



# Apache Ignite - In-Memory Data Fabric

Beyond the Data Grid

**NIKITA IVANOV**  
Founder, Apache PMC



[www.ignite.apache.org](http://www.ignite.apache.org)



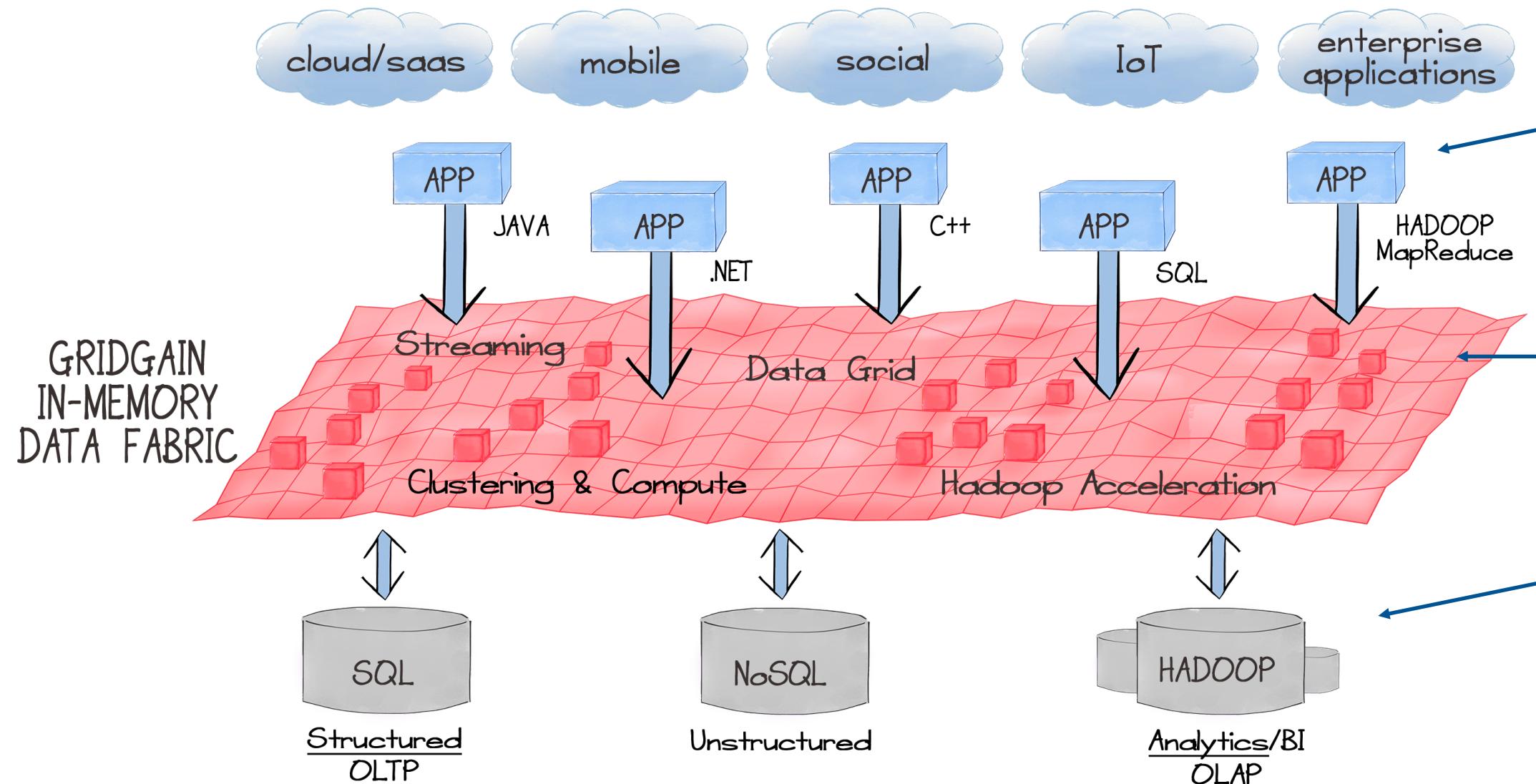
#apacheignite

# Agenda

- Project history
- **In-Memory Data Fabric**
  - Advanced Clustering
  - In-Memory Compute Grid
  - In-Memory Data Grid
  - In-Memory Service Grid
  - In-Memory Streaming & CEP
  - Plug-n-Play Hadoop Accelerator

# In-Memory Data Fabric

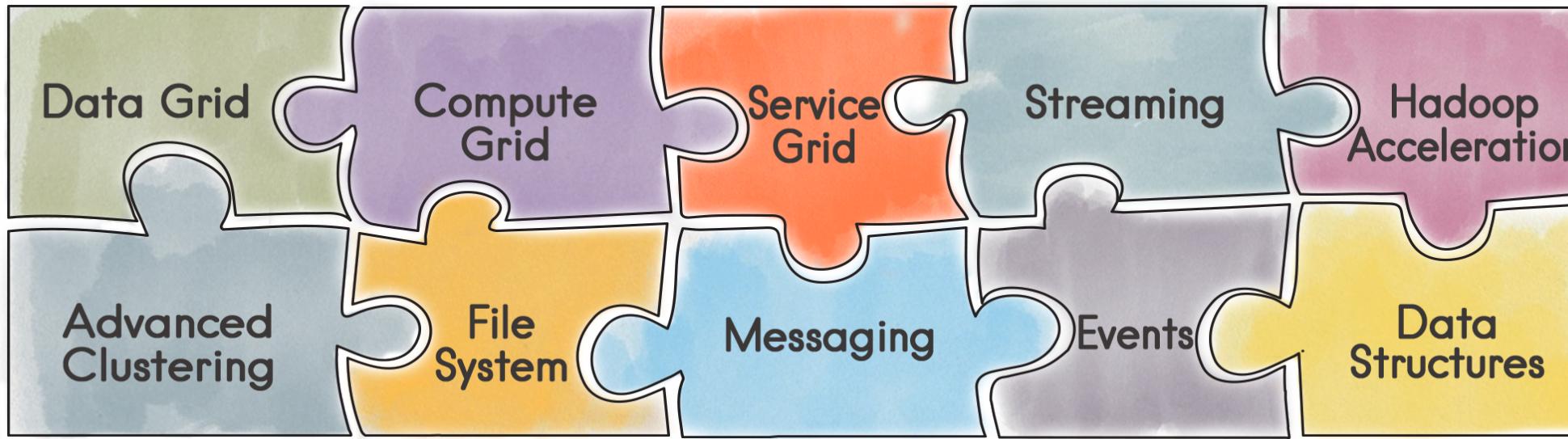
## Strategic Approach to IMC



- Supports Applications of various types and languages
  - Open Source – Apache 2.0
  - Simple Java APIs
  - 1 JAR Dependency
  - High Performance & Scale
  - Automatic Fault Tolerance
  - Management/Monitoring
  - Runs on Commodity Hardware
- 
- Supports existing & new data sources
  - No need to rip & replace

# In-Memory Data Fabric

## Main Characteristics



### Performance

- High Throughput
- Low Latencies

### Scalability

- Add Cluster Members (cores)
- Add Memory (RAM)

