GridGain

Adding Speed and Scale to MySQL Database Deployments

Rob Meyer
Outbound Product Management, GridGain

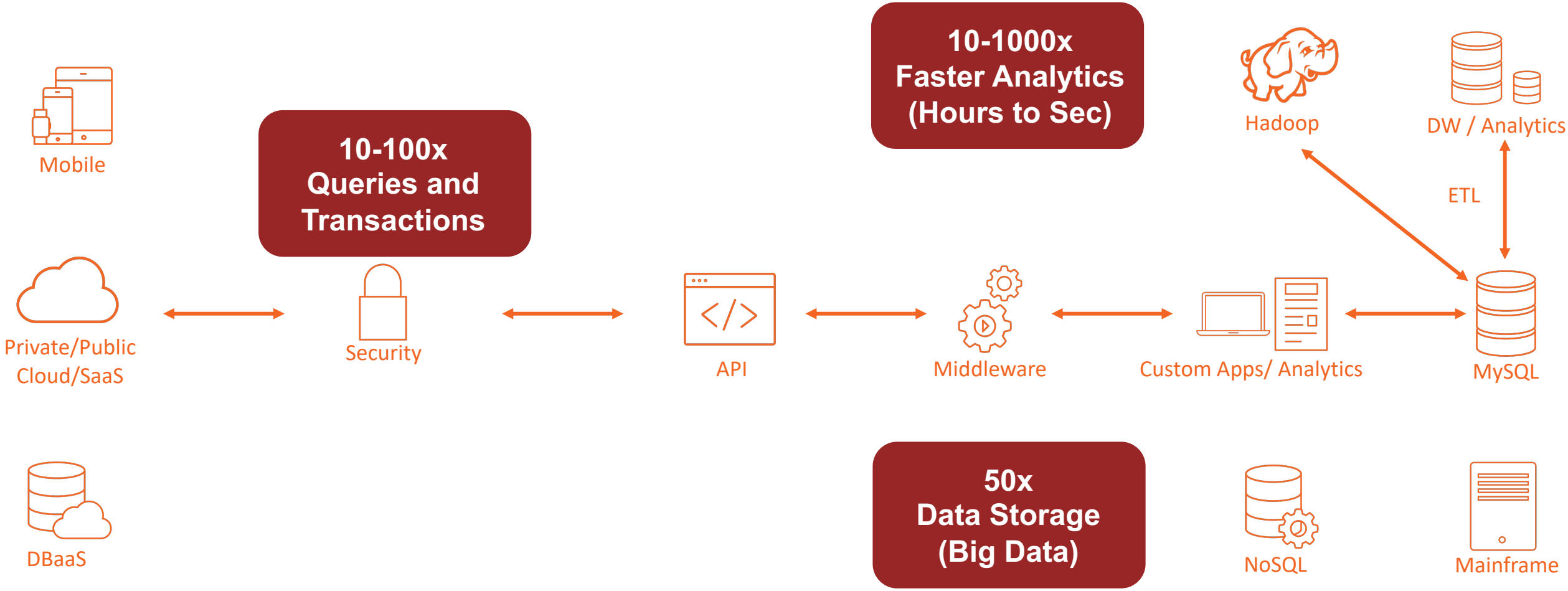
June 26, 2019

History of MySQL

- MySQL AB founded 1994
- MySQL Acquires Alzato Tech 2003
- Oracle Acquires InnoDB in 2005
- Percona Server for MySQL GA 2006
- Clustrix founded November 2006
- Galera Cluster released 2007
- MySQL acquired by Sun 2008
- Sun acquired by Oracle Jan 2010
- MariaDB started same day 2010
- Clustrix acquired by MariaDB Sep 2018

