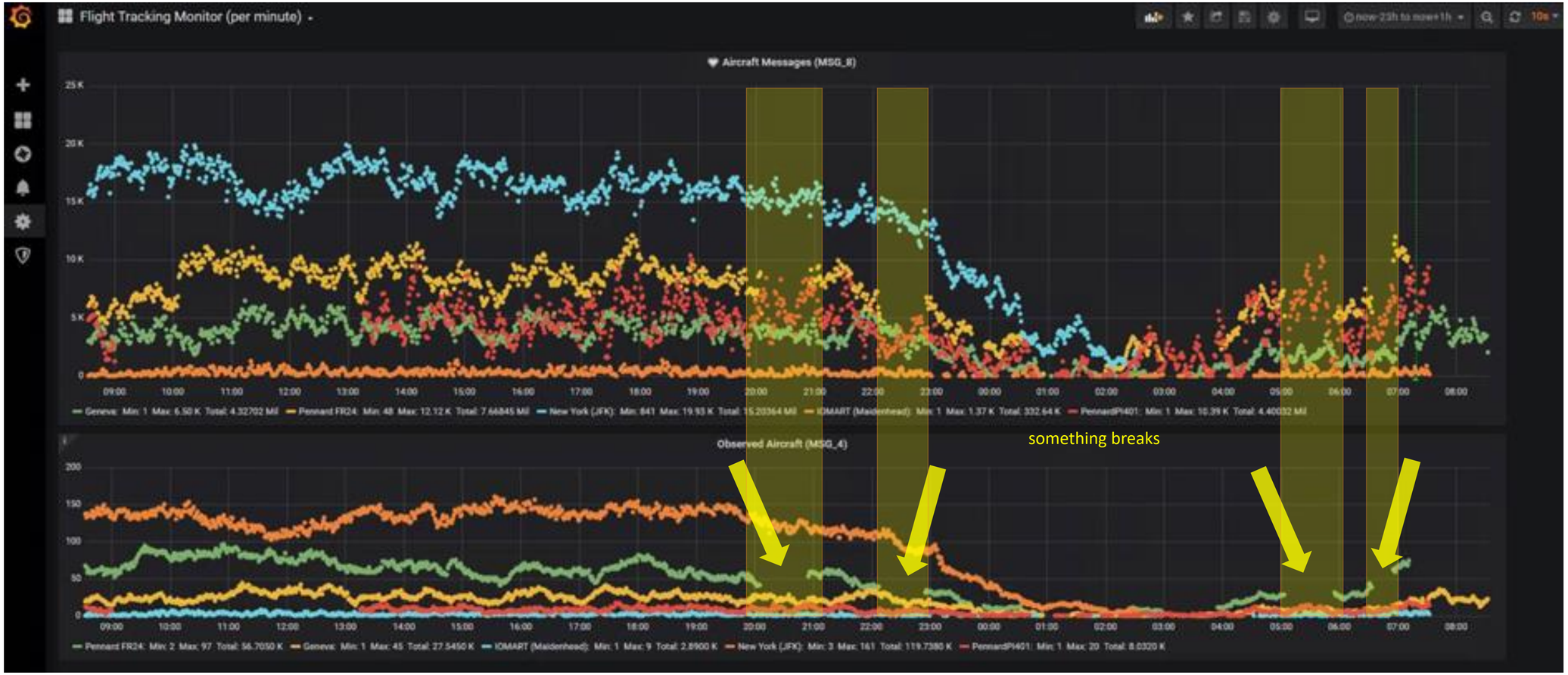
- An open client & protocol implementation enables other exciting projects
  - Database clients for other languages
  - Mocking & testing tools
  - Security & network monitoring products
  - Smart proxies & query routing (https://eng.uber.com/queryparser/)
- These are mostly just ideas today
  - Send us your ideas and requests
  - We're happy to offer advice and collaborate