### High Availability

- Data Backups
- Datacenter Replication

### Transactions

- Fully ACID Compliant
- Optimistic & Pessimistic

### Persistence

- SQL, NoSQL, Hadoop

### Security

- Authentication
- Authorization
- Tracing & Auditing

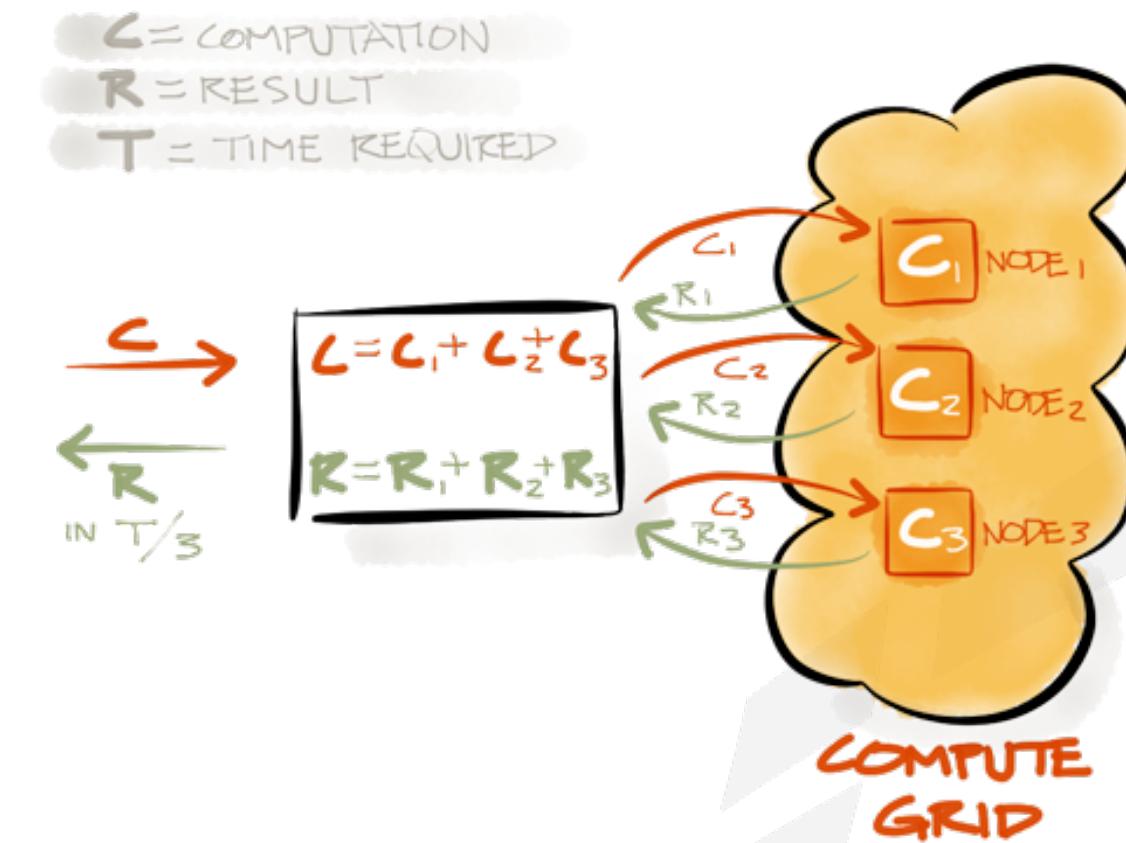
# In-Memory Clustering & Deployment

- Ease of Getting Started
  - Automatic Discovery
- Any Environment
  - Public Cloud
  - Private Cloud
  - Hybrid Cloud
  - Local Laptop
- Zero-Deployment
  - Auto-Deploy Code
- Full Cluster Management
- Pluggable Design