Challenges with Speed and Scale

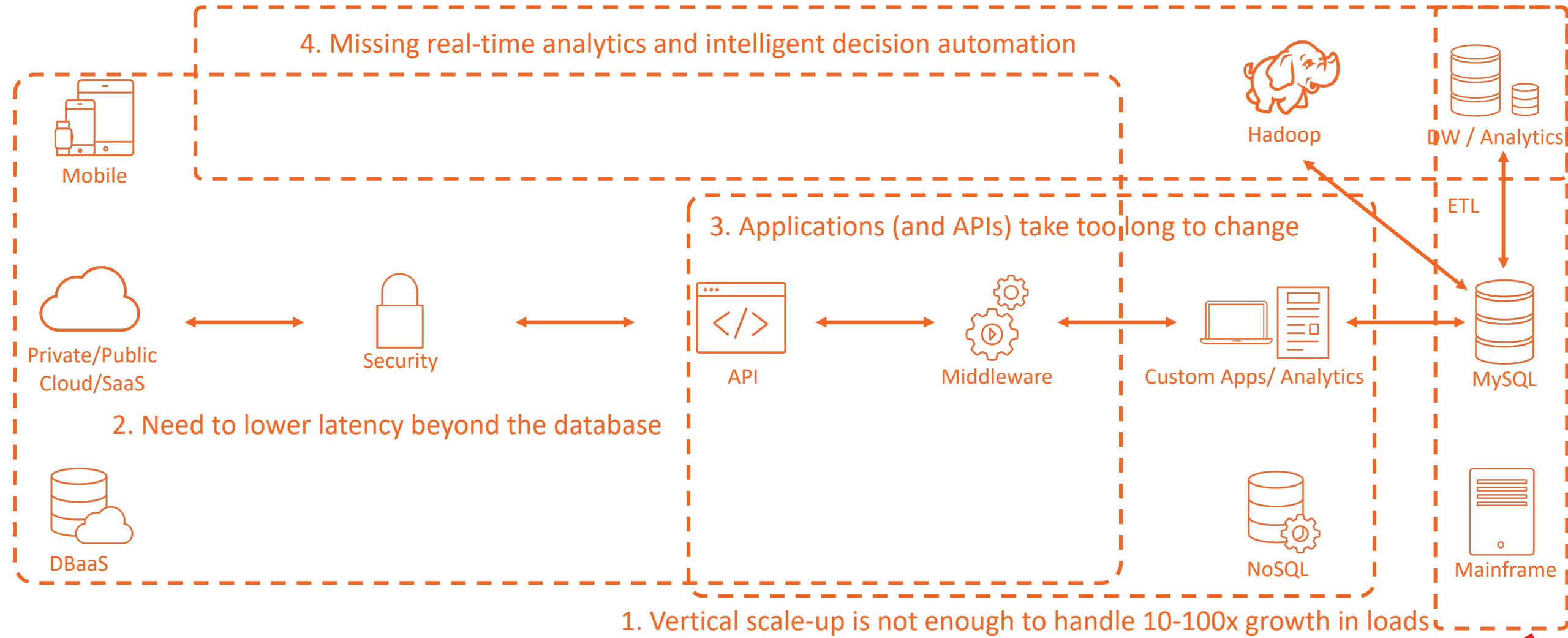
It's not just about the database



GridGain

Challenges with Speed and Scale

It's not just about the database

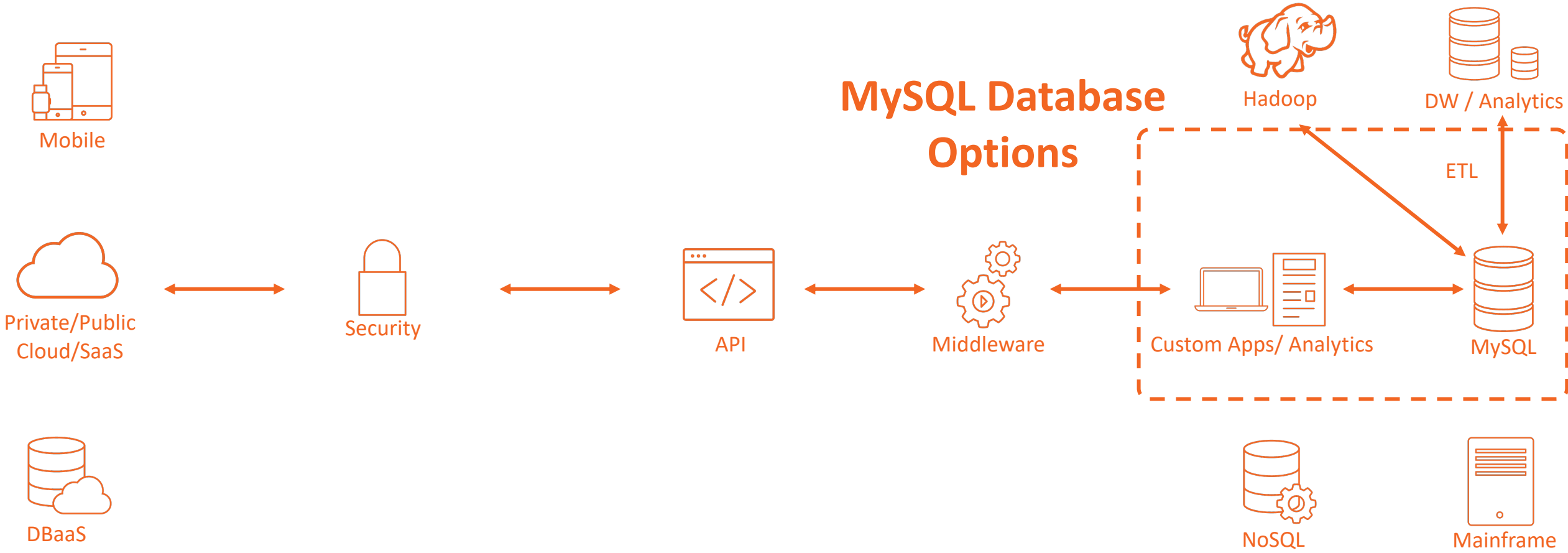


GridGain

Simple Rules for Success with Speed and Scale

1. Ask the “5 why’s” to find the speed and scale challenges
2. Optimize for end-to-end speed and scale
3. Look ahead over many projects
4. Choose the right long-term MySQL-related and 3rd party products

MySQL Database Options for Adding Speed and Scale



GridGain

MySQL and 3rd Party Options

- Options to Lower Latency
  - Memory tables
  - Memcached
  - Redis
- Options to Improve Scalability (OLTP)
  - MySQL InnoDB Cluster
  - MySQL NDB Cluster
  - Galera Cluster
  - MariaDB (with Galera Cluster)
  - Percona XtraDB with XtraDB Cluster
  - Vitess
  - Clustrix
- Honorable Mention: MySQL as a Service (cloud)

Memory Tables, Memcached and Redis
Can lower read latency but not at scale

- "Built-in" Options
  - Use a RAM disk for storage
  - Memory Tables: only recommended as temporary tables
    - Not persisted, don't support clusters or foreign keys
    - Don't support transactions (and MVCC), and only lock writes at table level
    - https://dev.mysql.com/doc/refman/8.0/en/memory-storage-engine.html
- Memcached
  - Widely used, well documented within MySQL docs
  - Cache-aside cache that lowers latency but must be managed by application
  - Doesn't help with scalability, so doesn't solve limited network bandwidth for large data sets
- Redis: Better than Memcached, But Still the Same Issues as a Cache
  - You can shard, but it becomes hard to manage at scale
  - Still have the network as a bottleneck

Hint of the Day

"With MySQL, Sharding Comes Last"

How to Scale MySQL Horizontally

- Replication for Read Scalability
  - Group Replication: sync/async replication
  - Manual failover when used with partitioning
  - No automatic recreation of secondary
- Sharding for Data Size and Write Scalability
  - A manual process
  - Hard to manage sharding to balance different workloads
- Cloud Offerings are One Option, But They’re the Same as-a-Service (Saas)
  - Amazon RDS and Aurora
  - Azure and Google offerings
- OLAP Offerings Exist as Well, But OLAP and OLTP Are Separate

Mysql InnoDB Cluster – Read/Write Replication

https://www.mysql.com/

- MySQL InnoDB Cluster ($5,000 per server for MySQL Enterprise Edition)
  - MySQL Router
  - MySQL Shell (cluster admin, uses AdminAPI)
  - MySQL Group Replication (rewritten in 2016)
  - MySQl InnoDB storage engine
- MySQL Group Replication
  - Single-master or multi-master replication
  - Uses Paxos consensus (majority), synchronous or asynchronous
  - Automatic failover and split brain prevention
- Challenges:
  - Limited scalability with ”shared-something” multi-master architecture
  - Optimistic transaction (multi-master) write conflict resolution
  - No partitioning: must be managed manually, and doesn’t work well with clustering
  - Flow control is managed by each node separately
  - Limited WAN support (expects low latency high throughput connections)

MySQL NDB Cluster – Read/Write Replication
https://www.mysql.com/

- MySQL NDB Cluster
  - Acquired by MySQL in 2003 (Alzato) for clustering
  - Separate NDB storage engine
  - 128TB storage limit per engine, and auto partitioning
  - Synchronous replication within cluster, asynchronous across clusters
  - Designed for 99.999% availability with <1 second failover/recovery times

- Challenges:
  - Price: $10,000 per server for MySQL Cluster CGE (vs $5,000 for Enterprise)
  - No MVCC, supports read committed transactions only

Galera Cluster – Read/Write Replication
http://galeracluster.com/

- Galera Cluster
  - For MySQL with InnoDB storage engine
  - Can use with ProxySQL, MaxScale or HAProxy routers
  - Multi-master replication
  - Uses optimistic certification-based replication (using a write set) with all nodes
  - Automatic failover and split brain prevention
  - Flow control is managed centrally
  - Does provide WAN support

- Challenges:
  - Limited scalability with ”shared-something” multi-master architecture
  - Optimistic transactions can require compensating transactions
  - No partitioning: must be managed manually, but will work well with clustering

MariaDB Cluster – Read/Write Replication
https://mariadb.com/

- MariaDB
  - Founded in 2010 the day Oracle completed the Sun (MySQL) acquisition
  - From the beginning focused on adding features (leads to some incompatibility)
  - One major feature is ColumnStore for better OLAP performance
  - Uses InnoDB storage engine (recently dropped XtraDB support due to shrinking differences)
  - Provides MaxScale for routing and CDC to ColumnStore
  - Includes its own version of Galera Cluster (and all the benefits of Galera Cluster)

- Challenges (same as Galera Cluster):
  - Limited scalability with ”shared-something” multi-master architecture
  - Optimistic transactions can require compensating transactions
  - No partitioning: must be managed manually, but will work well with clustering

Percona XtraDB Cluster – Read/Write Replication
https://www.percona.com/

- Percona: Open Source Expertise and Support for MySQL, MariaDB, PostgreSQL, Mongo, …
- Percona XtraDB and XtraDB Cluster
  - XtraDB first released in 2006: free (drop-in) version of MySQL Enterprise Edition
  - Implement some features (e.g. thread pools) and merges changes from different vendors to deliver performance, scalability, monitoring and management https://www.percona.com/software/mysql-database/percona-server/feature-comparison
  - Originally used Galera Cluster, then created its own version, XtraDB Cluster, based on Galera Cluster once it became open source
  - Includes ProxySQL as part of full clustering solution: solid performance, security, and GTID routing that keeps routing, which will work with partitioning
- Challenges (same as Galera Cluster):
  - Limited scalability with ”shared-something” multi-master architecture
  - Optimistic transactions only; does not support XA across the cluster and can require compensating transactions upon failure.
  - No partitioning: must be managed manually, but will work well with clustering (and ProxySQL)