DataDog Vertica Log Explorer

Open Database Integrations

Working With Data

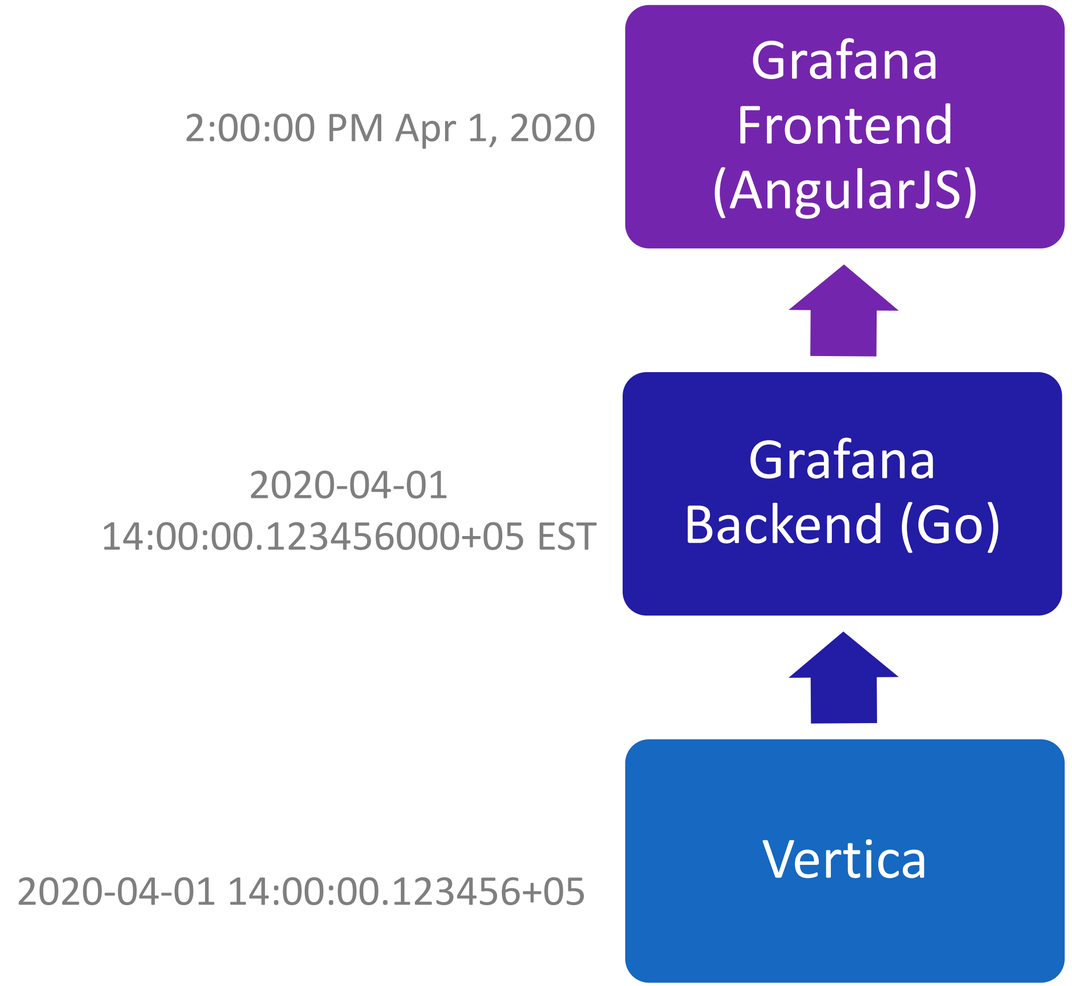


Kafka-Vertica-Grafana Flight Tracking Demo

https://www.vertica.com/blog/pennard-we-have-a-problem-troubleshooting-flight-tracking-with-the-grafana-plugin-for-vertica/

