



# Apache Ignite™ (Incubating) - In-Memory Data Fabric

Fast Data Meets Open Source

**DMITRIY SETRAKYAN**

Founder, PPMC

<http://www.ignite.incubator.apache.org>



@apacheignite

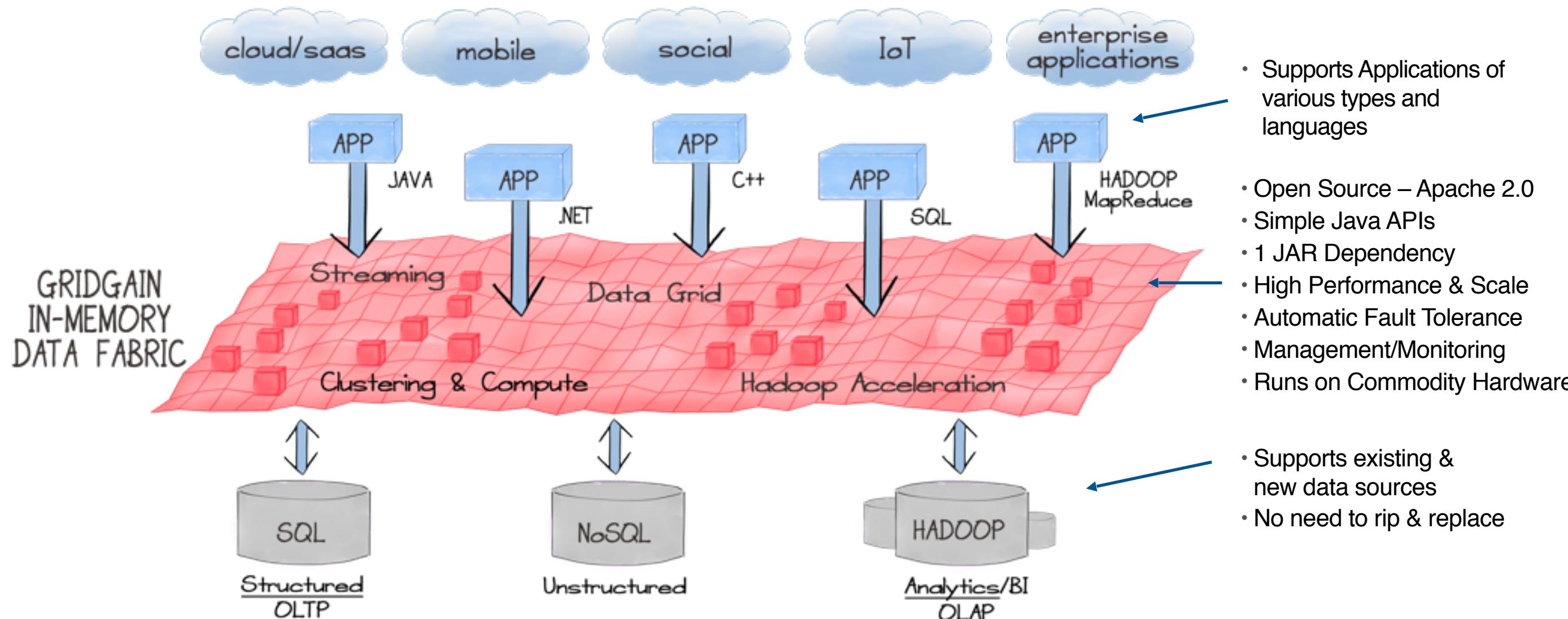


@dsetrakyan

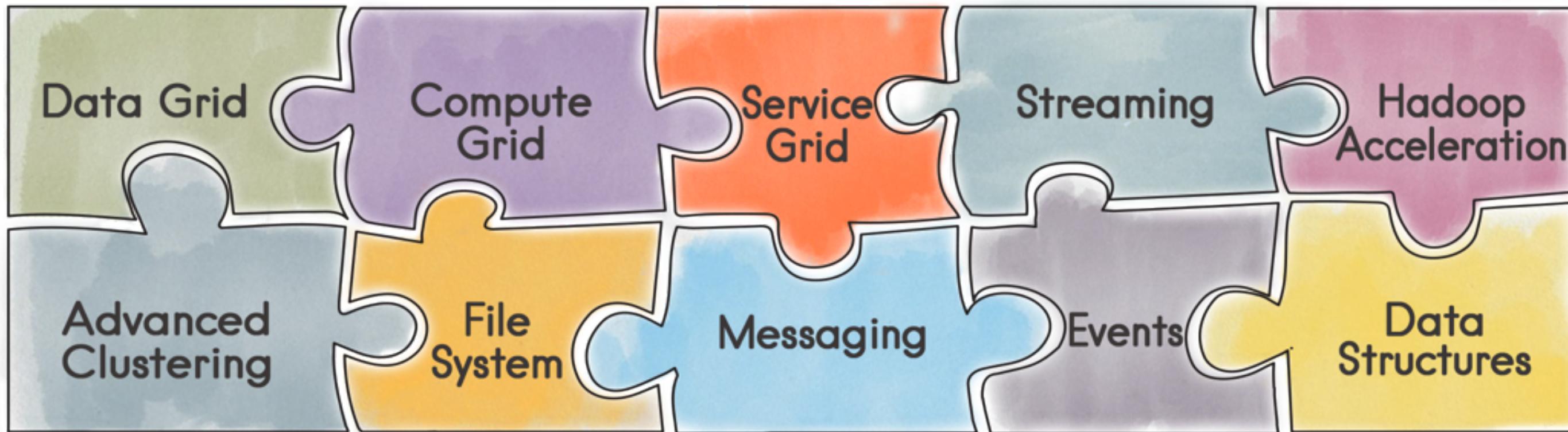
# Agenda

- About In-Memory Computing
- Apache Ignite™ In-Memory Data Fabric
  - Advanced Clustering
  - Data Grid
  - Compute Grid
  - Service Grid
- Ignite For Analytics
  - Streaming & CEP
  - Share State Across Spark Jobs
  - In-Memory MapReduce
  - Interactive SQL
  - DevOps: Yarn and Mesos
- Q & A