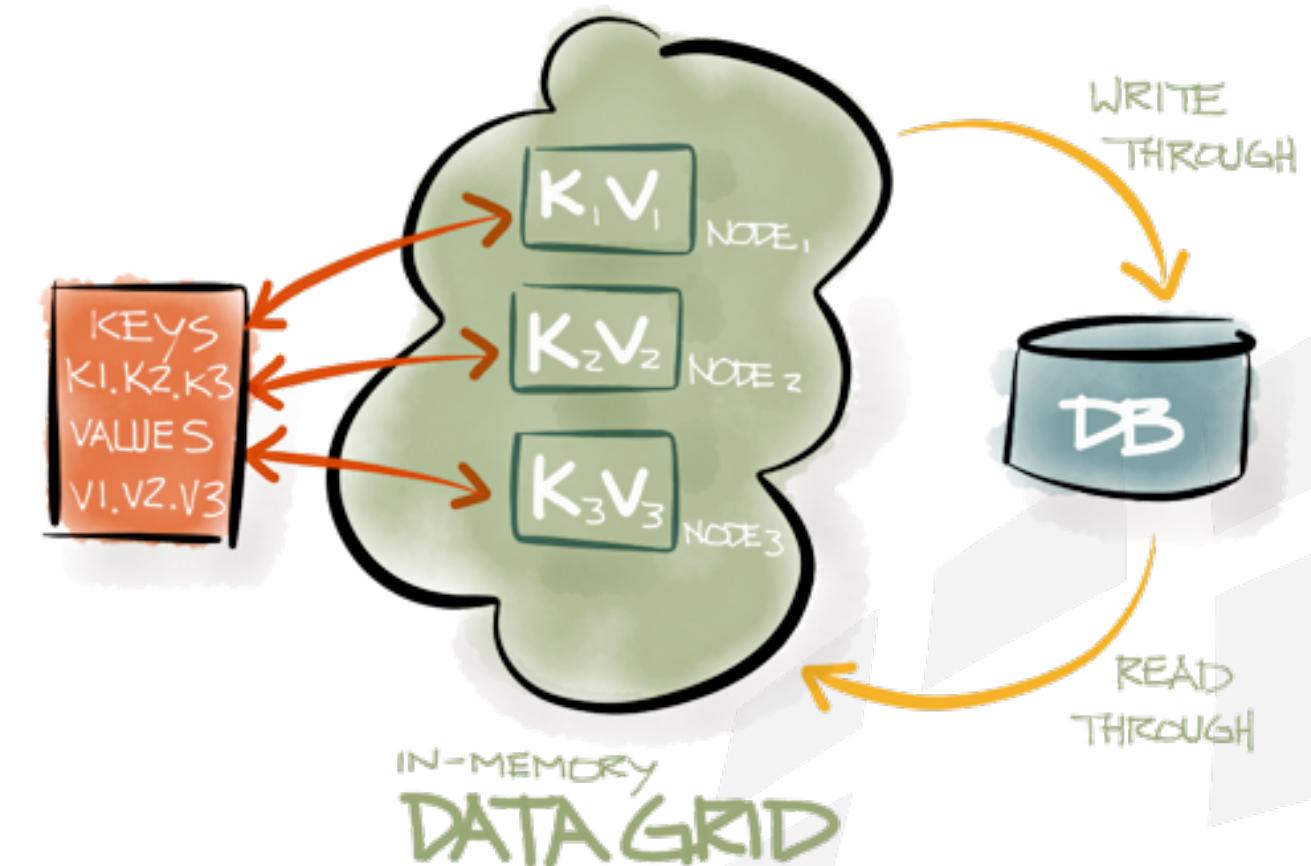
# In-Memory Compute Grid

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



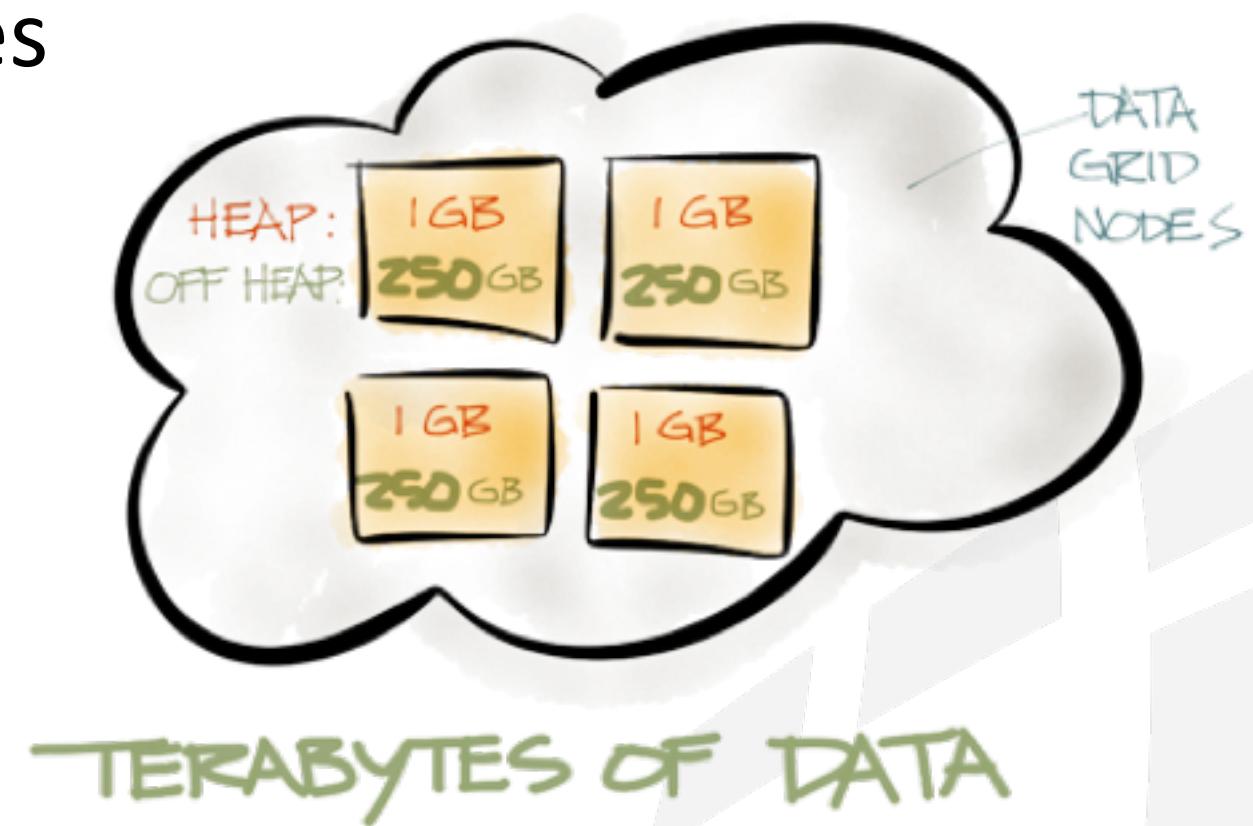
# In-Memory Data Grid

- Distributed In-Memory Key-Value Store
- Replicated and Partitioned data
- TBs of data, of any type
- On-Heap and Off-Heap Storage
- Highly Available In-Memory Replicas
- Automatic Failover
- Distributed ACID Transactions
- SQL queries and JDBC driver
- Collocation of Compute and Data



# In-Memory Data Fabric: 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



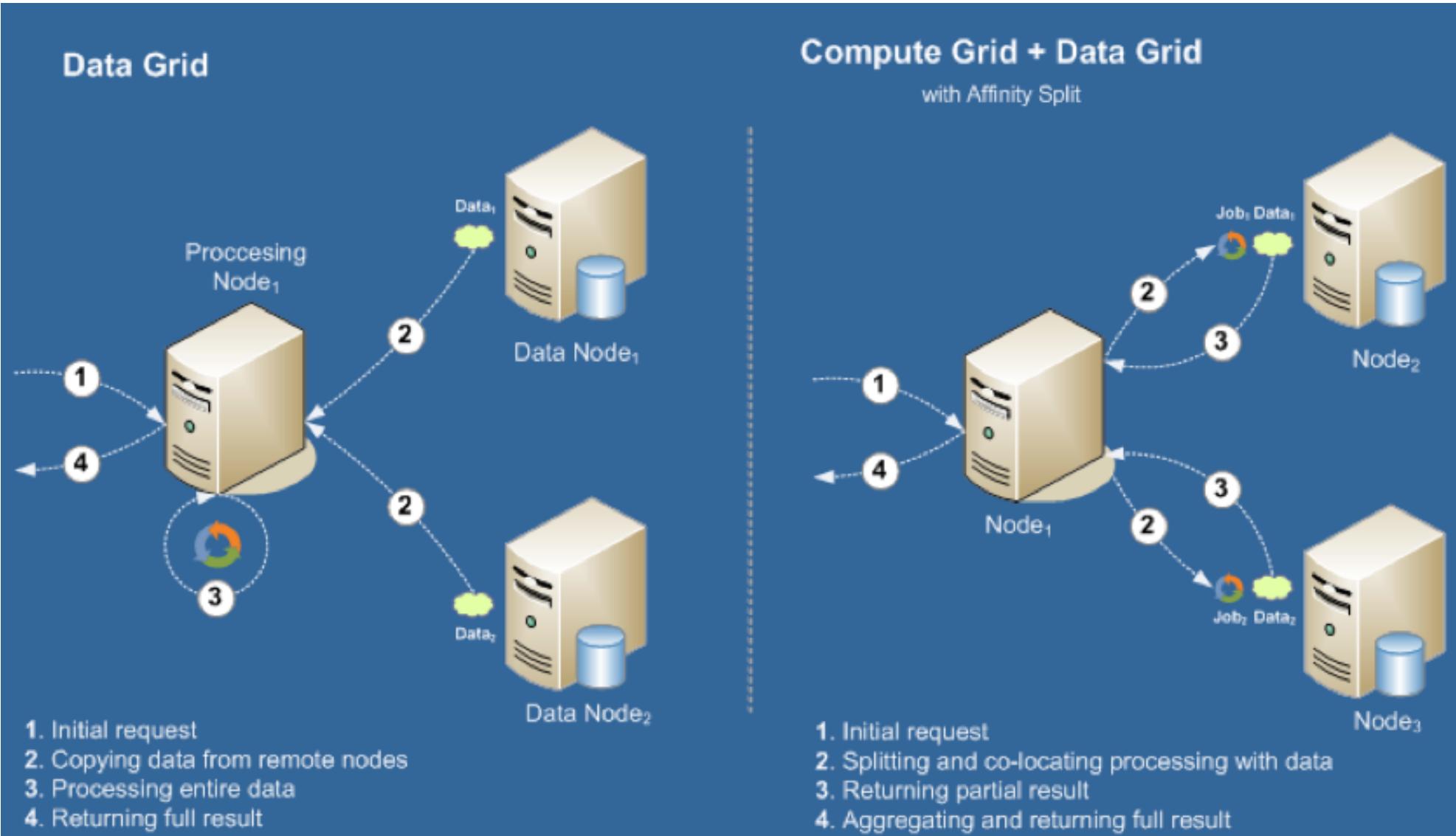
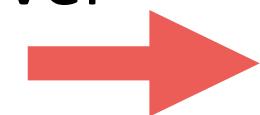
# Distributed Java Structures

- Distributed Map (cache)
- Distributed Set
- Distributed Queue
- CountDownLatch
- AtomicLong
- AtomicSequence
- AtomicReference
- Distributed ExecutorService

```
GridCacheQueue<Integer> queue =  
    ... dataStructures.queue("myQ",  
  
    // Distribute queue elements  
    // across grid.  
    for (int i = 0; i < 20; i++)  
        queue.add(i);  
  
    // Poll queue elements.  
    for (int i = 0; i < 20; i++)  
        queue.poll();
```