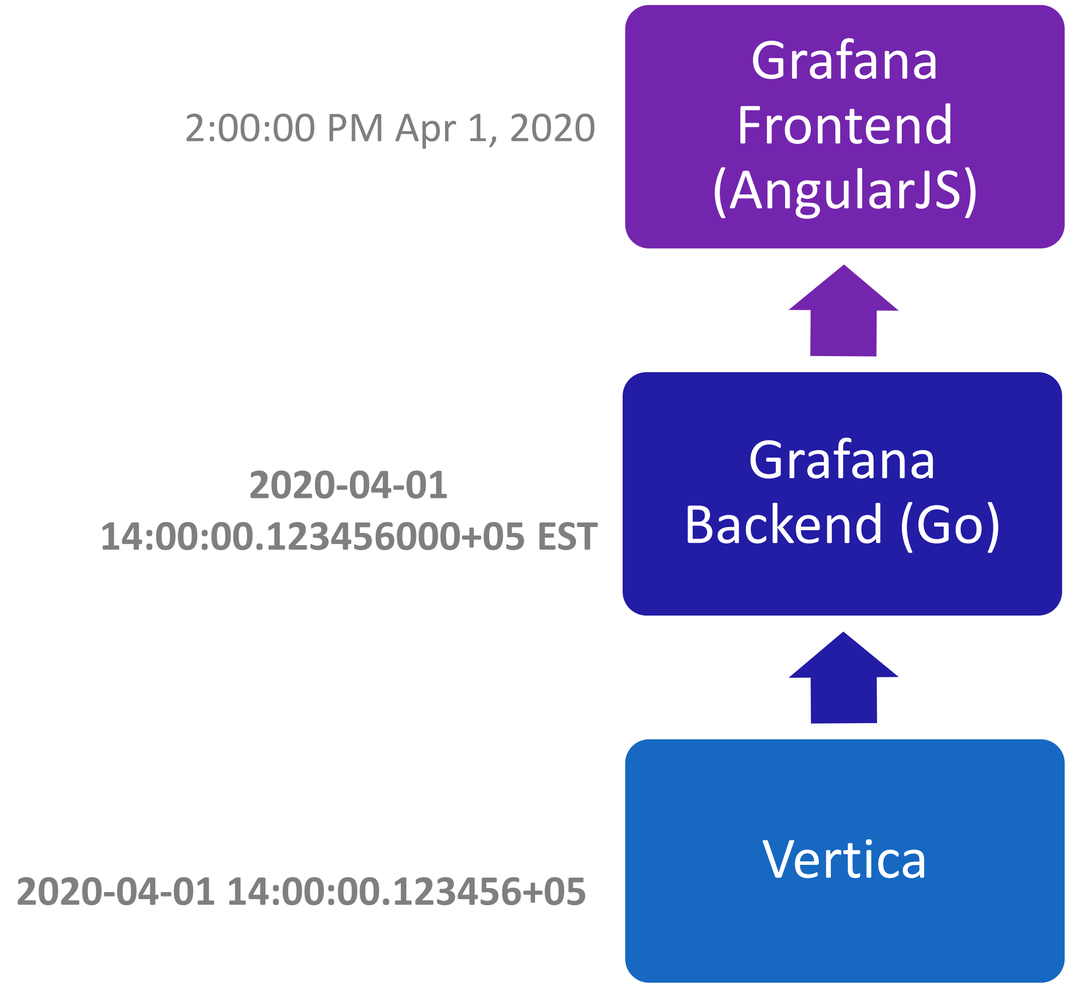
Working With Data

- Grafana specializes in visualizing timeseries data
- Time is challenging
  - Time Zones, DST, Leap Seconds, -Infinity
  - Every system does it differently
- How are semantic differences reconciled across domains?
  - Vertica Timestamp converted to Go time.Time
  - Go time.Time converted to AngularJS Date
  - Value space is the intersection of each system



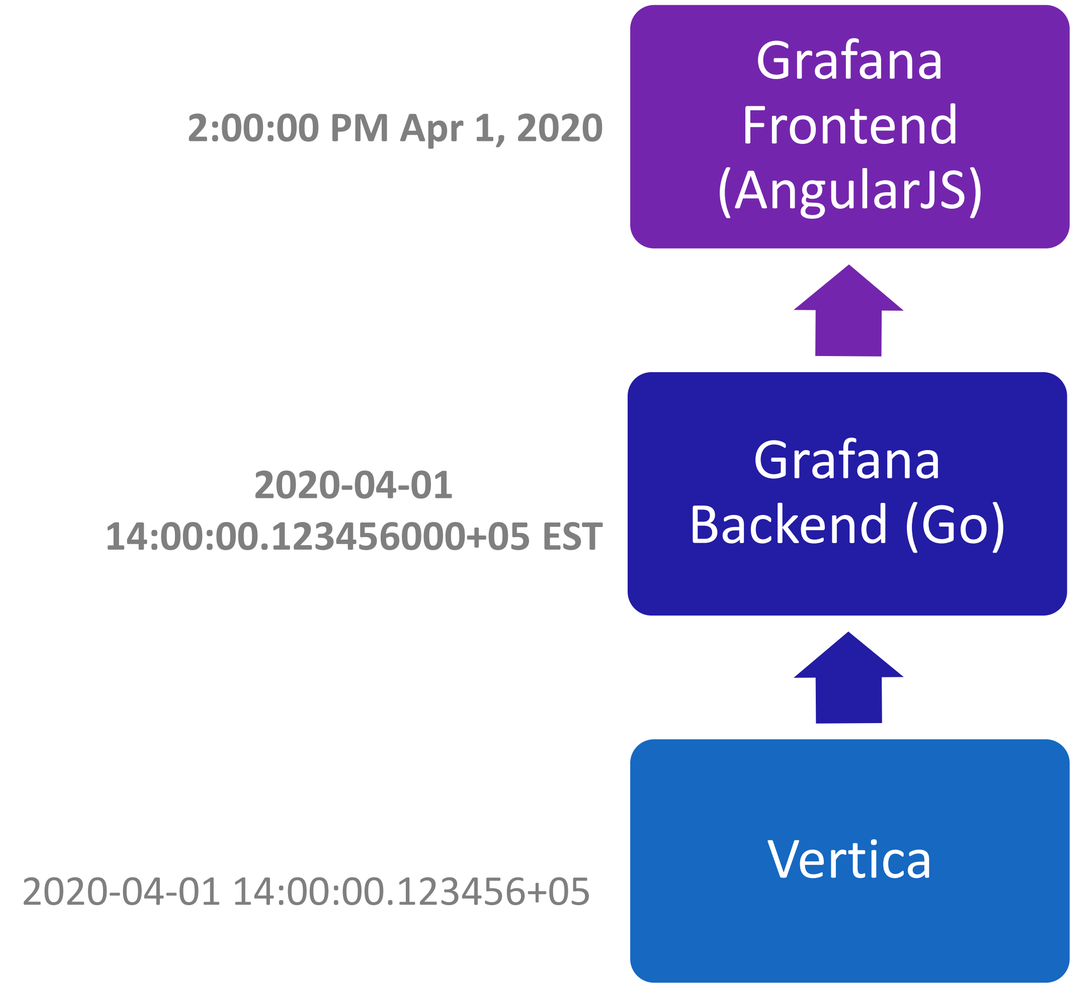
Working With Data

- Grafana specializes in visualizing timeseries data
- Time is challenging
  - Time Zones, DST, Leap Seconds, -Infinity
  - Every system does it differently
- How are semantic differences reconciled across domains?
  - **Vertica Timestamp converted to Go time.Time**
  - Go time.Time converted to AngularJS Date
  - Value space is the intersection of each system