# Apache Ignite™ In-Memory Data Fabric: Strategic Approach to IMC

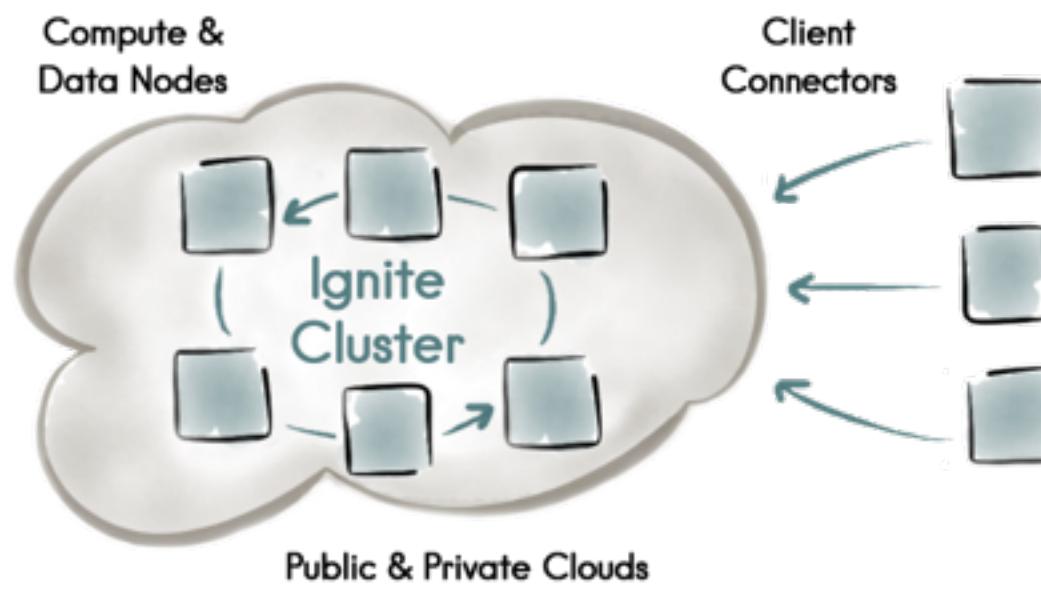


# In-Memory Data Fabric: More Than Data Grid



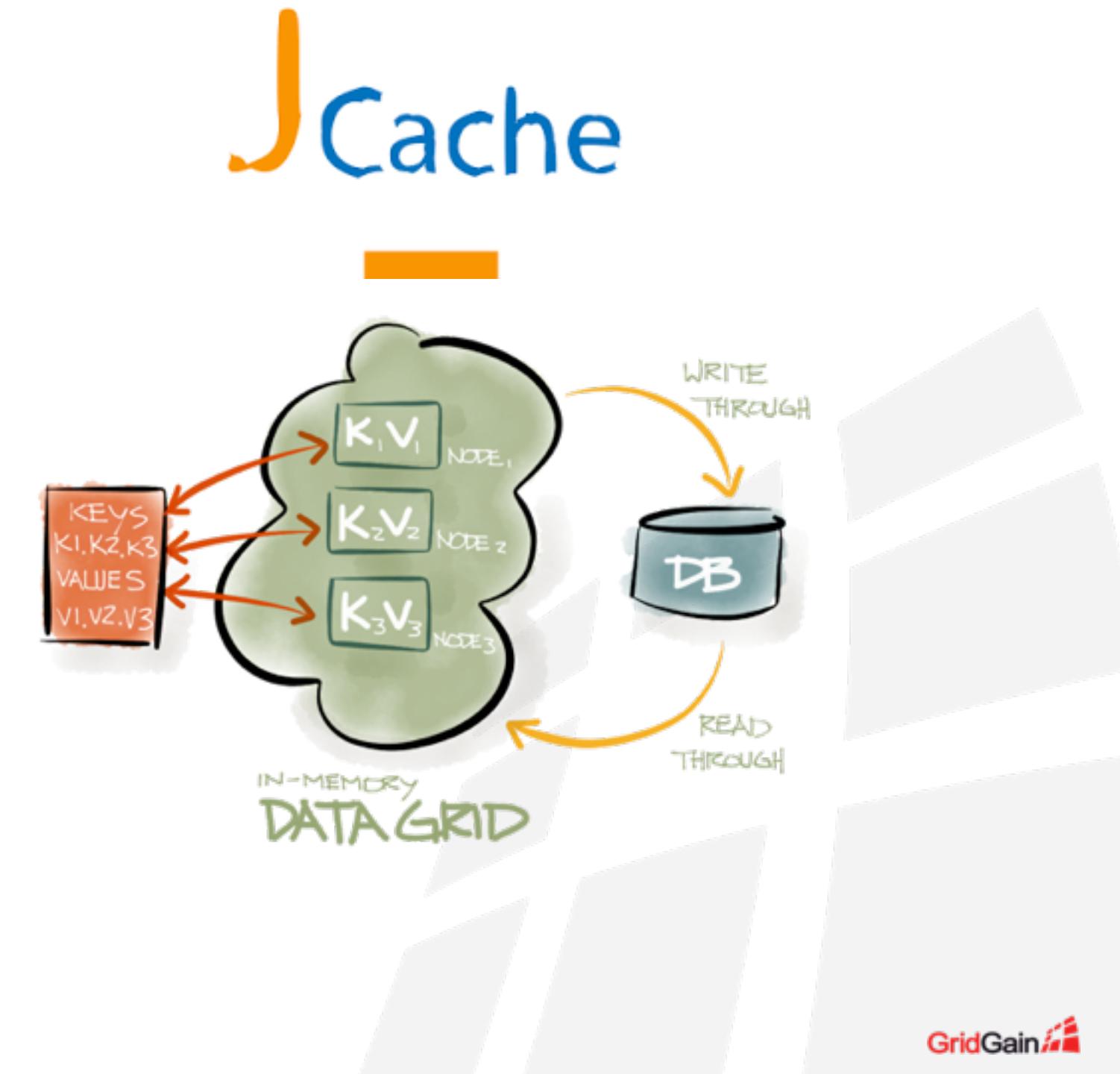
# Apache Ignite: Better Cloud Support

- Automatic Discovery
  - Simple Configuration
  - AWS/EC2/S3
  - Google Compute Engine (NEW)
  - Other Clouds with JClouds (NEW)
- Docker Support
  - Automatically Build and Deploy