# Client-Server vs Affinity Colocation

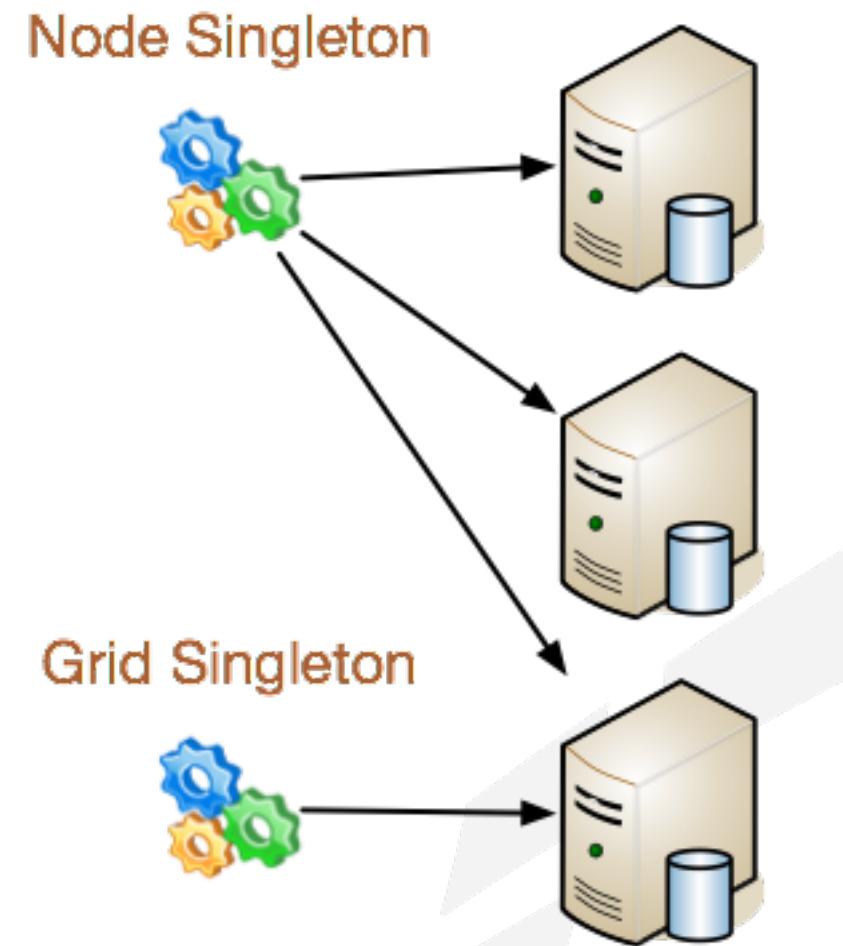
Client-  
Server



Affinity  
Colocation

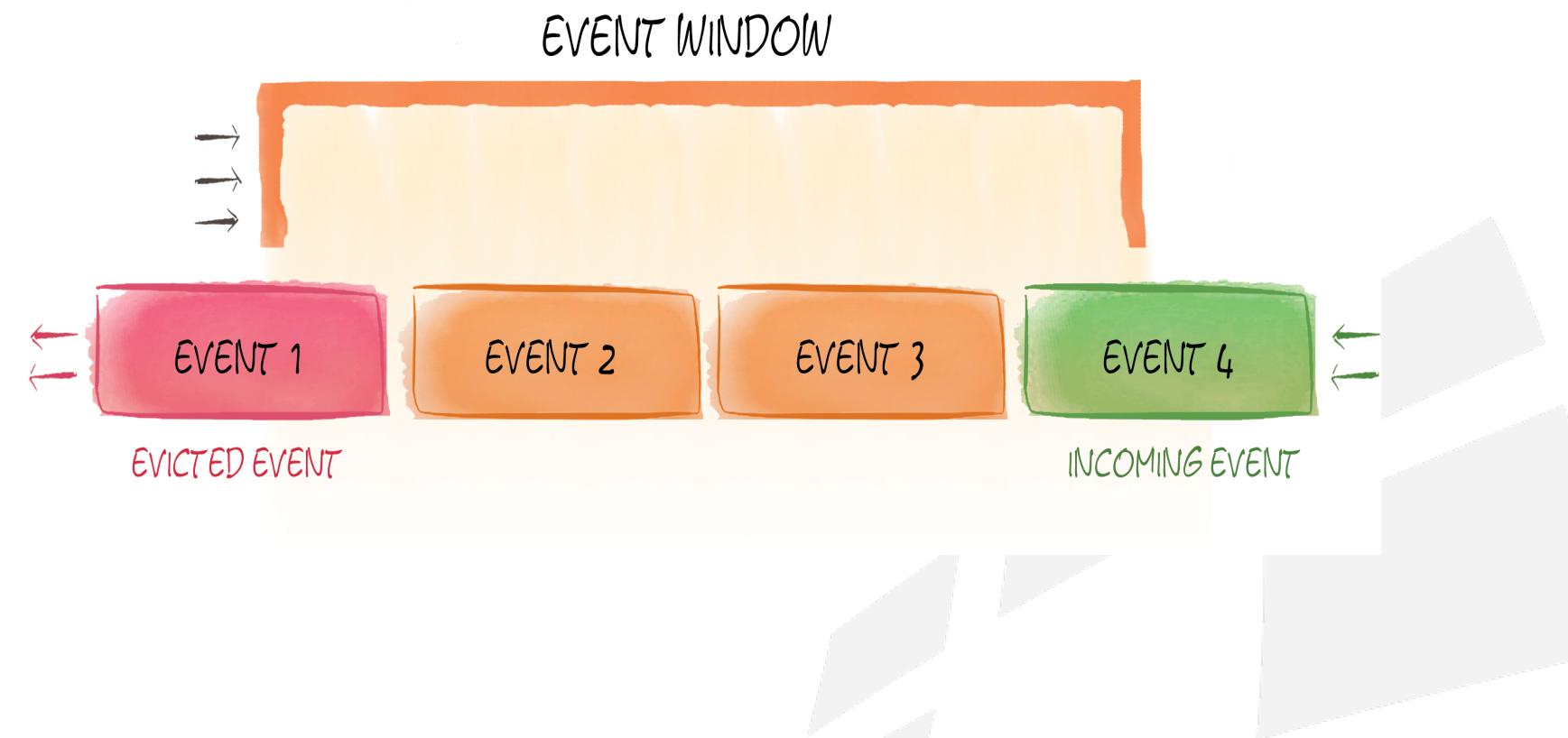
# In-Memory Service Grid

- Distribute Any Data Structure
  - Available Anywhere on the Grid
  - Automatic Remote Access via Proxies
- Controlled Deployment
  - Support for Cluster Singleton
  - Support for Node Singleton
  - Support for Custom Topology
  - Load Balanced
- Guaranteed Availability
  - Auto Redeployment in Case of Failures