Vitess – Distributed MySQL

https://vitess.io/ https://planetscale.com/

- Vitess: Database Clustering System for Horizontal Scaling of MySQL
  - Started in 2010 at YouTube to solve MySQL scalability, including sharding
  - Solid Kubernetes support: one of the first projects to adopt Kubernetes
  - PlanetScale founded in 2018 to provide commercial support
  - Became 16th Cloud Native Computing Foundation (CNCF) project in 2018
  - Automatic sharding and rebalancing transparent to application
  - Single master for each shard for true linear horizontal scale
  - Sharded materialized views (sharding using multiple keys)

- Challenges: still maturing as a technology
  - Does not support isolation for transactions, resulting in dirty reads
  - Early adopters (e.g. Slack) have mentioned it needed modifications for them
  - Lack of management features for managing each master (Orchestrator recommended)
  - No out-of-the-box monitoring, and no support yet by SeveralNines or others
  - Investigate FAQ for issues: https://vitess.io/docs/user-guides/faq/

Clustrix – Distributed MySQL
www.mariadb.com

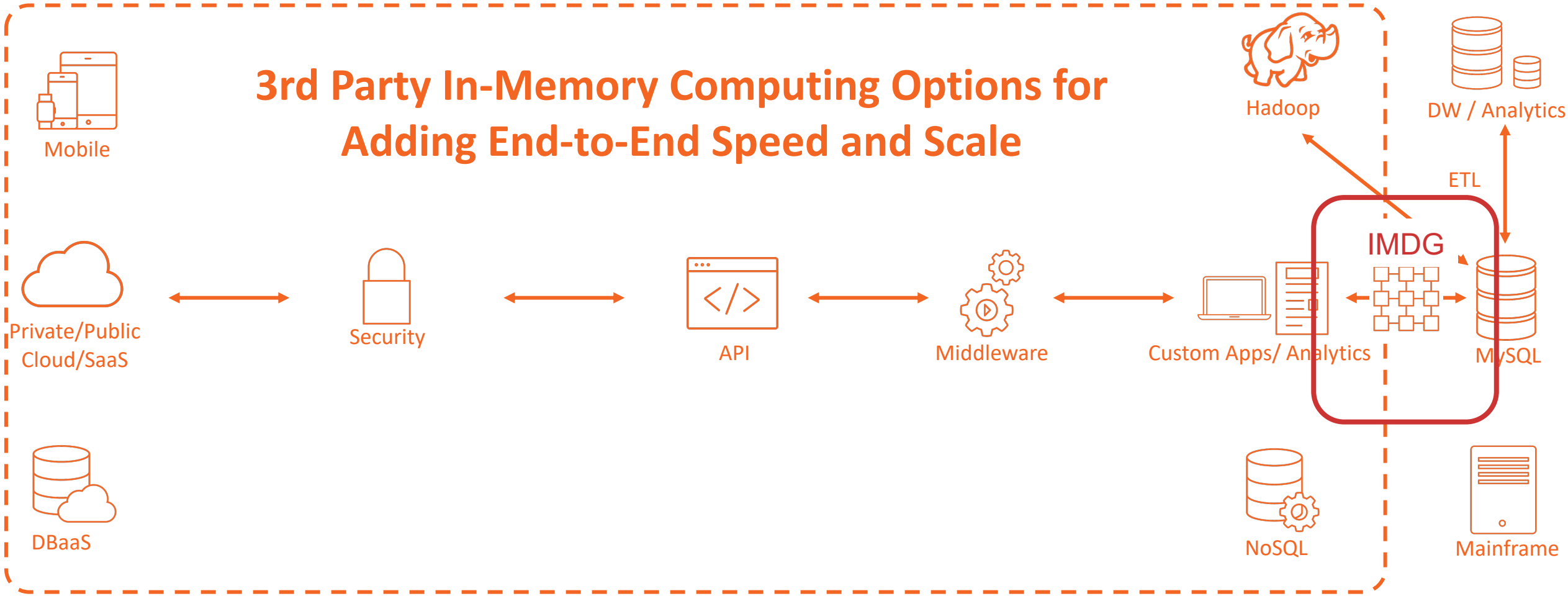
- Clustrix: Horizontally Scalable MySQL
  - Founded in 2006, acquired by MariaDB September 2018
  - Drop-in replacement for MySQL, but not the same MySQL codebase (C)
  - True horizontal linear scalability with automatic sharding, rebalancing
  - Not open source (licensed)
  - Enforces MVCC with read committed, repeatable read, pessimistic locking (2PC)
  - Uses serializable for synchronous replication in cluster, asynchronous across clusters

- Challenges:
  - Great for adding scalability, not for lowering latency

MySQL as a Service – Cloud Offerings
AWS, Azure and Google Cloud Platform

- Overall, if you are moving to the cloud, any one of the cloud vendors will provide good MySQL as a service with simplified management and read/write scalability, but no partitioning.
- Amazon RDS gives you choice: Amazon RDS for MySQL, Amazon Aurora
  - Aurora performs better (mostly) for a 20% premium
- Still have the same limitations as MySQL