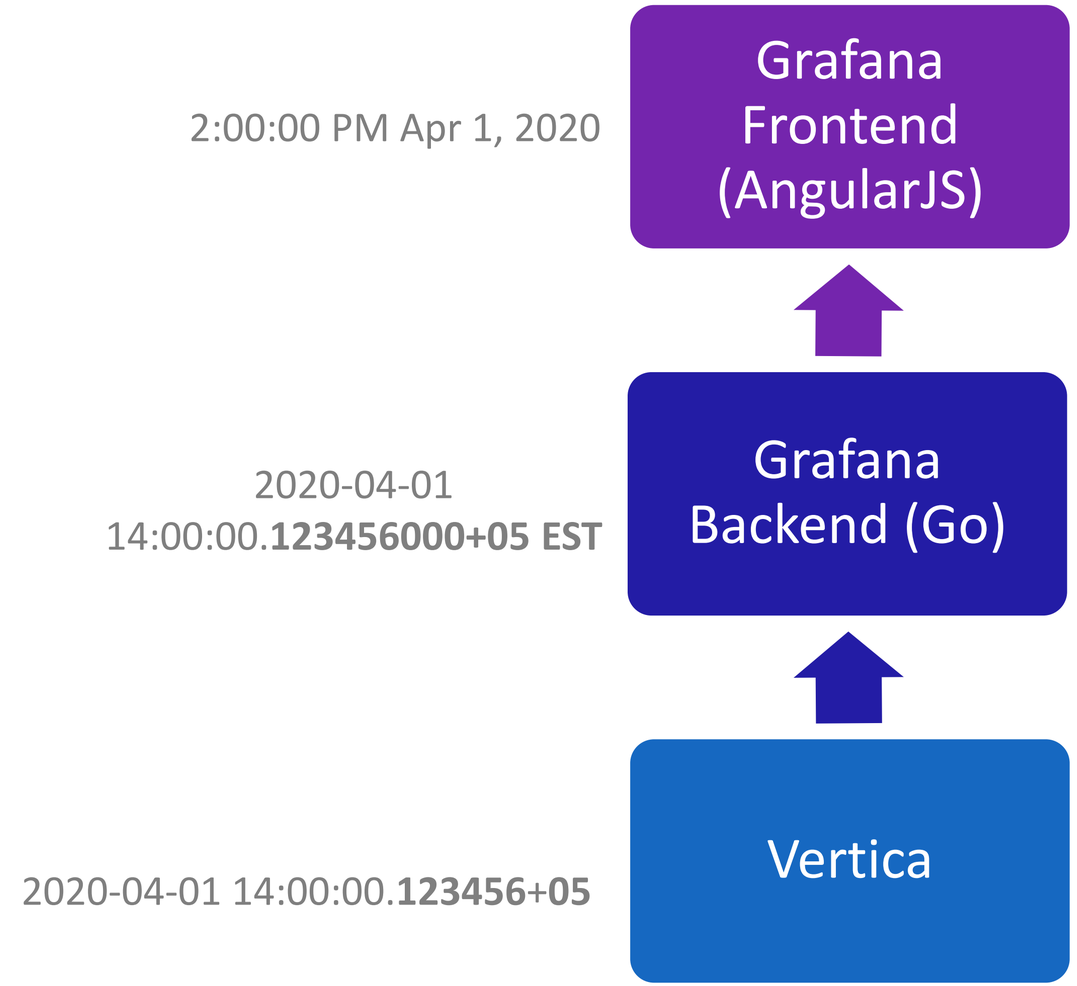
Working With Data

- Grafana specializes in visualizing timeseries data
- Time is challenging
  - Time Zones, DST, Leap Seconds, -Infinity
  - Every system does it differently
- How are semantic differences reconciled across domains?
  - Vertica Timestamp converted to Go time.Time
  - **Go time.Time converted to AngularJS Date**
  - Value space is the intersection of each system



Working With Data

- Grafana specializes in visualizing timeseries data
- Time is challenging
  - Time Zones, DST, Leap Seconds, -Infinity
  - Every system does it differently
- How are semantic differences reconciled across domains?
  - Vertica Timestamp converted to Go time.Time
  - Go time.Time converted to AngularJS Date
  - **Value space is the intersection of each system**