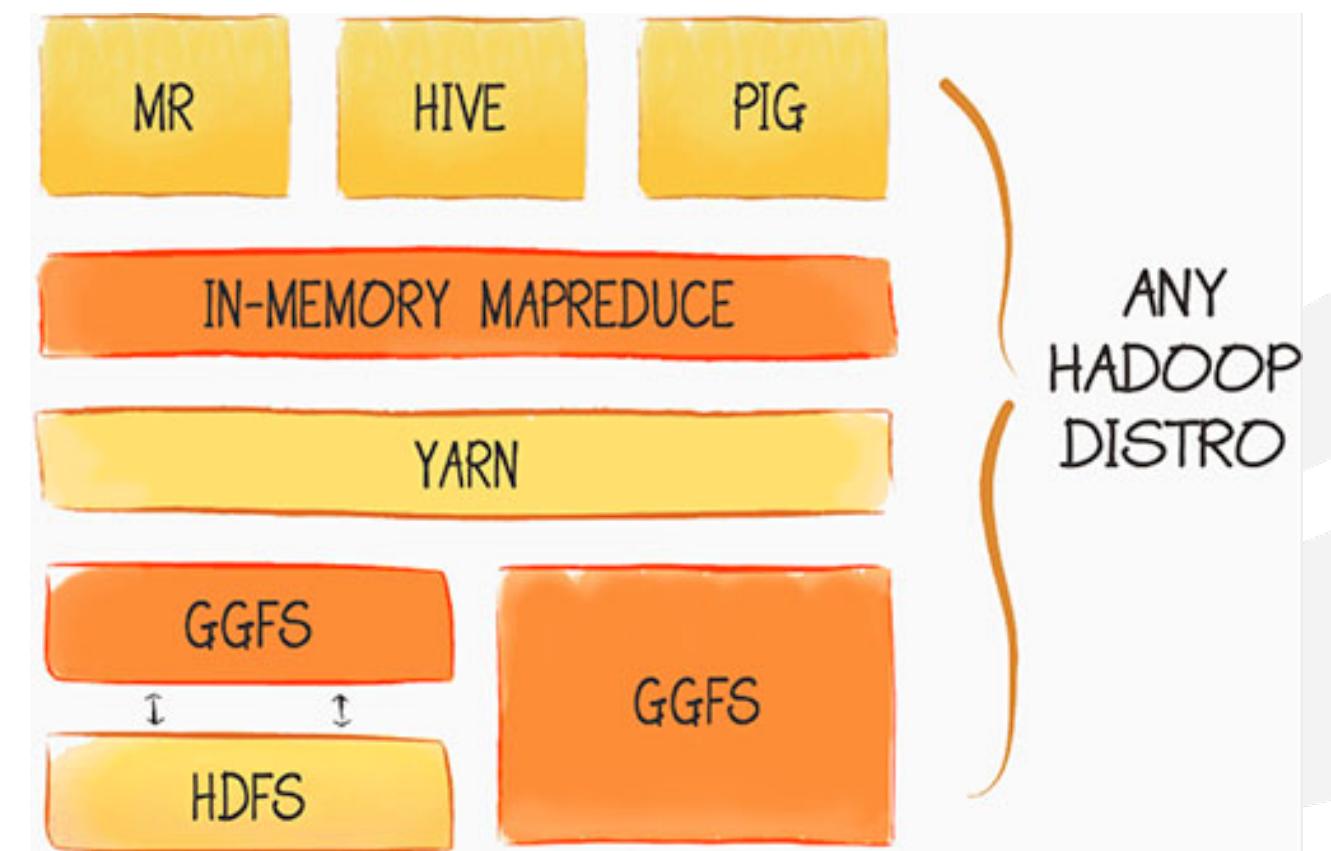
# In-Memory Streaming and CEP

- Streaming Data Never Ends
- Branching Pipelines
- Pluggable Routing
- Sliding Windows for CEP/Continuous Query
- Real Time Analysis



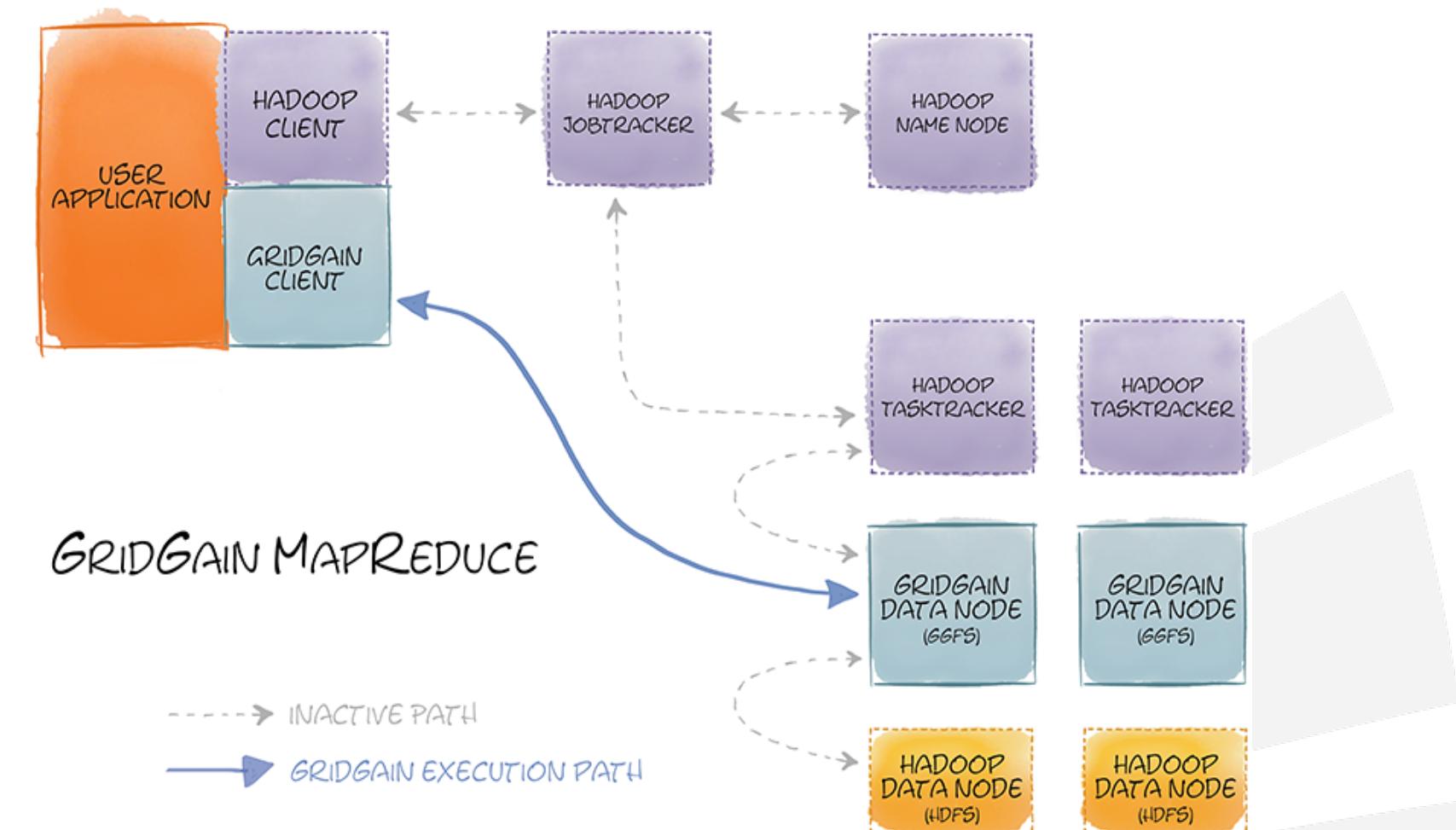
# In-Memory Hadoop Accelerator

- **Plug and Play installation**
- 10x to 100x Acceleration
- In-Memory Native MapReduce
- In-Process Data Colocation
- IgniteFS In-Memory File System
- Read-Through from HDFS
- Write-Through to HDFS
- Sync and Async Persistence