3rd Party Options for Adding Speed and Scale



GridGain

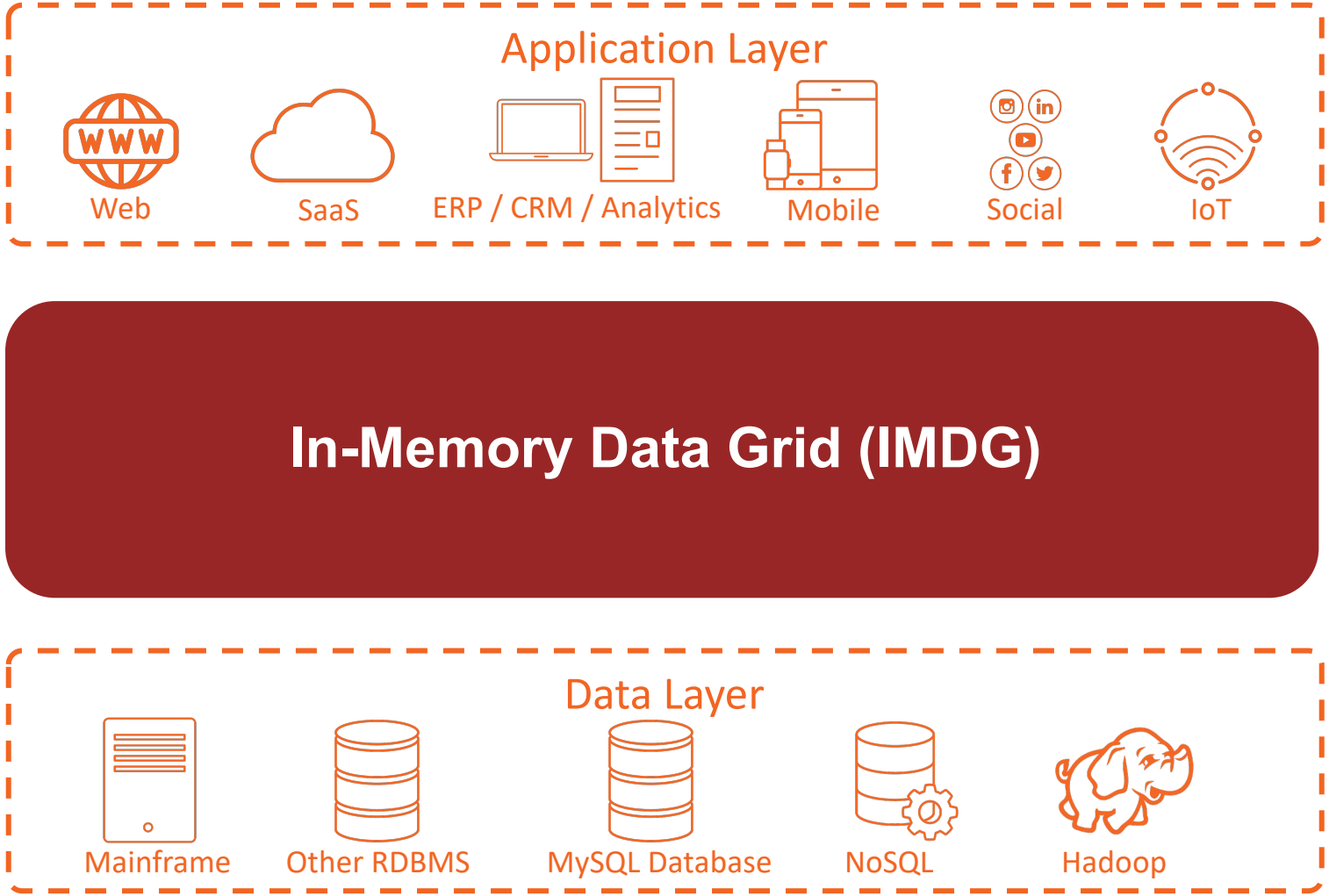
How an In-Memory Data Grid Works

How an IMDG Works

- Slides in-between MySQL Database and the app / analytics
- Acts as the new database

Benefits

- No rip-and-replace of MySQL
- In-memory speed
- Horizontal scalability (partitioning)
- Collocated computing
- Merges data across sources
- Unlocks data for new applications



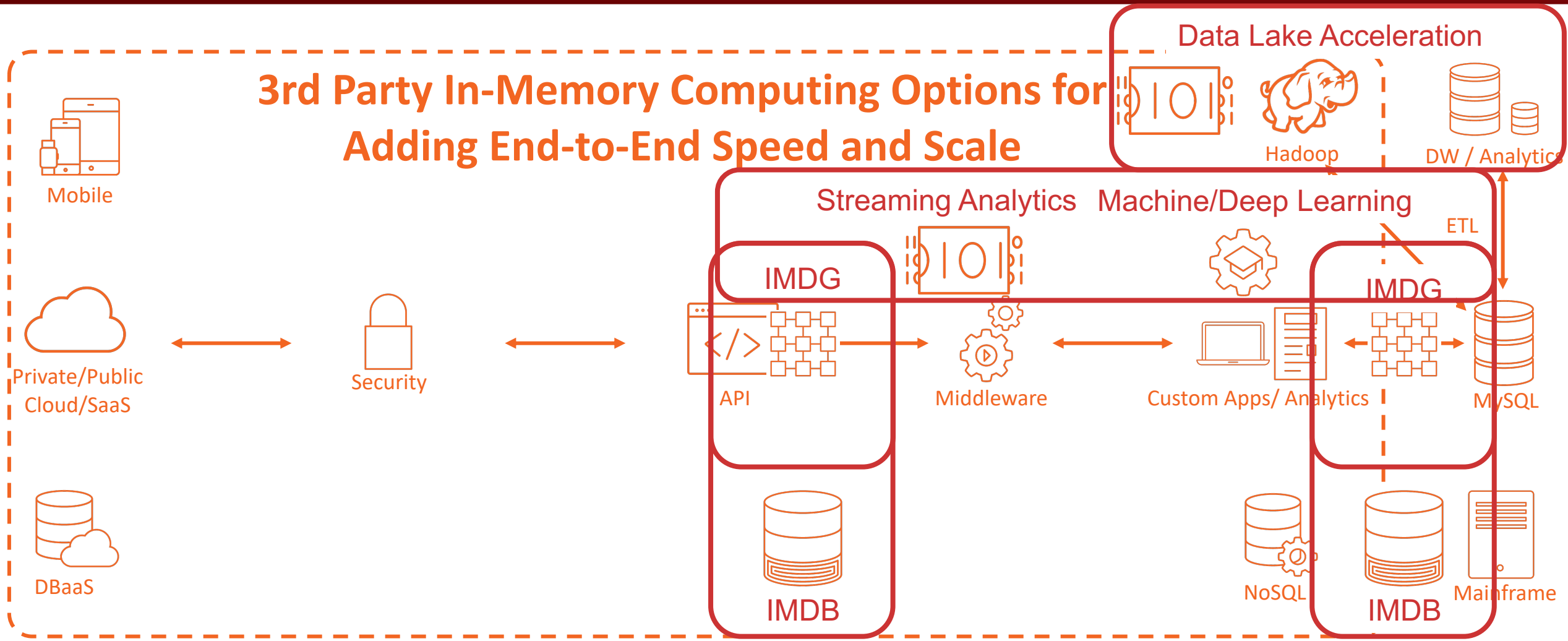
GridGain

In-Memory Computing Comparisons
https://www.gridgain.com/resources/product-comparisons

Vendors Compared

- Oracle Coherence
- Pivotal Gemfire (Apache Geode)
- GigaSpaces
- GridGain (Apache Ignite)
- Hazelcast
- Redis (Cache)
- Terracotta