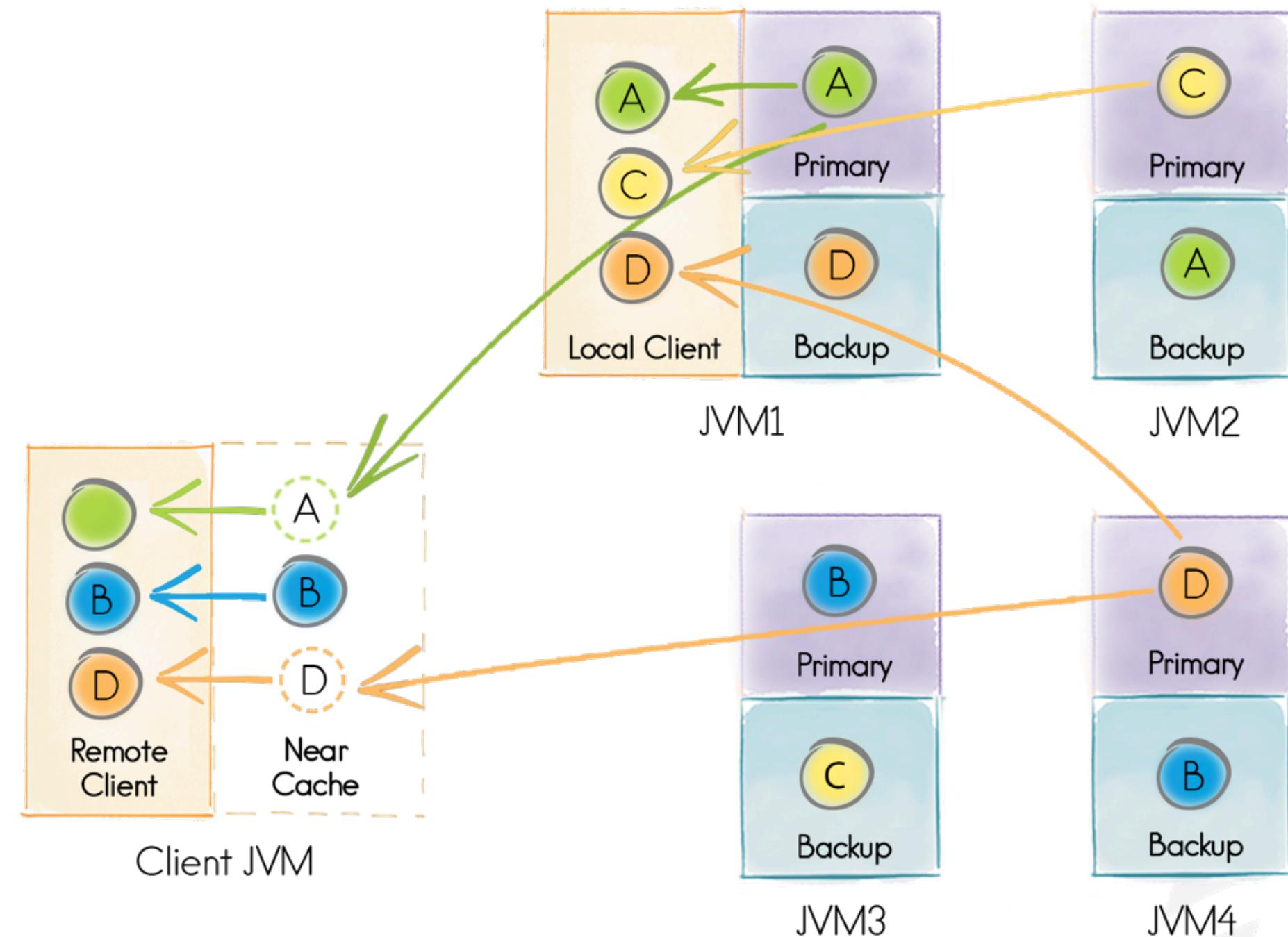


# Data Grid: JCache (JSR 107)

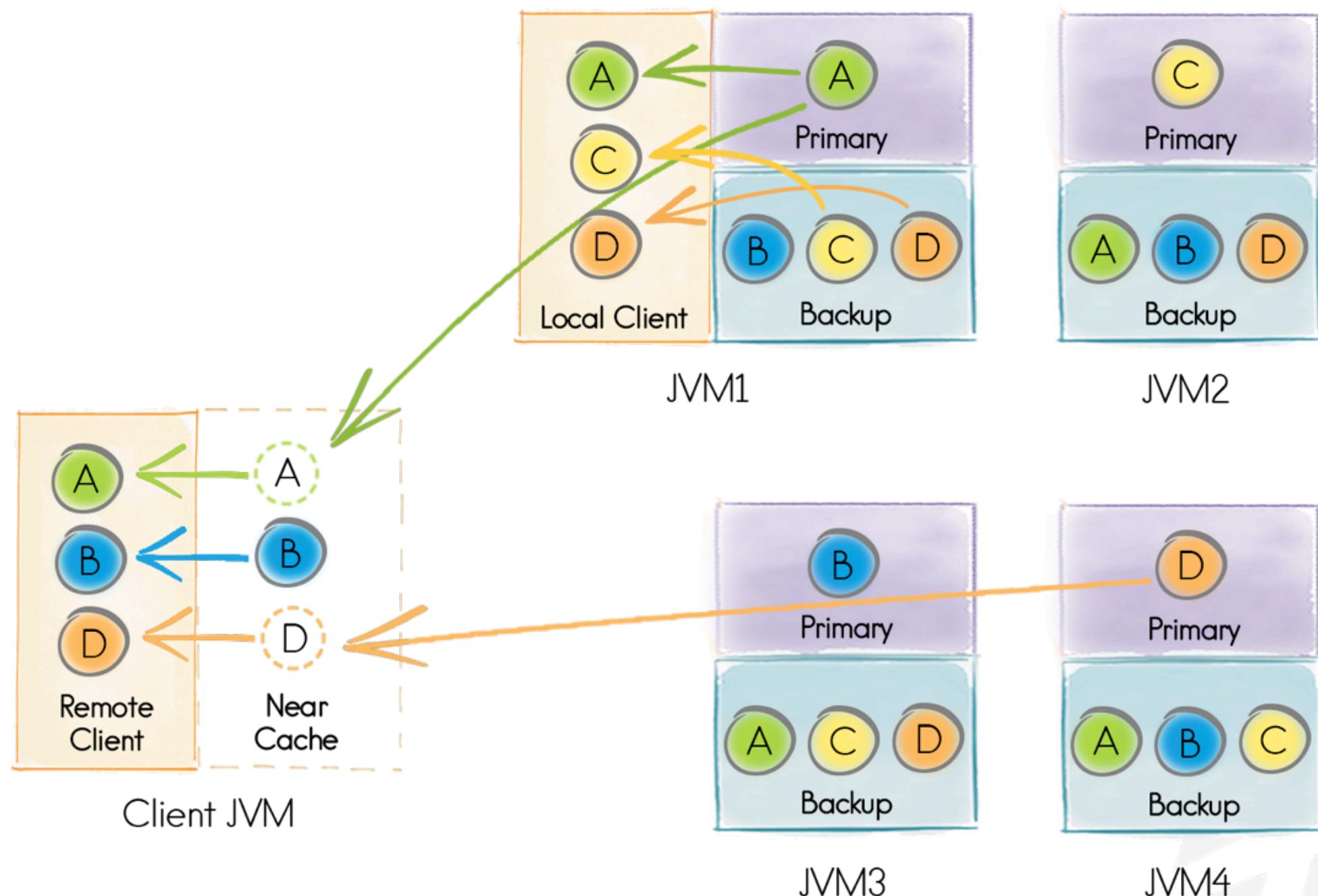
- JCache (JSR 107)
  - Basic Cache Operations
  - ConcurrentMap APIs
  - Collocated Processing (EntryProcessor)
  - Events and Metrics
  - Pluggable Persistence
- Ignite Data Grid
  - ACID Transactions
  - SQL Queries (ANSI 99)
  - In-Memory Indexes
  - Automatic RDBMS Integration