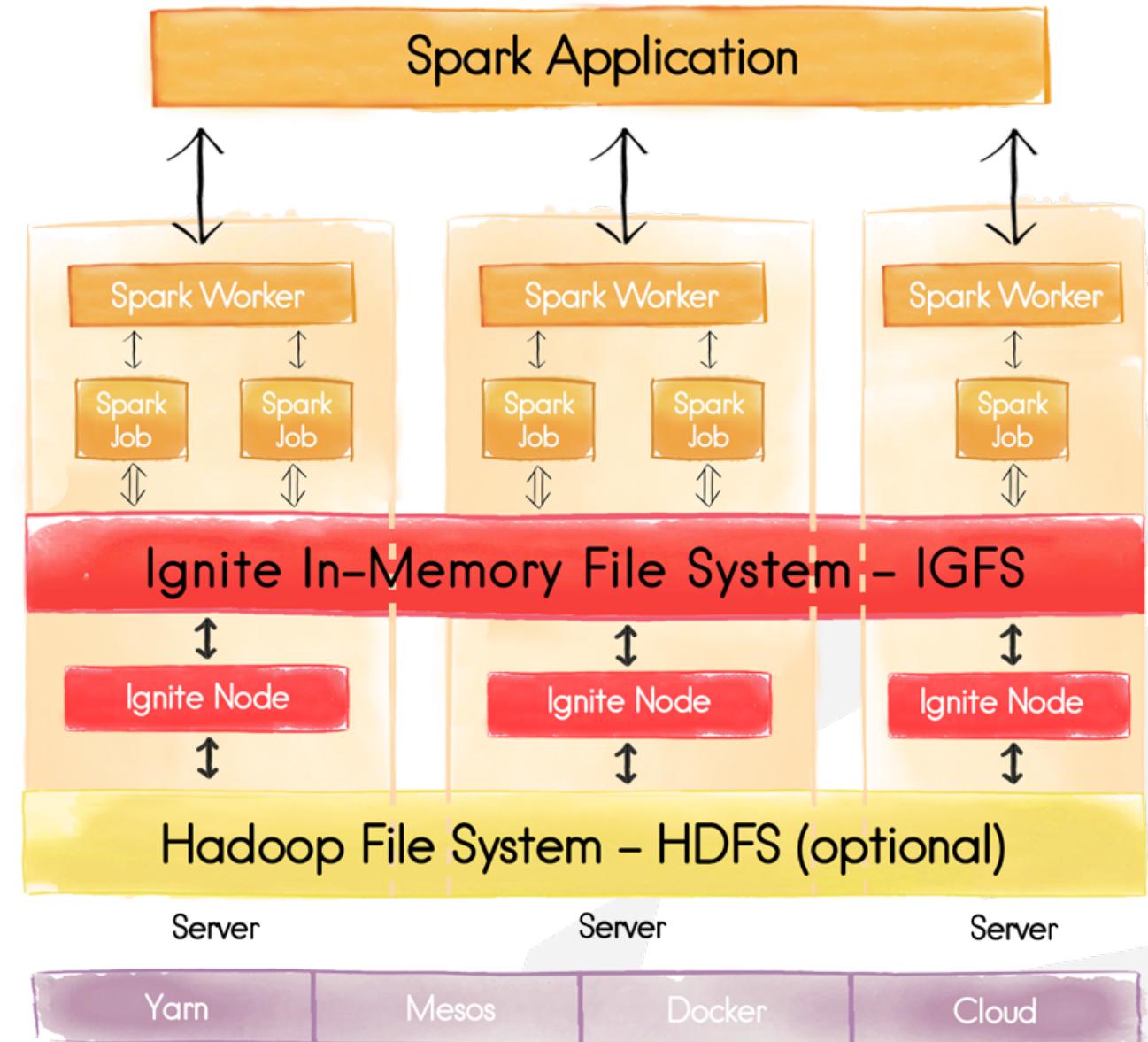
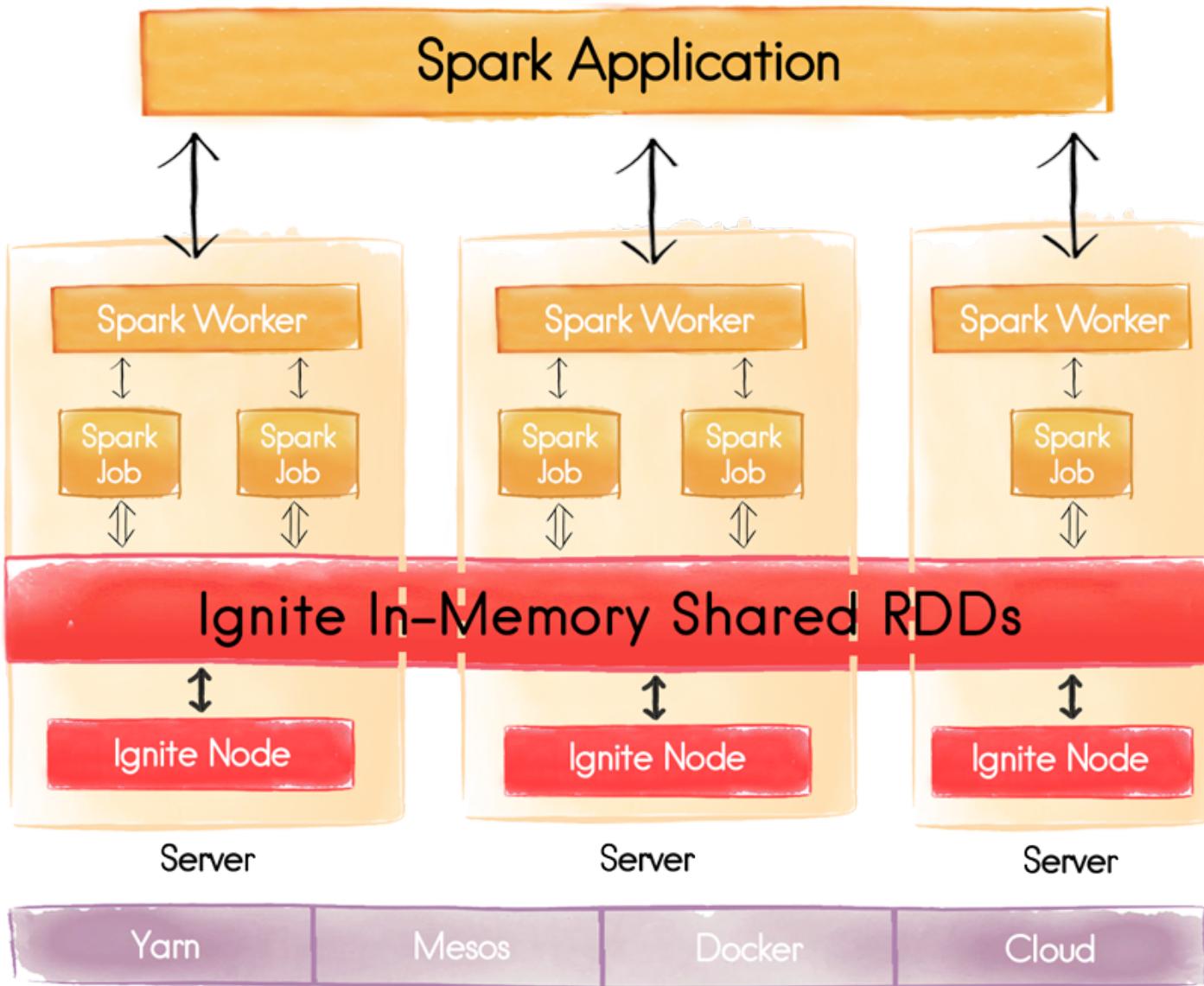


# In-Memory Hadoop Accelerator

- Zero Code Change
- In-Memory Native Performance
- Use existing MR code
- Use existing Pig/Hive queries
- No Name Node
- Eager Push Scheduling