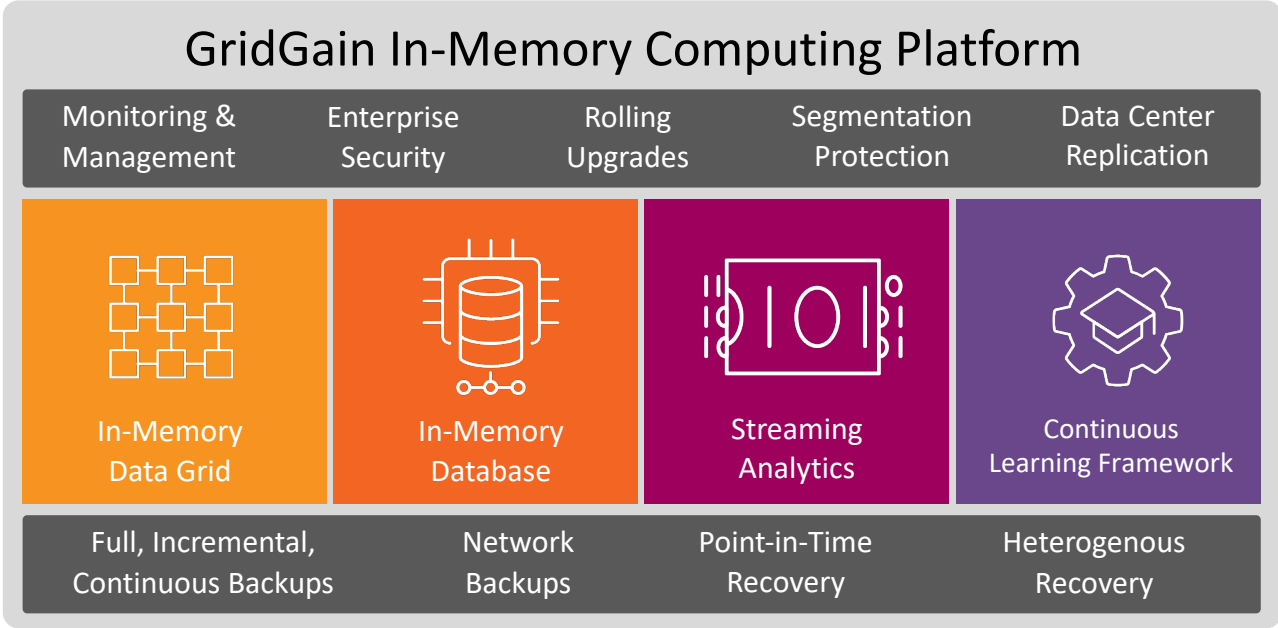
How In-Memory Computing is Used



GridGain

GridGain In-Memory Computing Platform

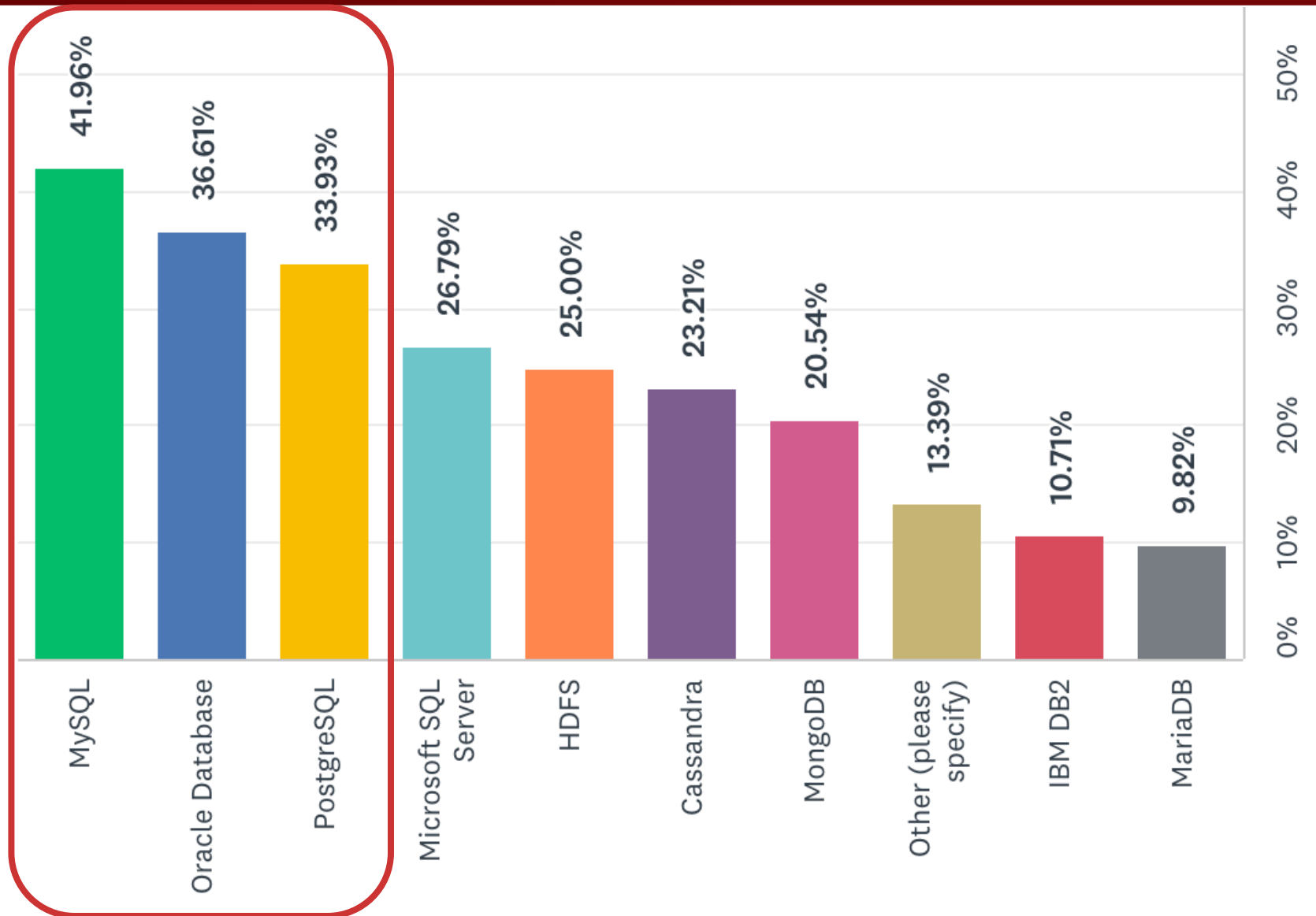
- Built on Apache Ignite
  - Comprehensive platform that supports all projects
  - No rip and replace
  - In-memory **speed**, petabyte **scale**
  - Enables HTAP, streaming analytics and continuous learning
- What GridGain adds
  - Production-ready releases
  - Enterprise-grade integration, security, deployment and management
  - Global support and services
  - Proven for mission critical apps



GridGain

Ignite Is Used The Most with MySQL

- Over 40% of Ignite Users leverage Ignite to Add Speed and Scale to MySQL
- 66% Use or Plan to Use Ignite with 2 or More Databases (40% 3 or More)

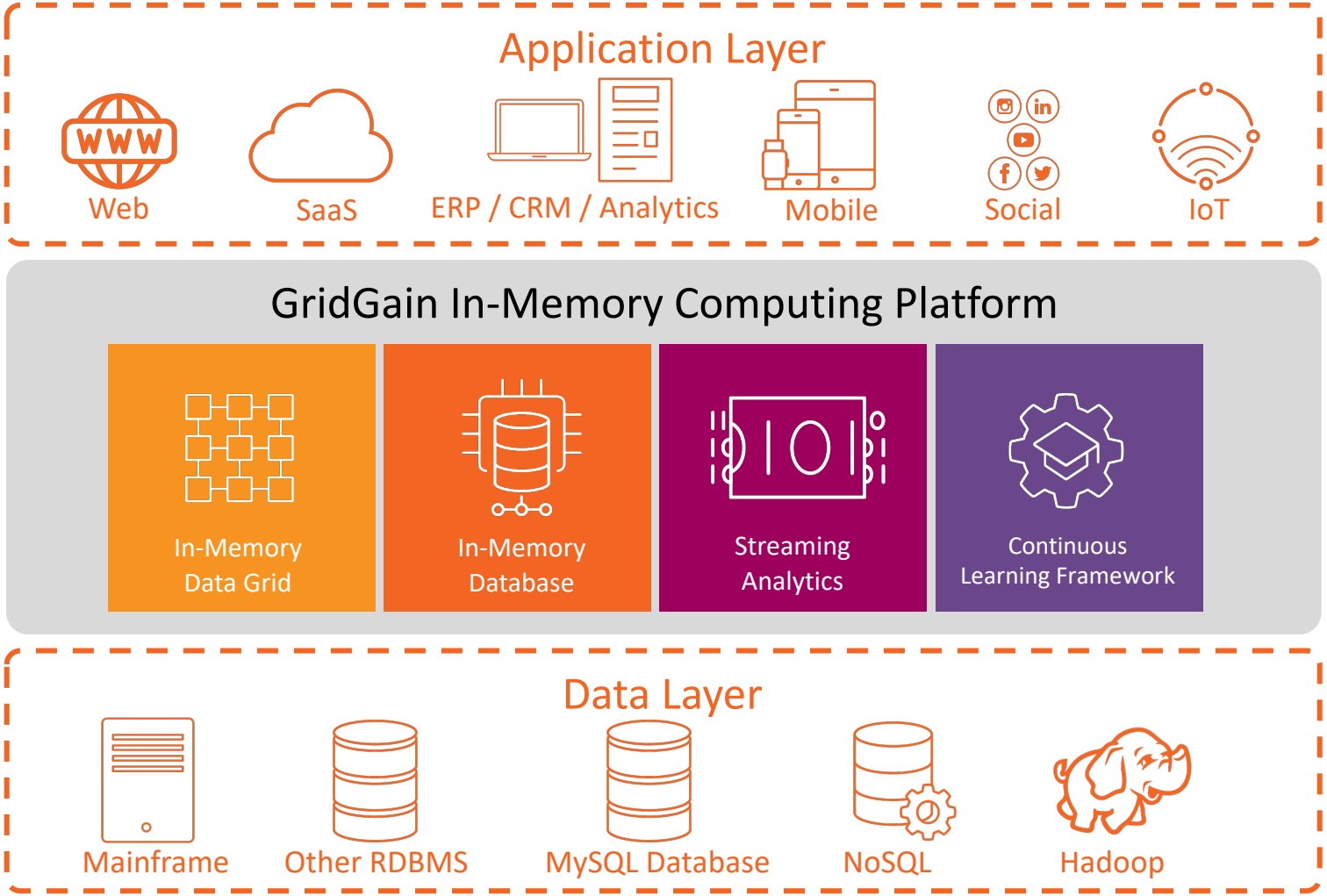


Source: 2018 In-Memory Computing Survey
(Completed and downloaded Ignite/GridGain N=112)