Working With Data

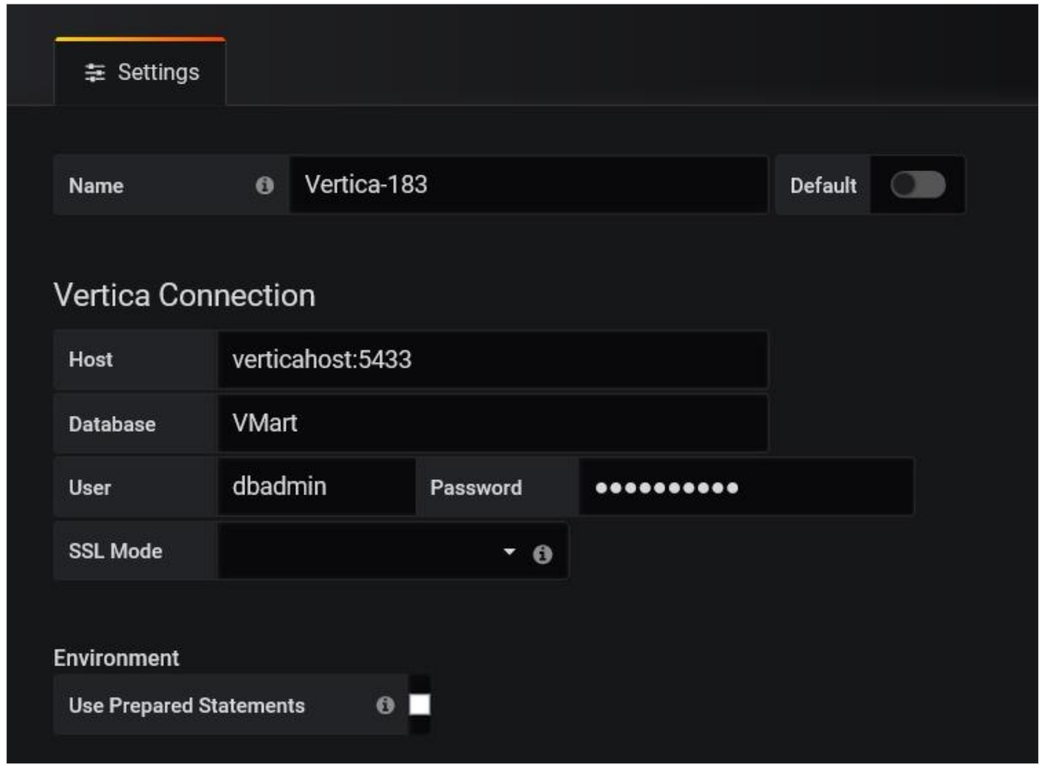
Working With Data

Working With Data

Working With Metadata

Grafana needs to know how time works for each data source, beyond basic I/O

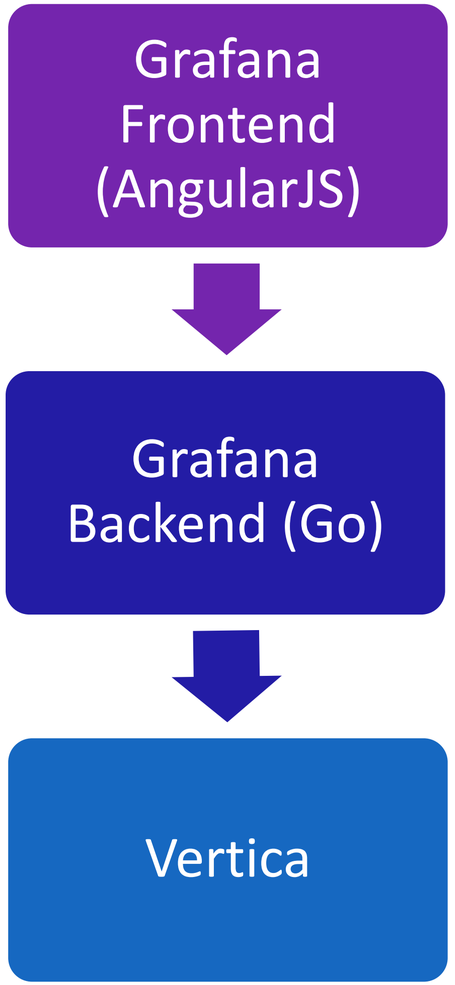
- **Configuration:** How to connect to the data source to extract time data?
- **Capabilities:** What time types & functions does this data source support?
- **Semantics:** What range of values are allowed for those types?
- **Syntax:** What do time literals look like?
- **Catalog:** What tables have time columns in them?