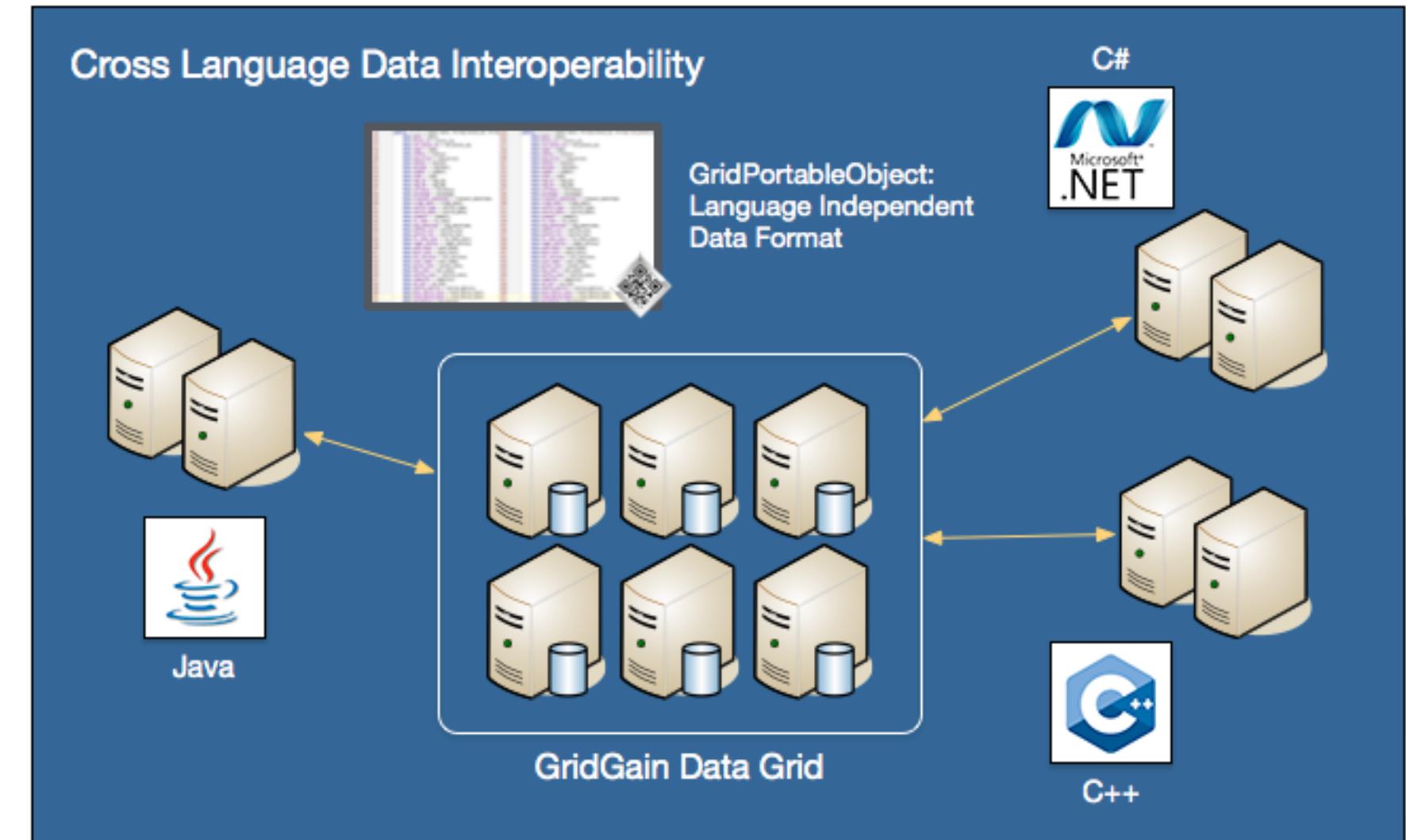


# Spark Integration – IGFS & Shared RDD



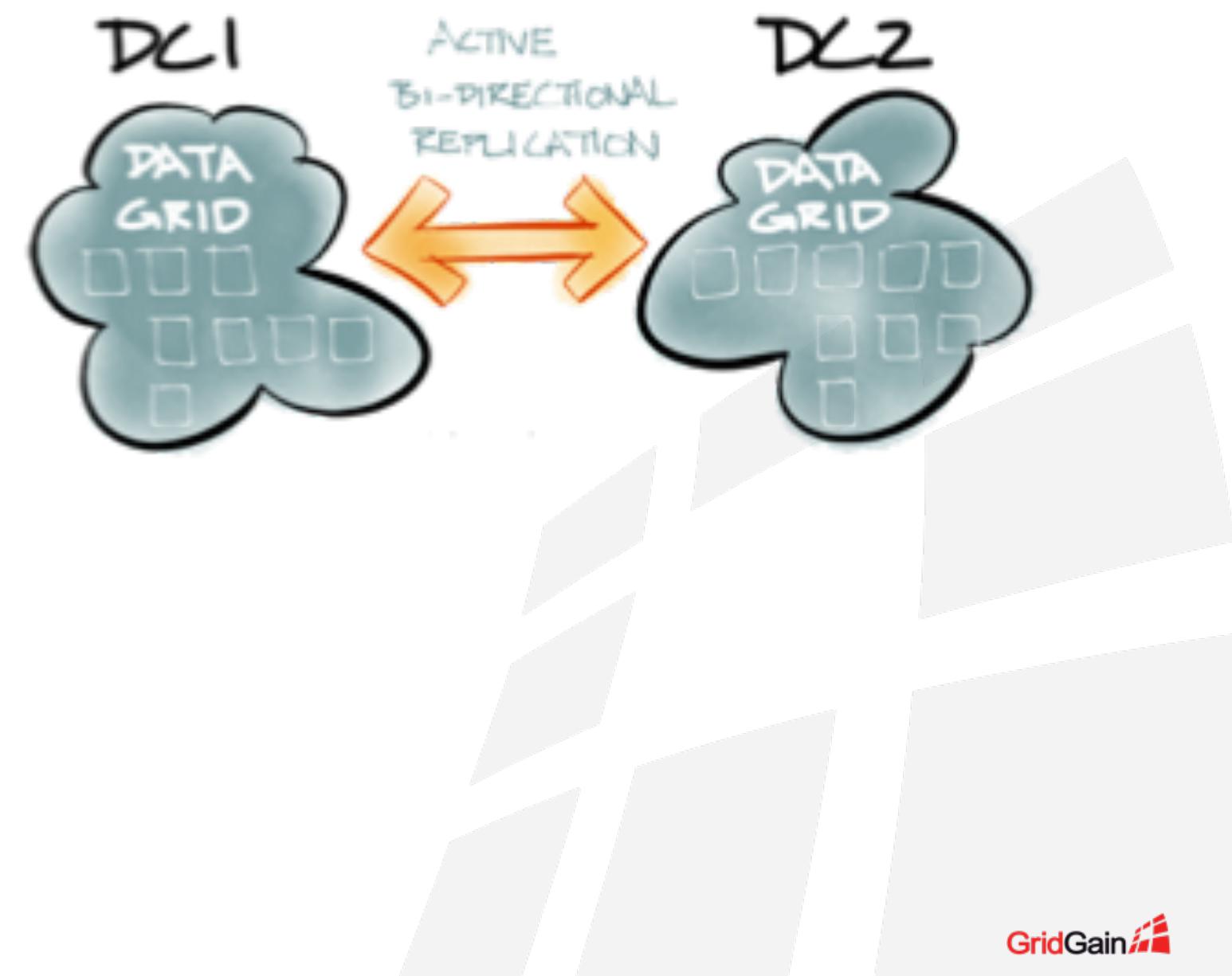
# Cross-Language Interoperability

- C++/.NET/PHP/Java/Node.js
- Portable Objects
- Performance Across Languages
- Client Feature Parity
- Dynamic Schema Changes
- Searchable/Indexable
- Version Independent