# Data Grid: Partitioned Cache

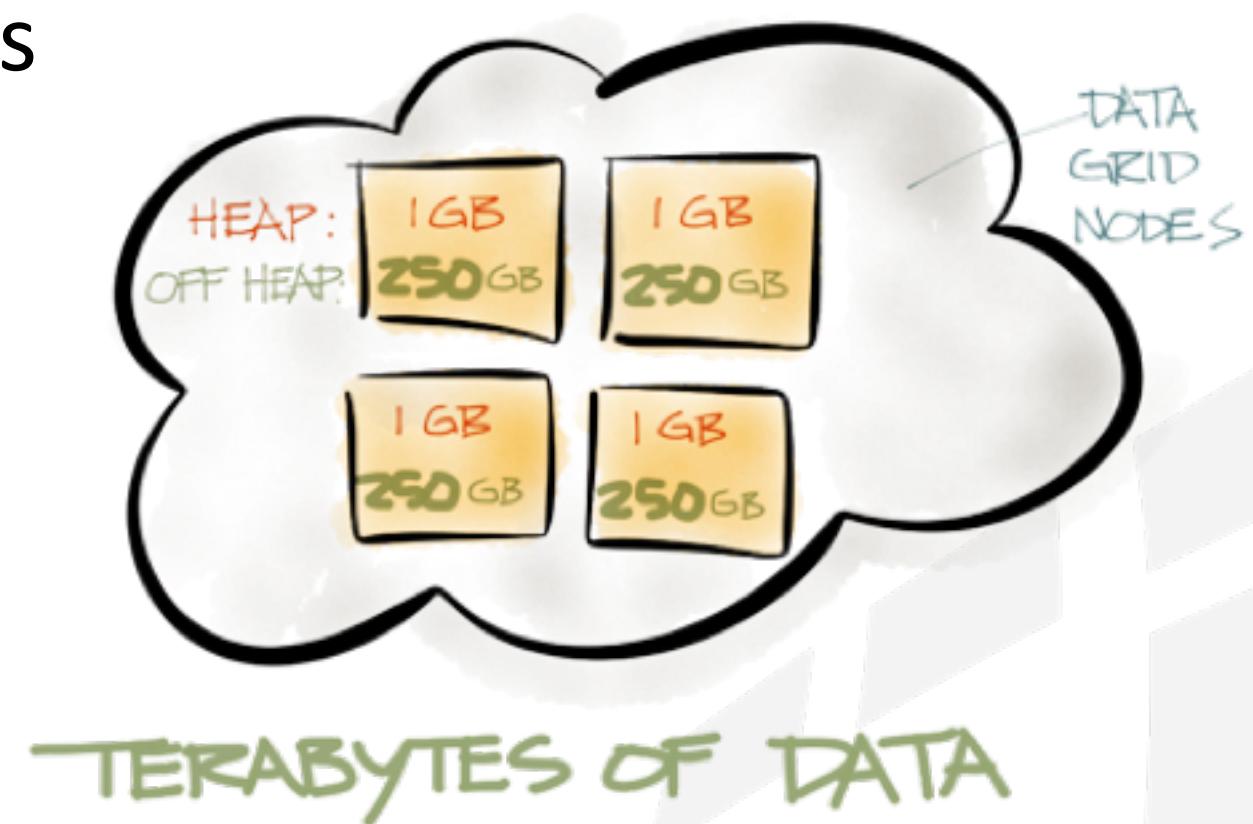


# Data Grid: Replicated Cache



# Data Grid: Off-Heap Memory

- Unlimited Vertical Scale
- Avoid Java Garbage Collection Pauses
- Small On-Heap Footprint
- Large Off-Heap Footprint
- Off-Heap Indexes
- Full RAM Utilization
- Simple Configuration



# Data Grid: Ad-Hoc SQL (ANSI 99)

- ANSI-99 SQL
- Always Consistent
- Fault Tolerant
- In-Memory Indexes (On-Heap and Off-Heap)
- Automatic Group By, Aggregations, Sorting
- Cross-Cache Joins, Unions, etc.
- Ad-Hoc SQL Support



# SQL Cross-Cache JOIN Example

```
IgniteCache<AffinityKey<UUID>, Person> cache = ignite.cache("persons");

// Execute query to get names of all employees.
SqlFieldsQuery qry = new SqlFieldsQuery(
    "select concat(firstName, ' ', lastName), org.name " +
    "from Person, \"Organizations\".Organization as org " +
    "where Person.orgId = org.id");

QueryCursor<List<?>> cursor = cache.query(qry);

for (List<?> row : cursor)
    print(row);
```

# SQL Cross-Cache GROUP BY Example

```
IgniteCache<AffinityKey<UUID>, Person> cache = ignite.cache("persons");

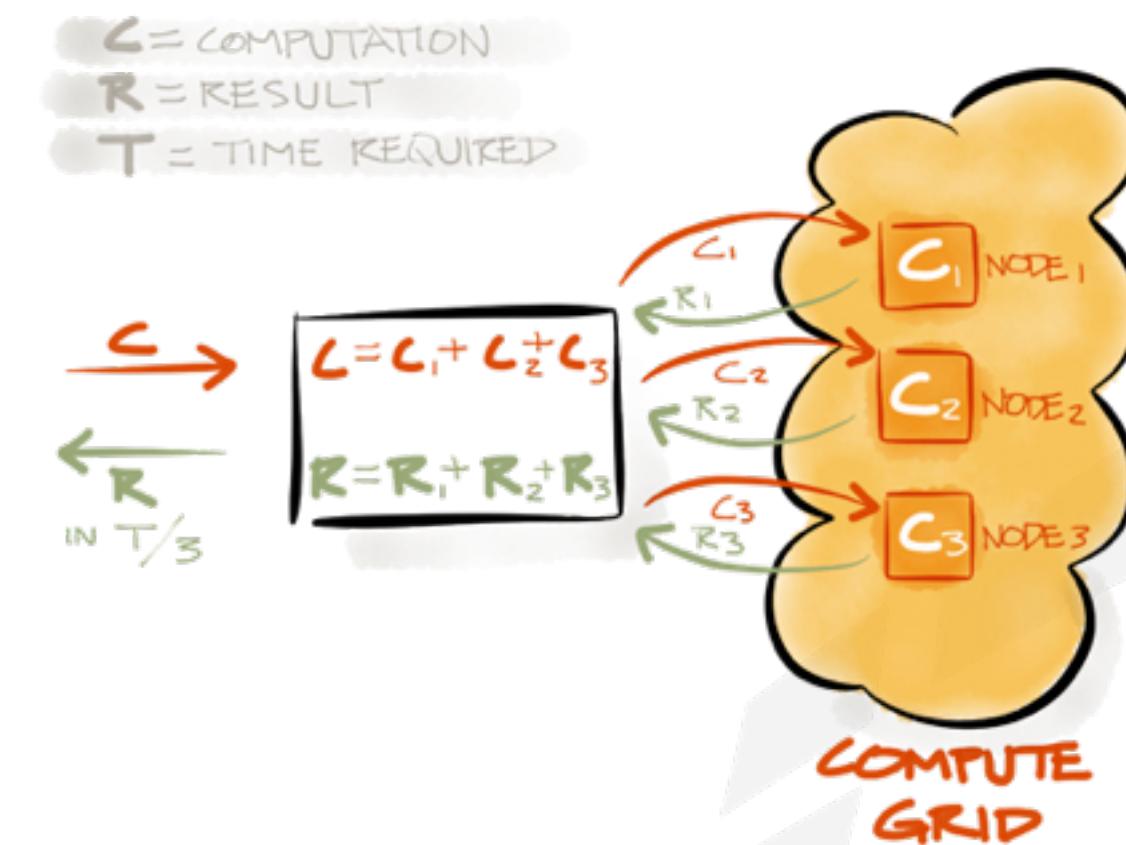
// Query to get salaries grouped by organization.
SqlFieldsQuery qry = new SqlFieldsQuery(
    "select org.name, avg(salary), max(salary), min(salary) " +
    "from Person, \"Organizations\".Organization as org " +
    "where Person.orgId = org.id " +
    "group by org.name " +
    "order by org.name");

QueryCursor<List<?>> cursor = cache.query(qry);

List<List<?>> res = cursor.getAll();
```