Grafana-Vertica Connection UI

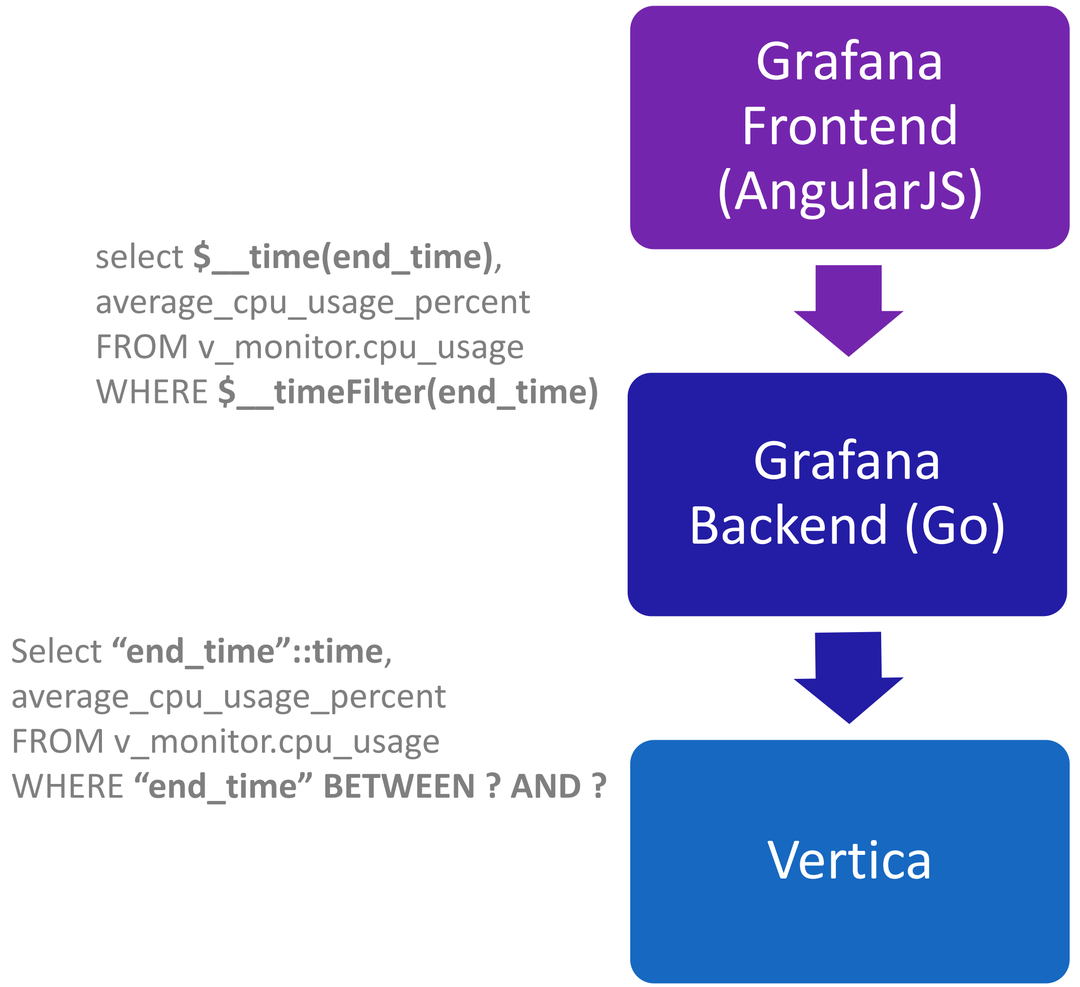
Working With Metadata

- Go's database/sql interface doesn't offer many metadata interfaces
- To compensate, Grafana's plugin frameworks define a standardizing layer where each data source can provide hints & customization
  - Frontend UI plugins in HTML/AngularJS
  - Backend data processing plugins in go
- The **vertica-grafana-datasource** project leverages Grafana's plugin frameworks to provide Vertica functionality
  - Frontend plugin customizes Connection UI and SQL query dispatch
  - Backend plugin uses **vertica-sql-go** to provide data, convert data types



Working With Metadata

- Plugin framework defines a set of standard functions that every data source must implement
  - $__time(), $__timeFilter(), etc.
- Vertica-grafana-datasource rewrites them in terms of Vertica SQL syntax
  - $__time() rewritten as a Vertica cast
  - $__timeFilter() becomes BETWEEN predicate
  - $__timeFilter() parameters filled in according to UI interactions