Why an In-Memory Computing Platform is Needed

A Common Data Layer for Real-time Business

- Add speed and scale to existing applications
- Store new types of (big) data
- Ingest and process streaming data with Apache Spark and other streaming analytics technologies to support real-time analytics
- Implement real-time decision automation including continuous machine and deep learning



GridGain

Case Study: ING
Next Generation Banking

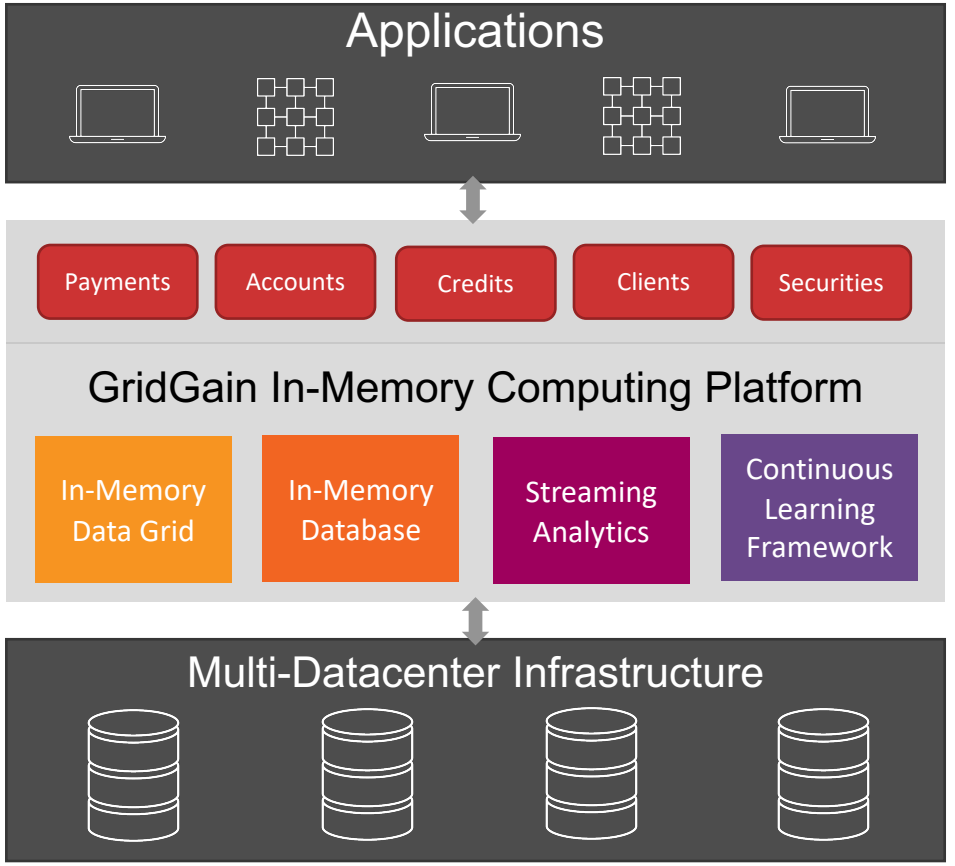
A Dutch multinational financial services firm with primary businesses in retail banking, direct banking, commercial banking, investment banking, asset management, and insurance services

Challenges

- Deliver new competitive customer services fast
- The high cost of running on mainframe infrastructure
- Transactional consistency over multiple geo-locations

GridGain Enables

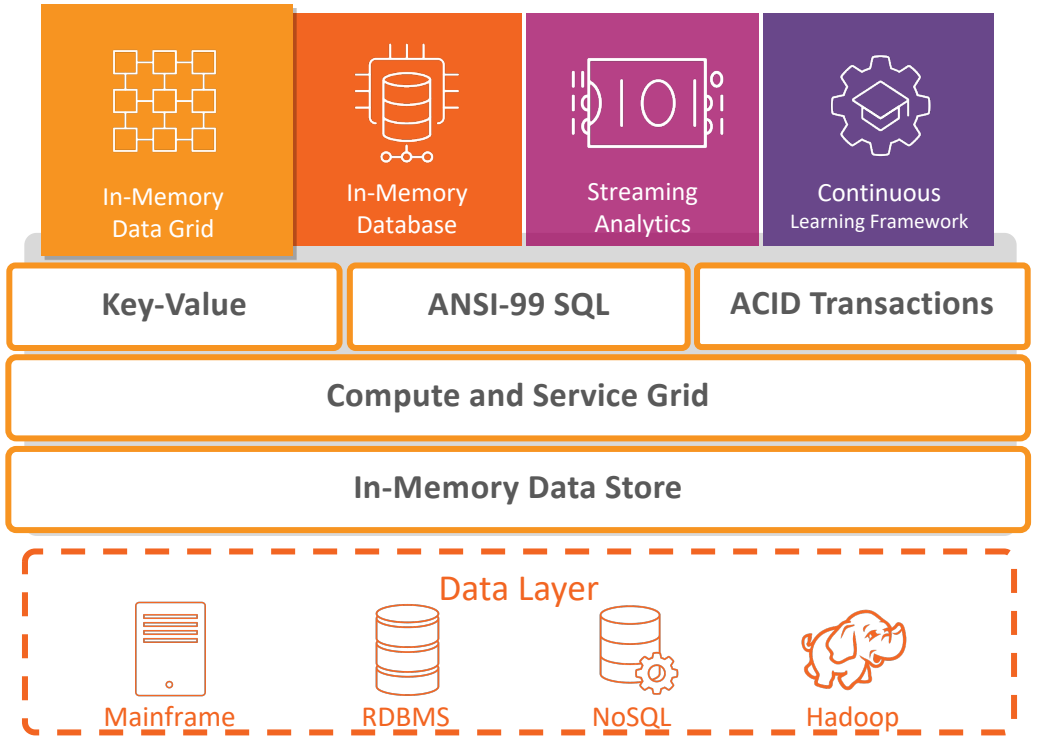
- Rapid delivery of new services that helped ING be first to market for PSD2, SEPA and STET
- Sub-100ms latency for APIs that aggregate data across multiple sources
- Reduced infrastructure costs while supporting 25% annual growth of mobile usage



Accelerate Existing Applications with No Rip and Replace

GridGain as an In-Memory Data Grid (IMDG)

- Slides in-between apps and RDBMSs with no rip and replace
  - ANSI-99 SQL compliant
  - Support for ACID transactions
- Accelerates existing app performance
- Offload new data and computing requirements (real-time auditing and compliance, analytics, computations)