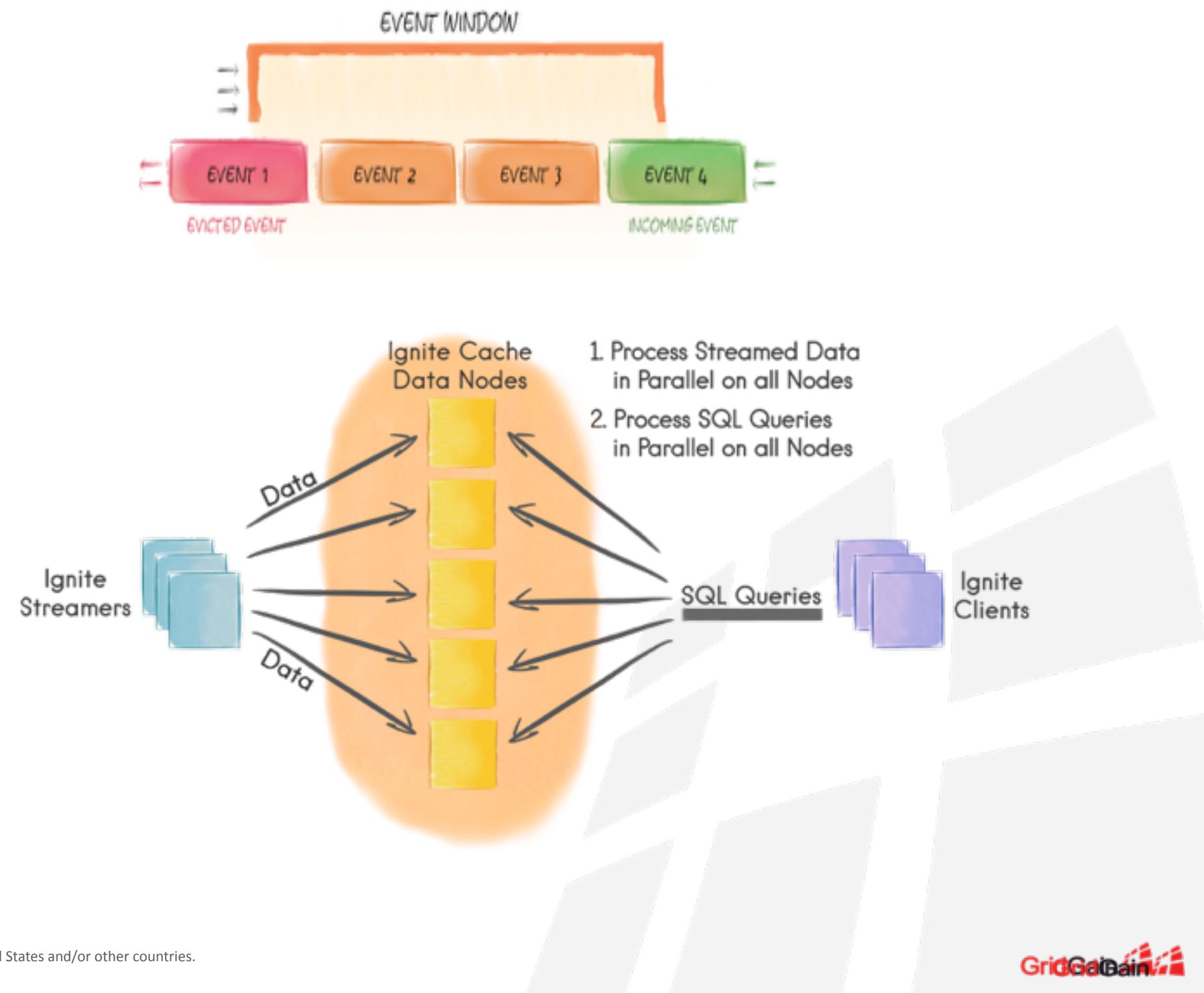
# In-Memory Compute Grid

- Direct API for MapReduce
- Direct API for ForkJoin
- Zero Deployment
- Cron-like Task Scheduling
- State Checkpoints
- Load Balancing
- Automatic Failover
- Full Cluster Management
- Pluggable SPI Design