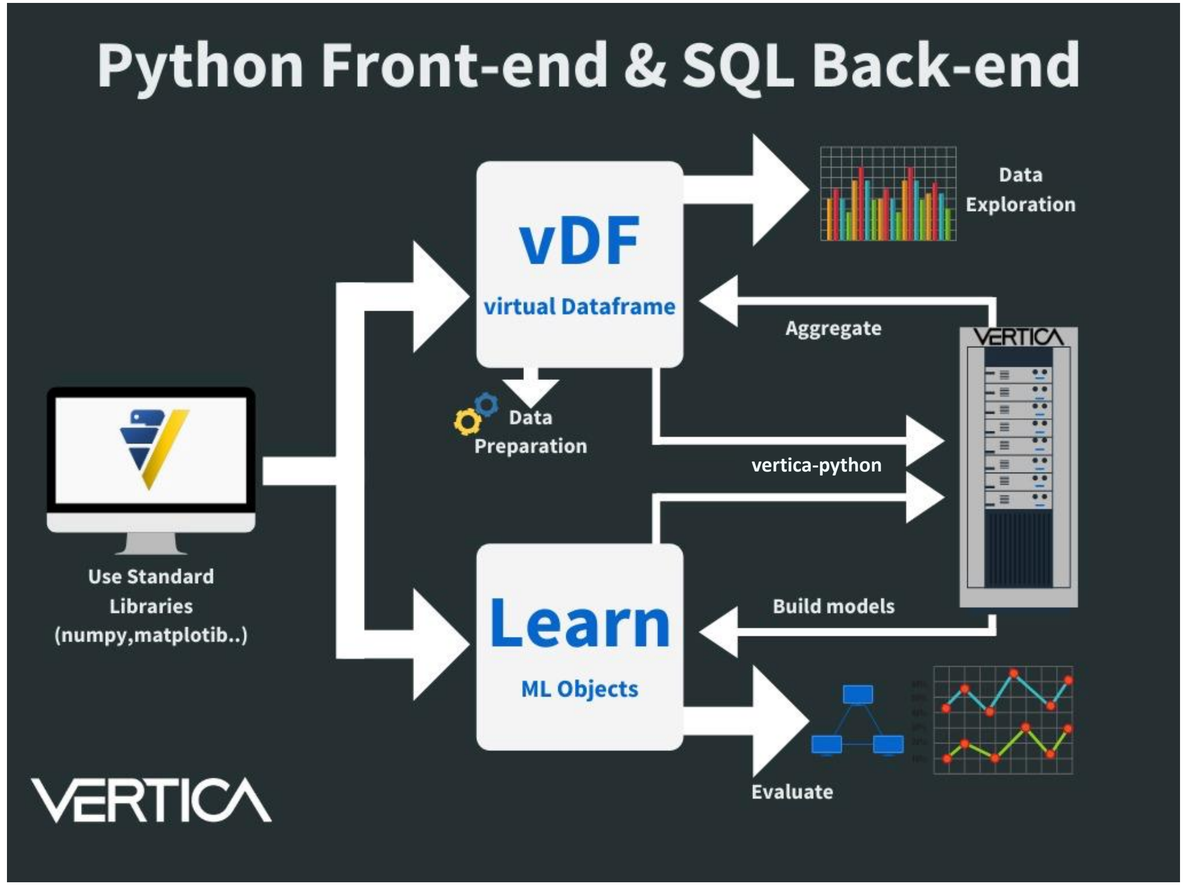
Open Reference Architectures & Examples

Advanced Integrations – Beyond Standards

- SQL and the surrounding standards were key to Vertica's early success
  - Generic database tools work great with Vertica thanks to robust standards
  - They will continue to play an important role
- The big data technology space is moving faster than standards can keep up
  - Advanced analytics & pushdowns
  - Advanced data-oriented application workflows
- Open implementations make for easier integrations and applications elsewhere
  - Prefer $TOOL over Grafana for dashboards?
  - They probably have to deal with the same kinds of problems

Vertica Machine Learning Integrations

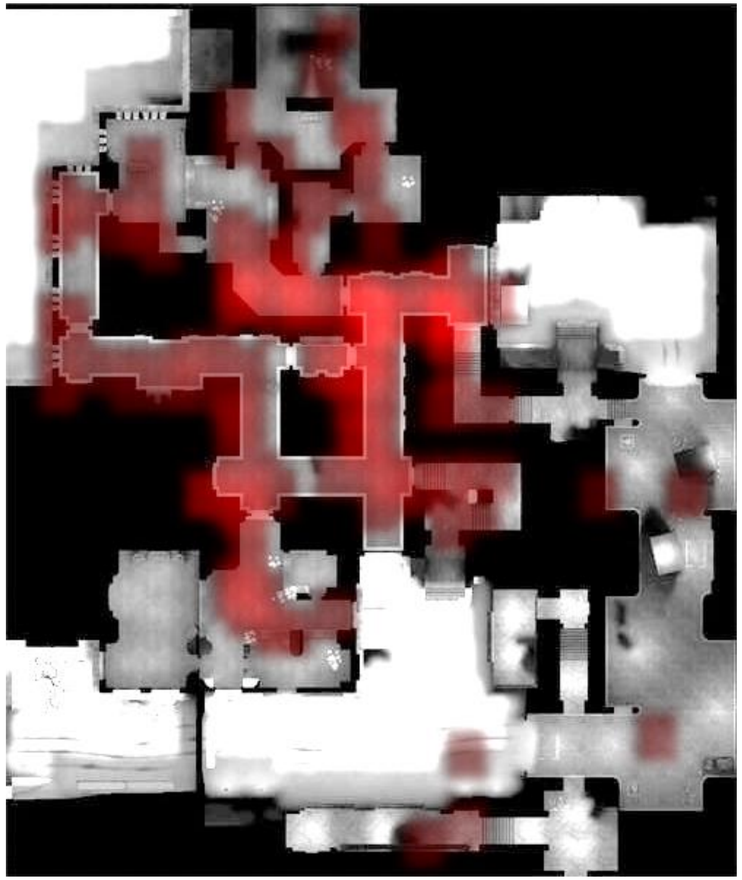
- Many excellent tools have arisen to make data science and machine learning easier
- Some of them have basic database connectivity
- None take full advantage of Vertica
- The **vertica-ml-python** demonstrates a familiar machine learning workflow that scales with Vertica
  - Experience similar to scikit-learn
  - Aggregation & processing happens in Vertica
  - Use it yourself, or see the code to customize your own approach



Vertica-ml-python data architecture

Vertica UDX Examples

- The Vertica User Defined eXtensions (UDX) SDK lets you extend the operators in a load and query plan
  - You write your domain logic in C++, Java, Python & R
  - You call them with SQL
  - We bring your logic to the data and make it fast
- UDX examples demonstrate how to use the Vertica SDK to solve interesting problems
  - Load from curl endpoints
  - Load from external databases via ODBC
  - Custom parsers & string processing
  - Heat map overlays
  - And more!