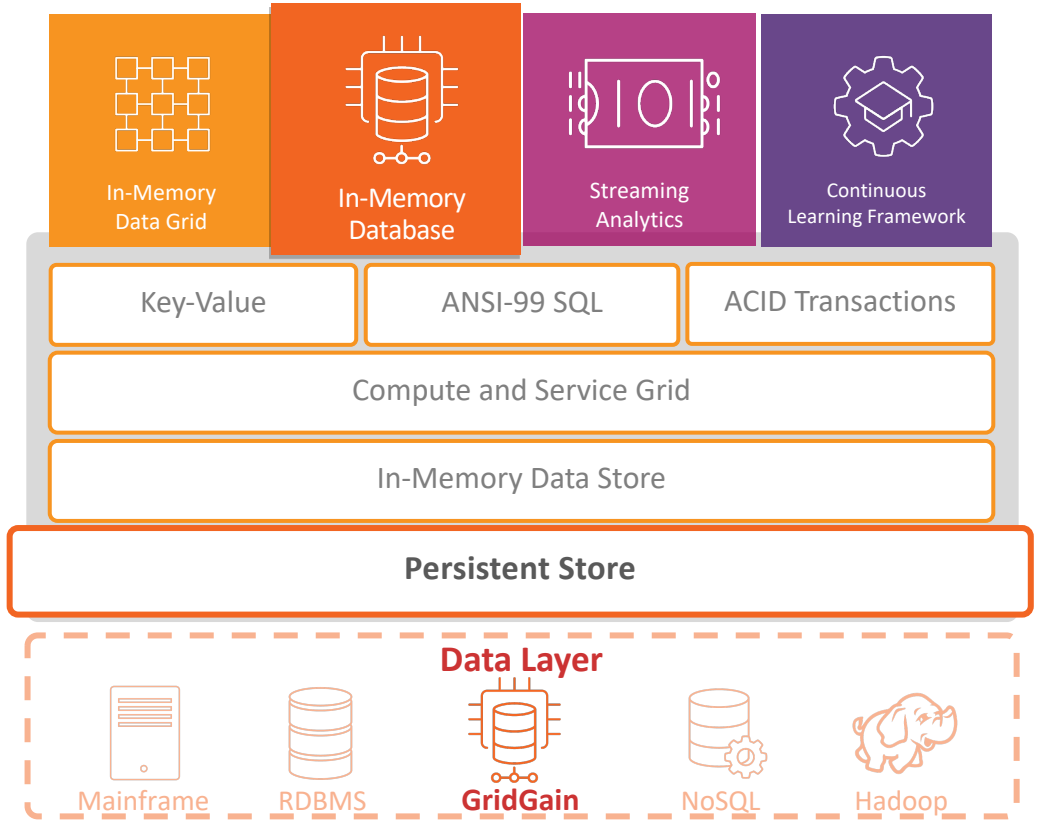


GridGain

Innovate with Existing and New Data

GridGain as an In-Memory Database (IMDB)

- **Memory-centric** storage
  - From 100% in-memory to 100% disk
  - Leverages any combination of RAM, Flash, SSD, Intel 3D Xpoint and disk
  - Low cost, disk-based reliable persistence
  - Immediate restart during recovery
- Highest **read+write** performance
  - In-memory with unlimited linear, scale-out on commodity servers
  - SQL and NoSQL (multi-model)
  - Always-on availability
- Single data access layer for **ALL** data
- Extensible compute grid