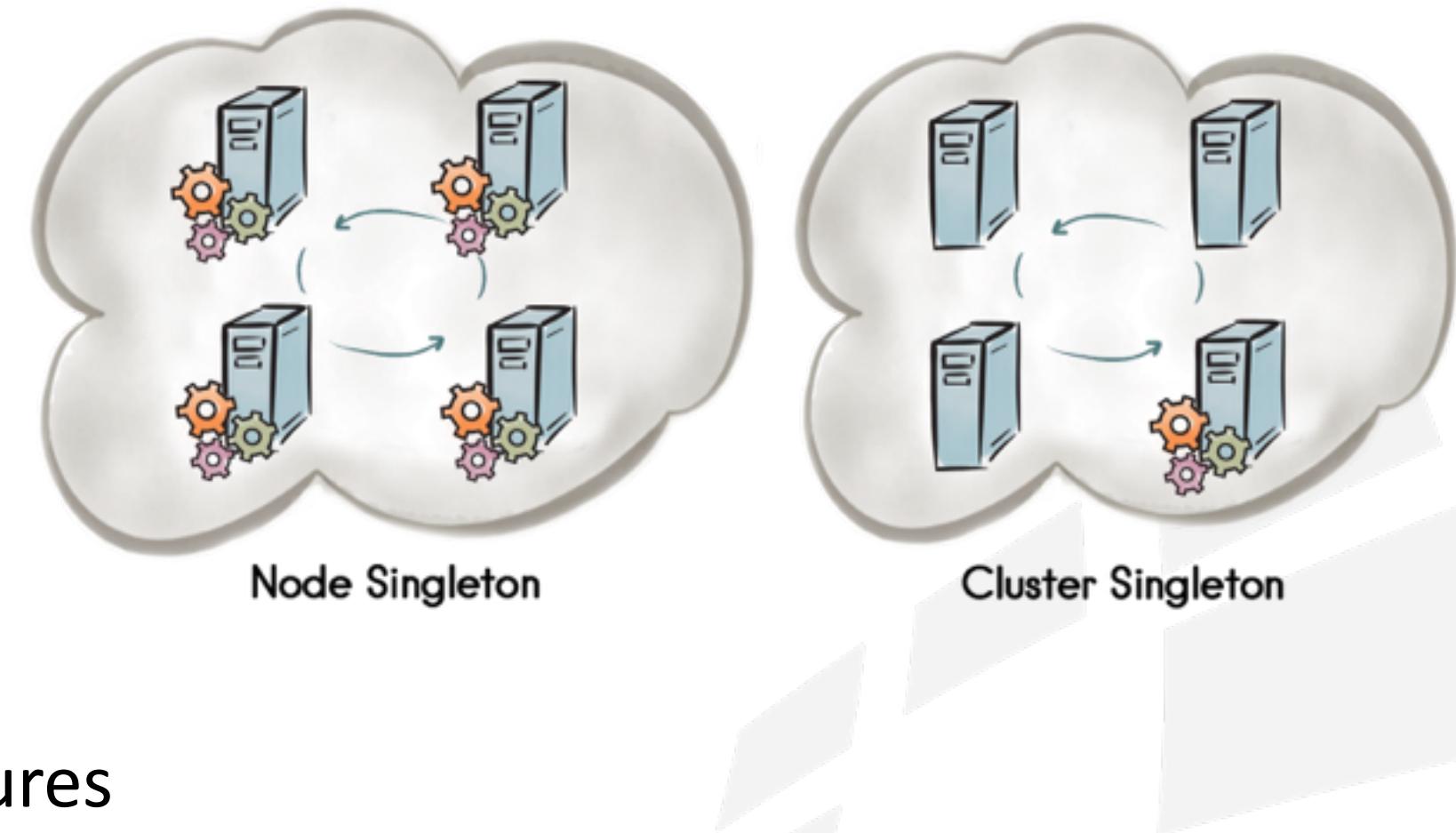
# In-Memory Streaming and CEP

- Streaming Data Never Ends
- Branching Pipelines
- Pluggable Routing
- Sliding Windows for CEP/Continuous Query
- SQL Queries (ANSI 99)
- Query Across Sliding Windows
- Real Time Analysis



# In-Memory Service Grid

- **Singletons on the Cluster**
  - Cluster Singleton
  - Node Singleton
  - Key Singleton
- Distribute any Data Structure
  - Available Anywhere on the Grid
  - Access Anywhere via Proxies
- Guaranteed Availability
  - Auto Redeployment in Case of Failures





# Apache Ignite for BI and Analytics



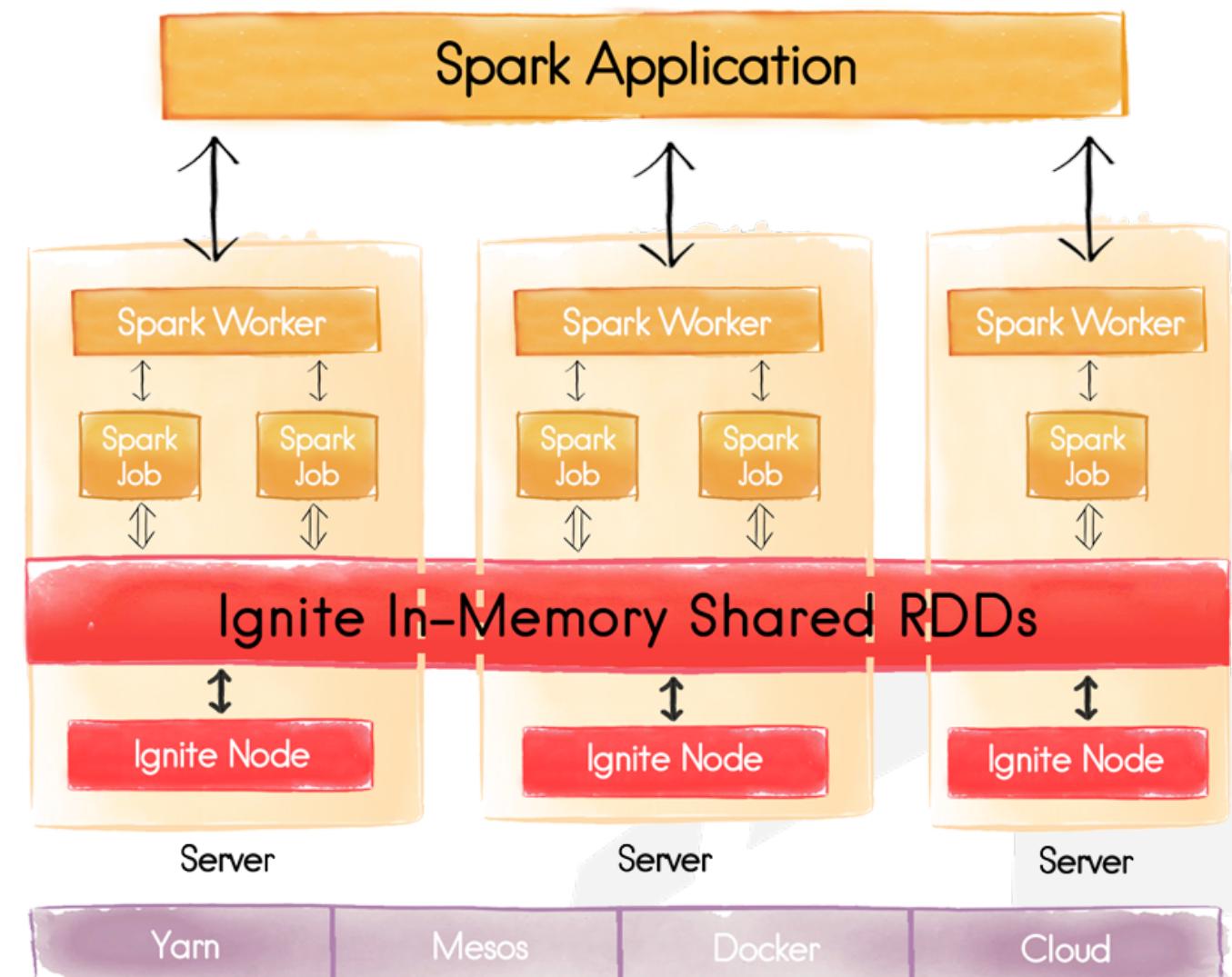
# DevOps: Integration with Yarn and Mesos