Vertica-generated heatmap

Data generated from a Vertica intern gaming session

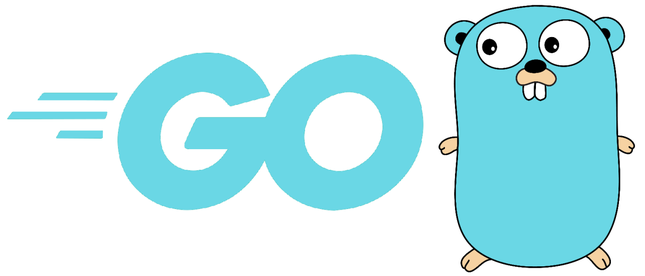
Vertica Integration Testing

- The client test automation strategy generalizes beyond database clients
  - Test structure and support code to facilitate smooth, deterministic results
  - Automated download, install & deploy of Vertica Community Edition
  - Testing multiple for multiple environments with docker & tox
- Today: see how we do it in vertica-python
- Future: standalone testbed starter project

Travis CI

Open Source Futures

Tentative Roadmap

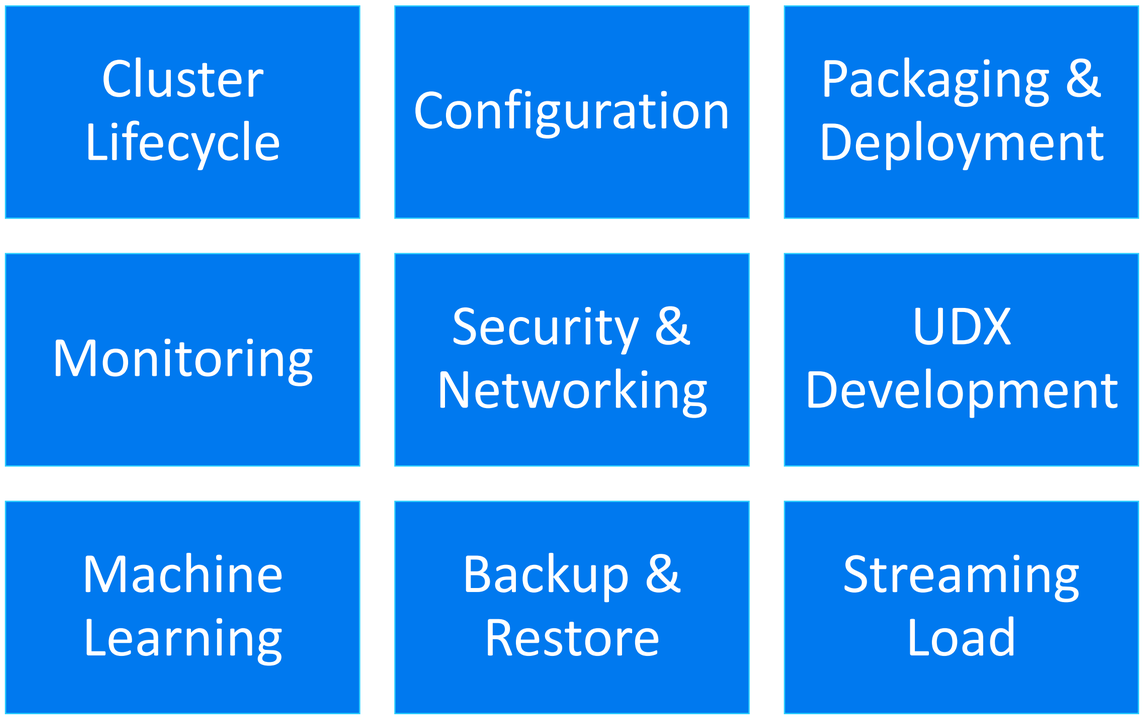


- Protocol complete for v1.0.0
  - COPY LOCAL
  - Complex Types (New in Vertica 10!)
- Cursor enhancements
- <your request here>

- Continue with protocol implementation
  - Load Balancing
  - Full data type support
  - Advanced authentication mechanisms
- Test automation
- <your request here>

Beyond Database Clients

- What about integration needs beyond SQL dispatch & result processing?
- These aren't standardized but there's many common tools & workflows
- Well established protocols, architectures & reference implementations are the next best thing



Announcing Awesome-Vertica

- A collection of interesting Vertica open source projects, tools and examples from the Vertica community
- We've seeded it with some of our favorites; if you want to add yours submit a PR!
- https://github.com/vertica/awesome-vertica

Inspired by the awesome project:
https://github.com/sindresorhus/awesome

Thank You

- Special thanks to project leads Siting Ren (vertica-python) & Roger Huebner (vertica-sql-go)
- Thanks to all of you out there that have already contributed!
- Reach us all at vertica-opensrc@microfocus.com to ask questions, provide feedback and discuss new ideas. Or post a github issue ☺

- https://github.com/vertica
- https://github.com/vertica/vertica-python
- https://github.com/vertica/vertica-sql-go
- https://github.com/vertica/vertica-grafana-datasource/
- https://github.com/vertica/Vertica-ML-Python
- https://github.com/vertica/UDx-Examples/
- https://github.com/vertica/awesome-vertica


VERTICA
Q&A
Learn More: academy.vertica.com
Try it Free: vertica.com/try
MICRO FOCUS

Vertica Academy

Your source for new comprehensive Vertica on-demand training

- Self-Paced Learning
- Online Training
- Knowledge Check
- Certificate of Completion & Certifications

Sign up for free today at **academy.vertica.com**

VERTICA

Vertica Forums

Vertica Engineers are available to answer your questions and moderate discussions right now

For more information visit **forum.vertica.com**