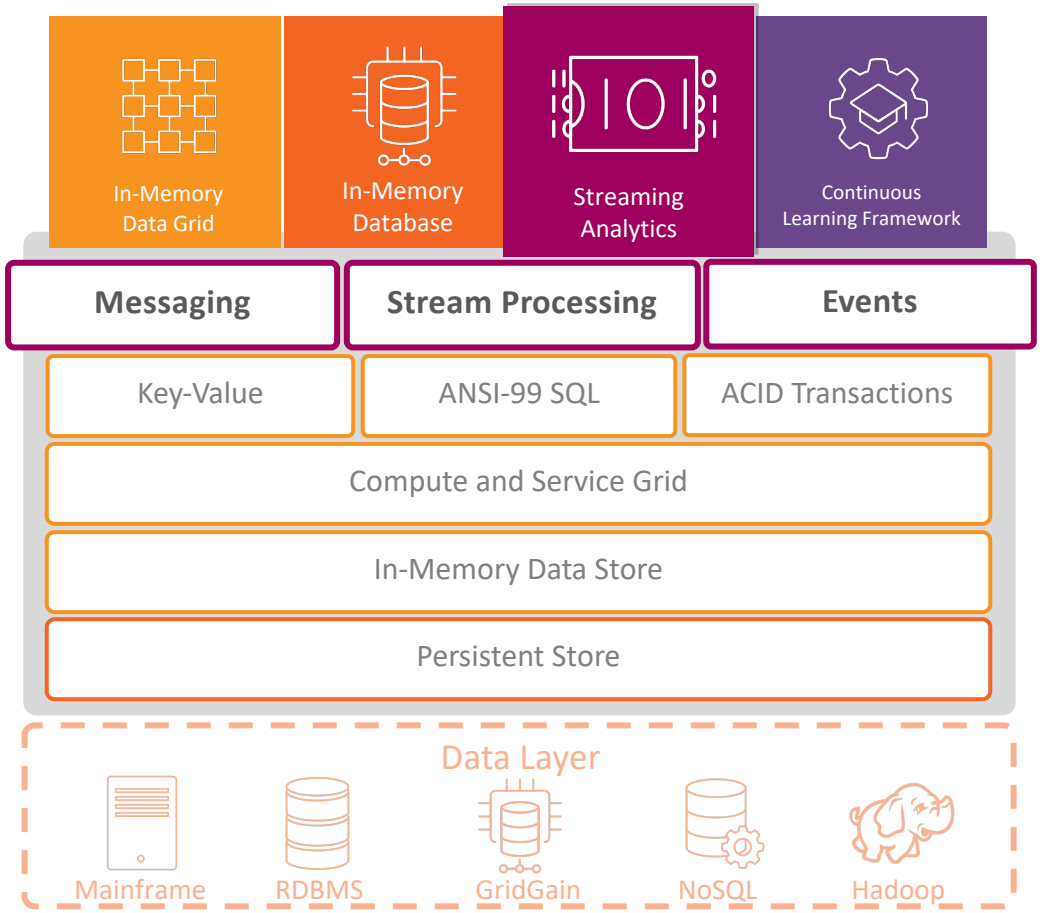


GridGain

Innovate with Streaming Analytics

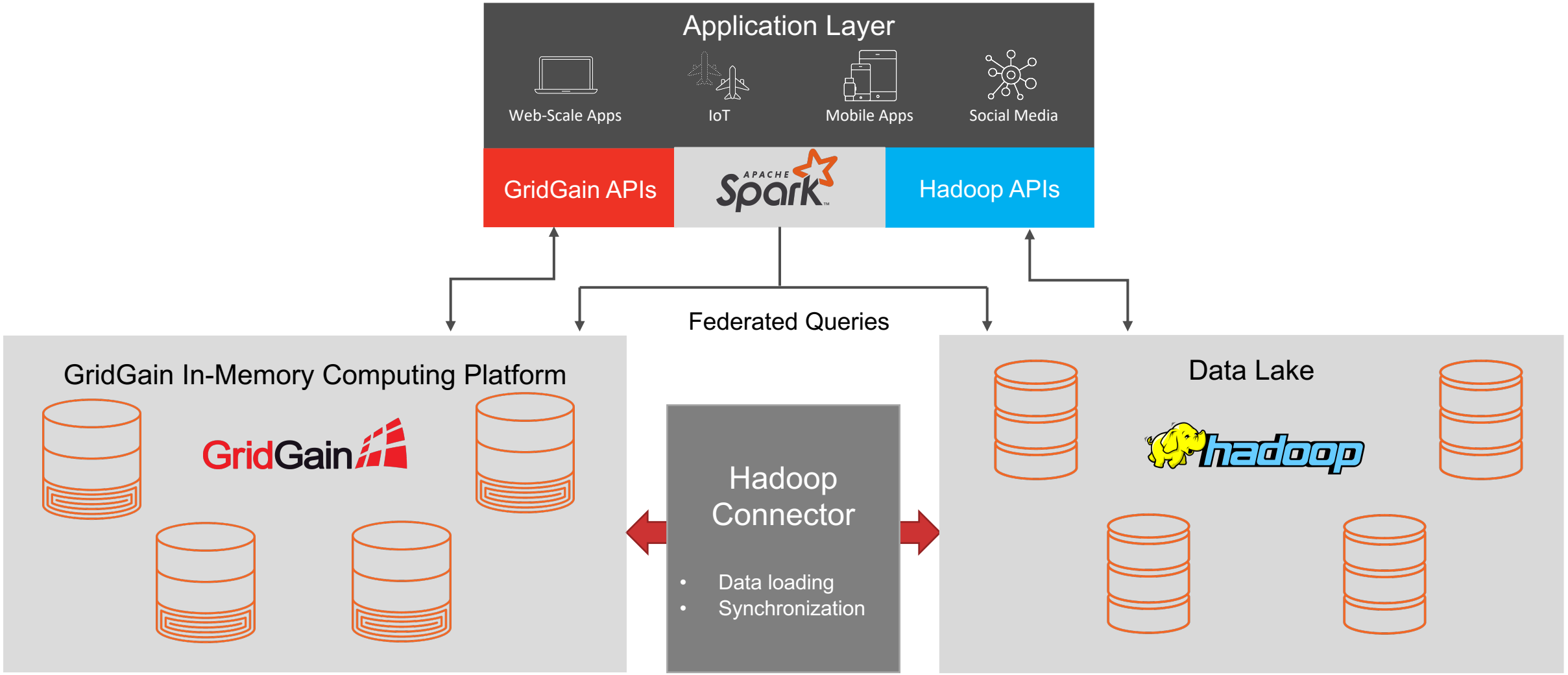
GridGain for Stream Ingestion, Processing and Analytics

- Native support for stream ingestion
  - Built-in support for high speed ingestion from Apache Camel, Flink, Flume, Spark, Storm, JMS, Kafka and MQTT
  - Combines streams with data-at-rest
  - Collocated data processing across all data, including optimized SQL querying
  - Publish/subscribe (continuous queries)
- Broadest in-memory support for Apache Spark
  - Native in-memory RDD, DataFrame support
  - Shares state in memory across Spark jobs
  - Native access to ANY data across GridGain cluster
  - Optimizes SparkSQL using distributed SQL and indexing



GridGain

Leverage Historical and Real-Time Data Together

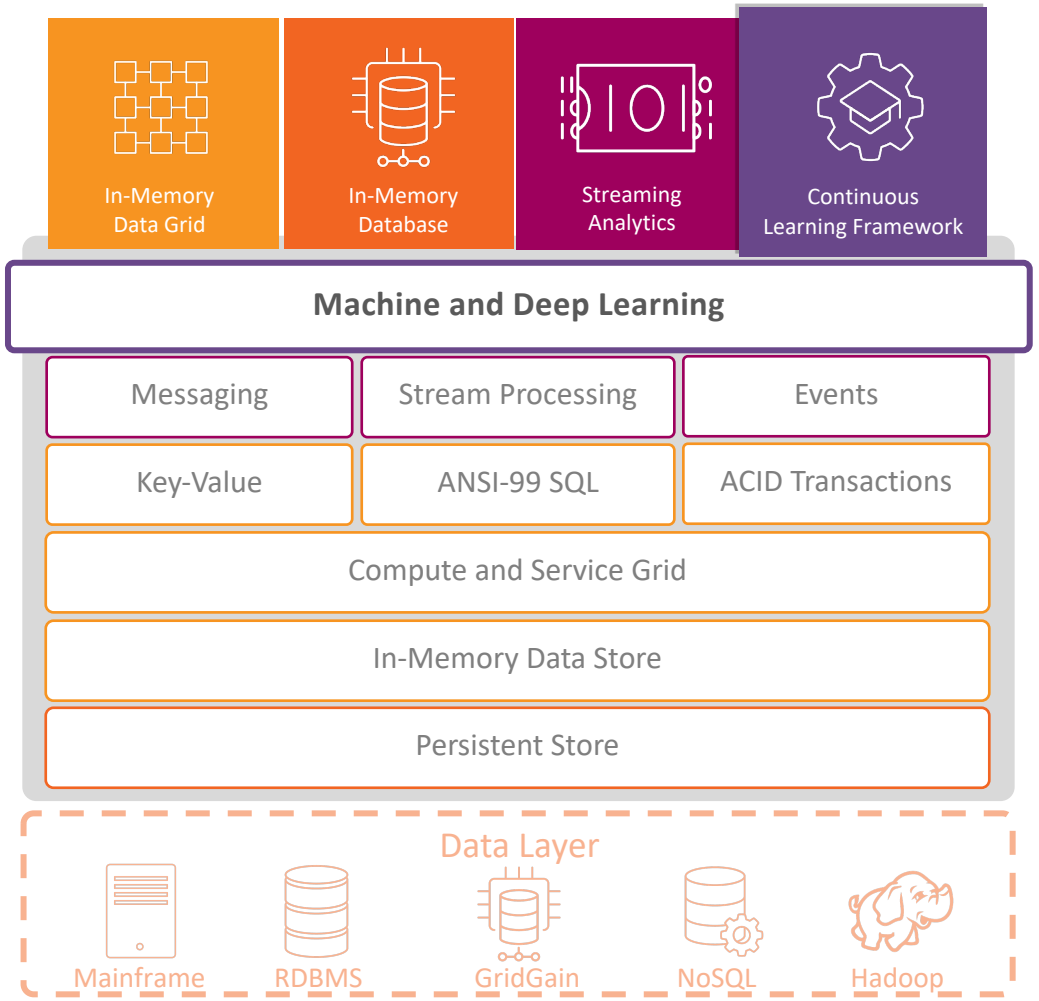


GridGain

Innovate with Continuous Learning

Continuous Learning Framework for Machine and Deep Learning

- Real-time performance on petabytes of data
  - No ETL (runs learning in place)
  - In-memory performance
  - Horizontal, linear scalability
- Machine learning
  - Linear, multi-linear regression
  - K-means clustering
  - Decision trees
  - K-NN classification and regression
- Deep Learning
  - TensorFlow integration



GridGain

Questions?

Driving In-Memory Computing Thought Leadership

GridGain produces the world's only In-Memory Computing Conference

https://www.imcsummit.org/

In-Memory Computing Summit **Europe**

- Just Happened in London, June 3-4 2019

In-Memory Computing Summit **North America**

- Next event in Silicon Valley, Nov 13-14 2019

GridGain

GridGain Resources
https://www.gridgain.com/

Ignite Resources
https://ignite.apache.org/

- Webinars
  - GridGain: https://www.gridgain.com/resources/webinars
  - In-Memory Computing Summit: https://www.imcsummit.org/
- White Papers: https://www.gridgain.com/resources/papers
- Videos: https://www.gridgain.com/resources/videos
- Blogs: https://www.gridgain.com/resources/blog
- Downloads
  - Apache Ignite: https://ignite.apache.org/download.cgi
  - Free GridGain Community Edition, or free 30-Day Enterprise or Ultimate Edition Trial
    https://www.gridgain.com/resources/download

GridGain

GridGain

Adding Speed and Scale to MySQL Database Deployments

Rob Meyer
Outbound Product Management, GridGain

June 26, 2019