- Automatic Resource Management
- Easy Data Center Installation
- Easy Data Center Configuration
- On-Demand Elasticity



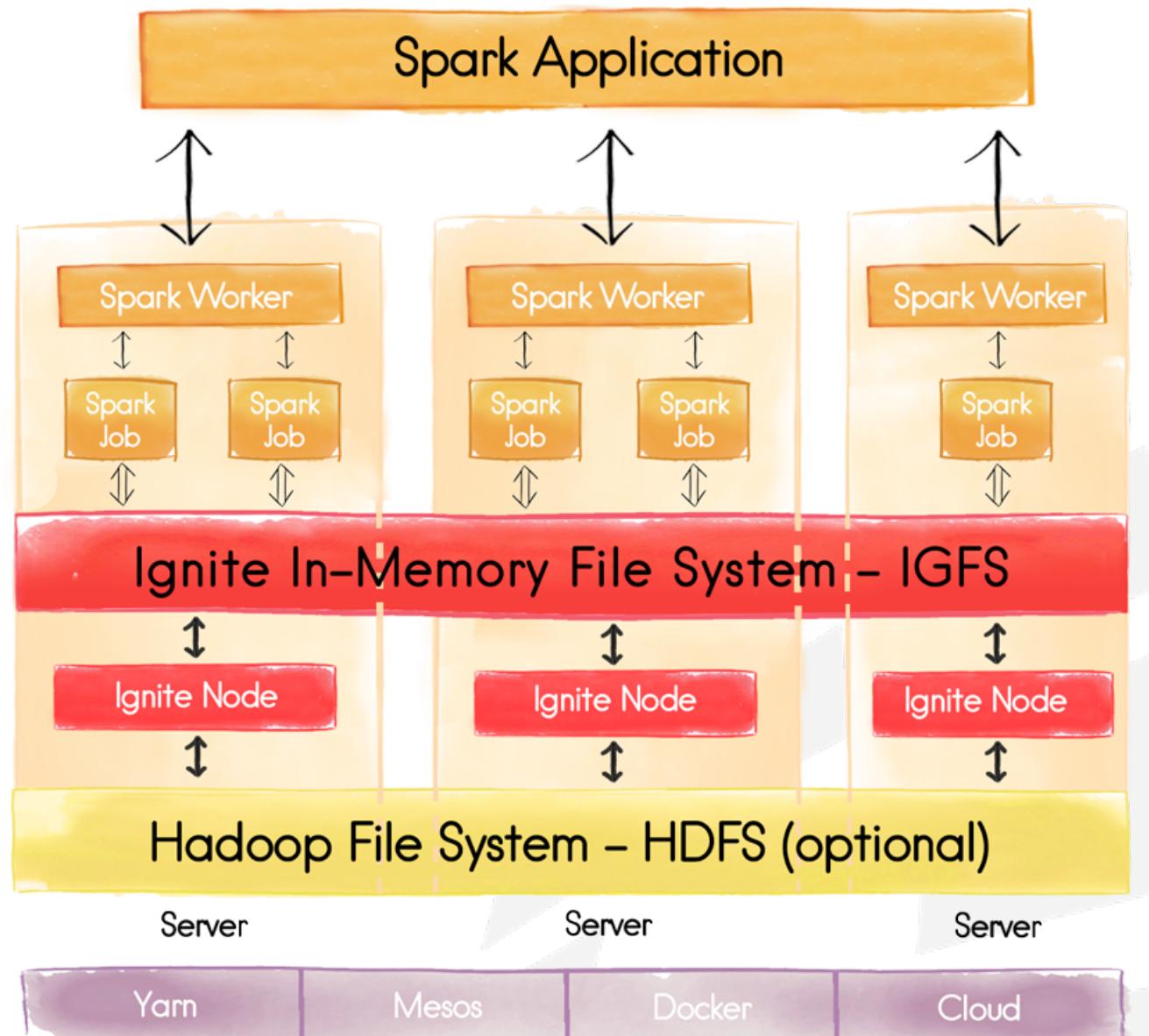
# Share RDDs Across Spark Jobs

- IgniteRDD
  - Share RDD across jobs on the host
  - Share RDD across jobs in the application
  - Share RDD globally
- Faster SQL
  - In-Memory Indexes
  - SQL on top of Shared RDD



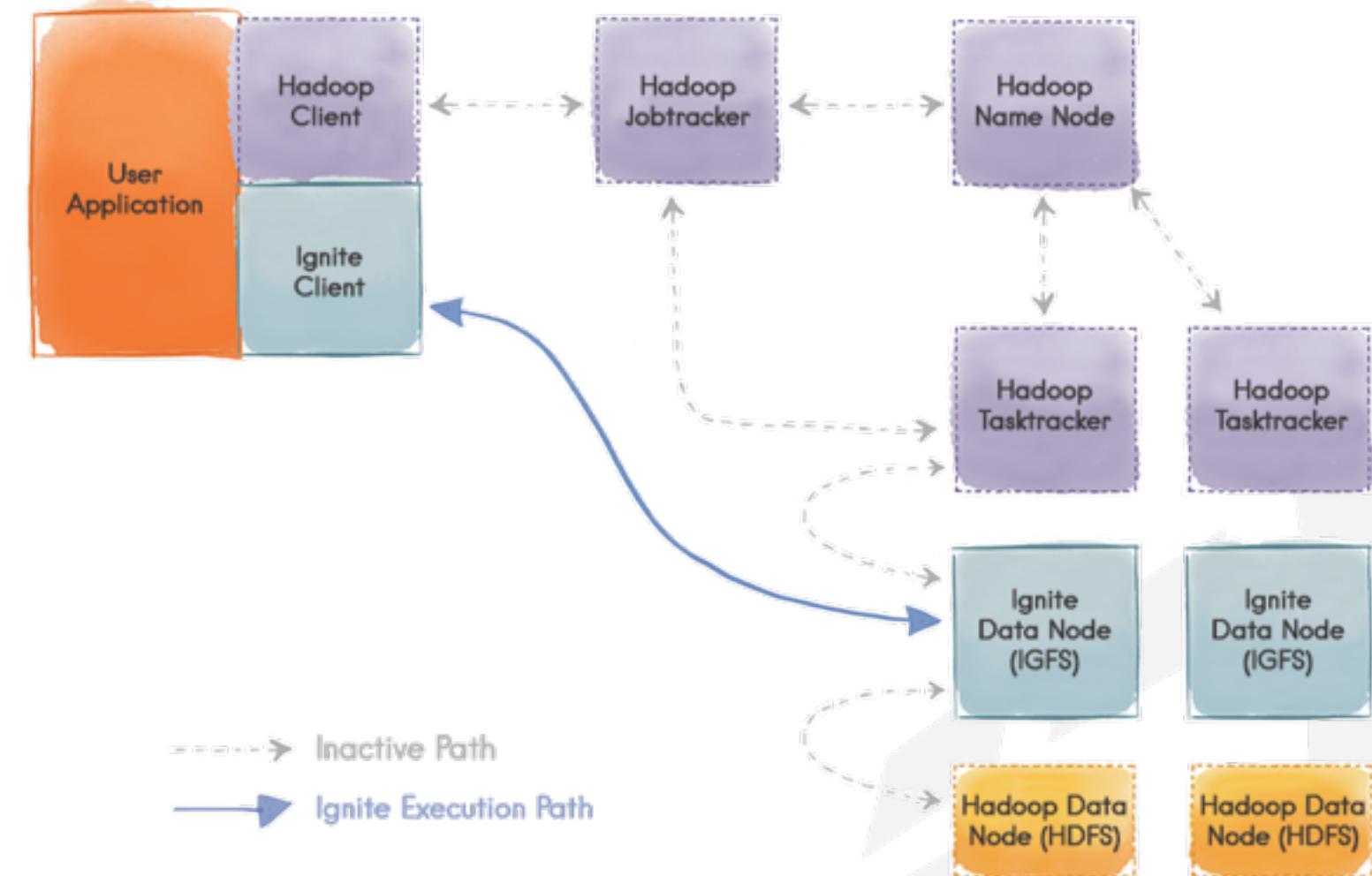
# Ignite In-Memory File System

- Ignite In-Memory File System (IGFS)
  - Hadoop-compliant
  - Easy to Install
  - On-Heap and Off-Heap
  - Caching Layer for HDFS
  - Write-through and Read-through HDFS
  - Performance Boost

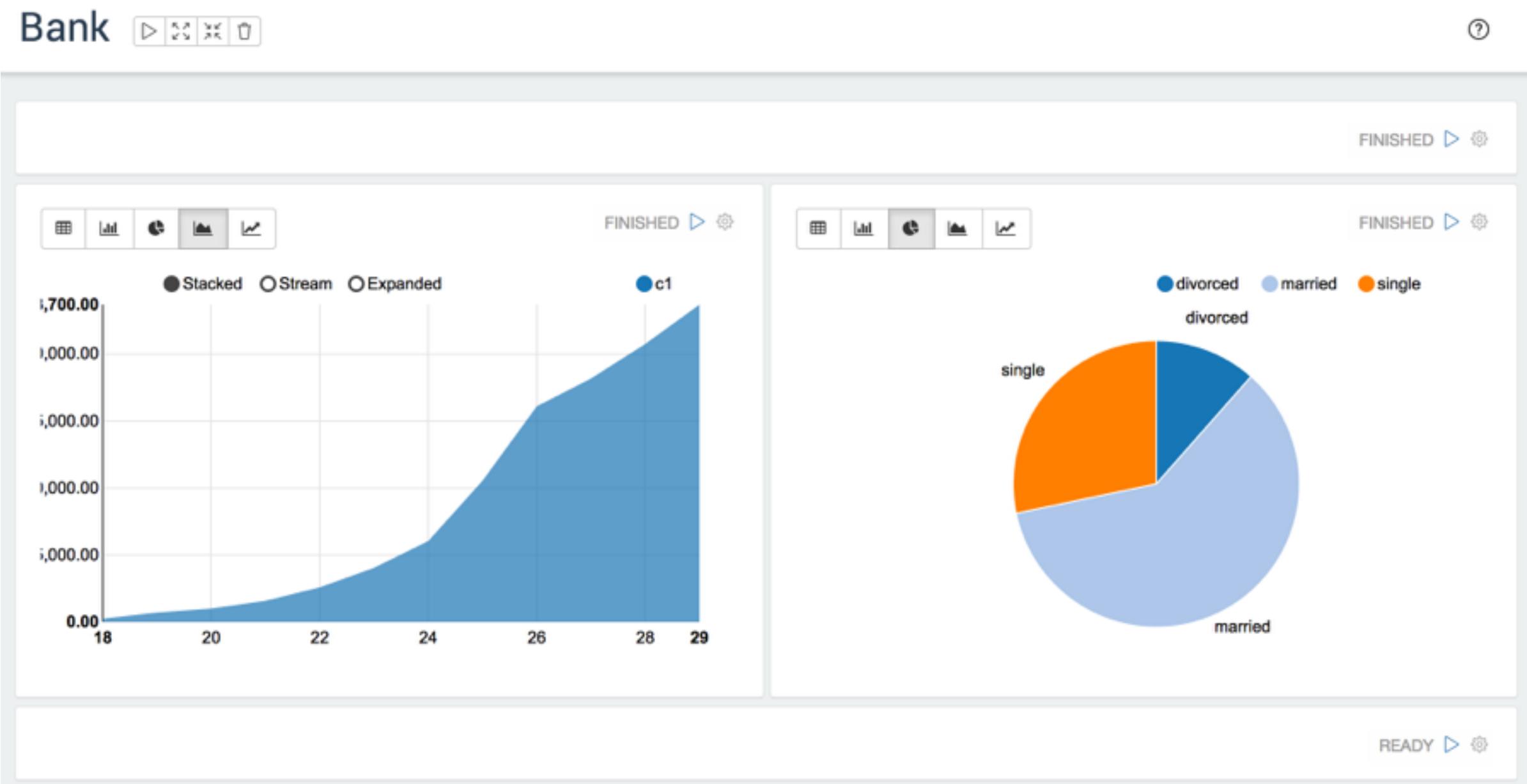


# Ignite In-Memory Map Reduce

- In-Memory Native Performance
- Zero Code Change
- Use existing MR code
- Use existing Hive queries
- No Name Node
- No Network Noise
- In-Process Data Colocation
- Eager Push Scheduling



# Interactive SQL with Apache Zeppelin



# GridGain Enterprise & Apache Ignite Comparison Chart

Features	Apache Ignite	Enterprise Edition
In-Memory Data Grid	✓	✓
In-Memory Compute Grid	✓	✓
Real-Time Streaming & CEP	✓	✓
Hadoop Acceleration	✓	✓
Management & Monitoring GUI		✓
Portable Objects		✓
.Net and C++ APIs		✓
Enterprise-grade Security		✓
Network Segmentation Protection		✓
Local Restartable Store		✓
Rolling Production Updates		✓
Datacenter Replication		✓
9x5 and 24x7 Support		✓
Long Term Support & Patches		✓

**GridGain Enterprise Subscriptions include the following during the term of the subscription:**

- > Right to use GridGain Enterprise Edition
- > Bug fixes, patches, updates and upgrades
- > 9x5 or 24x7 Support
- > Ability to procure Training and Consulting Services from GridGain
- > Confidence and protection, not provided under Open Source licensing, that only a commercial vendor can provide, such as indemnification



# ANY QUESTIONS?

Thank you for joining us. Follow the conversation.

<http://www.ignite.incubator.apache.org>



@apacheignite



@dsetrakyan