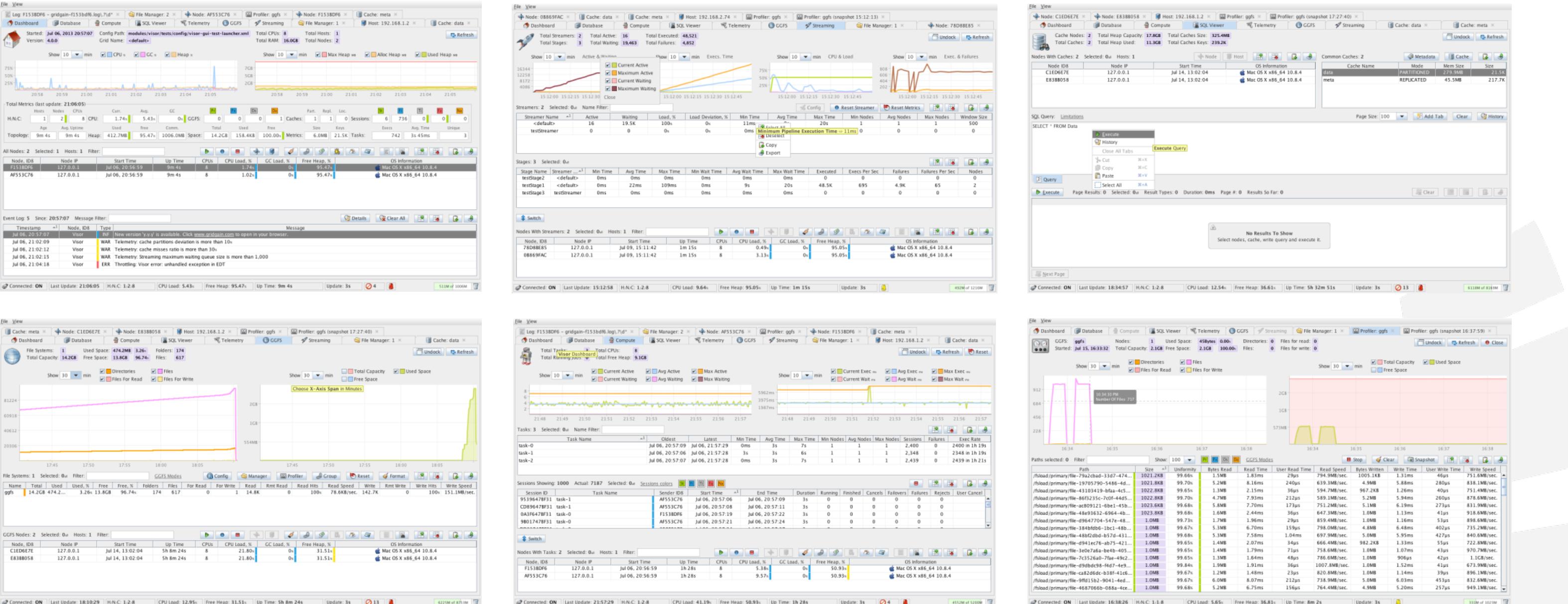


# In-Memory Data Fabric: Data Center Replication

- Up to 32 Data Centers
- Active-Active & Active-Passive
- Smart Conflict Resolution
- Durable Persistent Queues
- Automatic Throttling
- \* *Enterprise Edition Only*



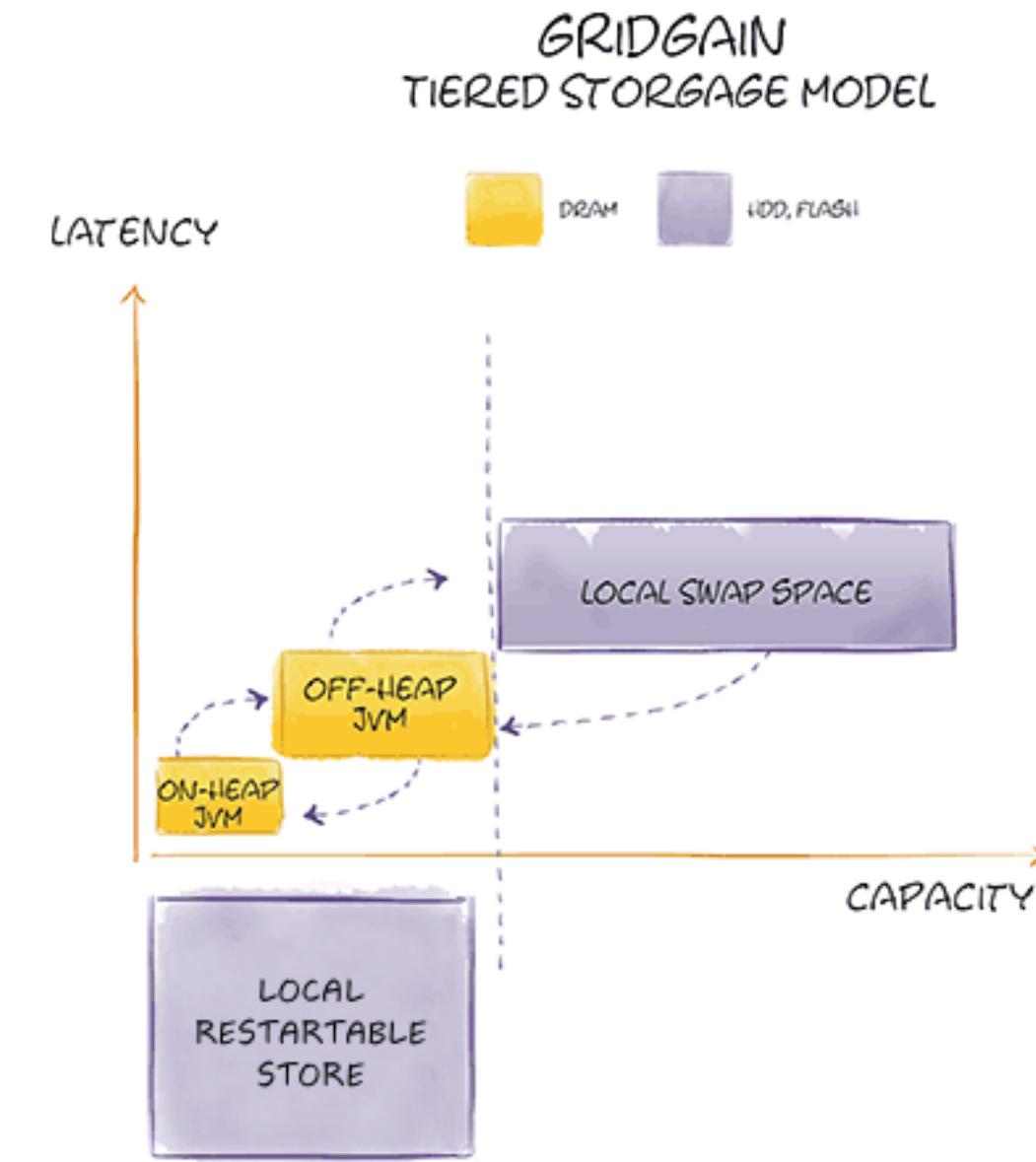
# Management & Monitoring



★ Enterprise Edition Only

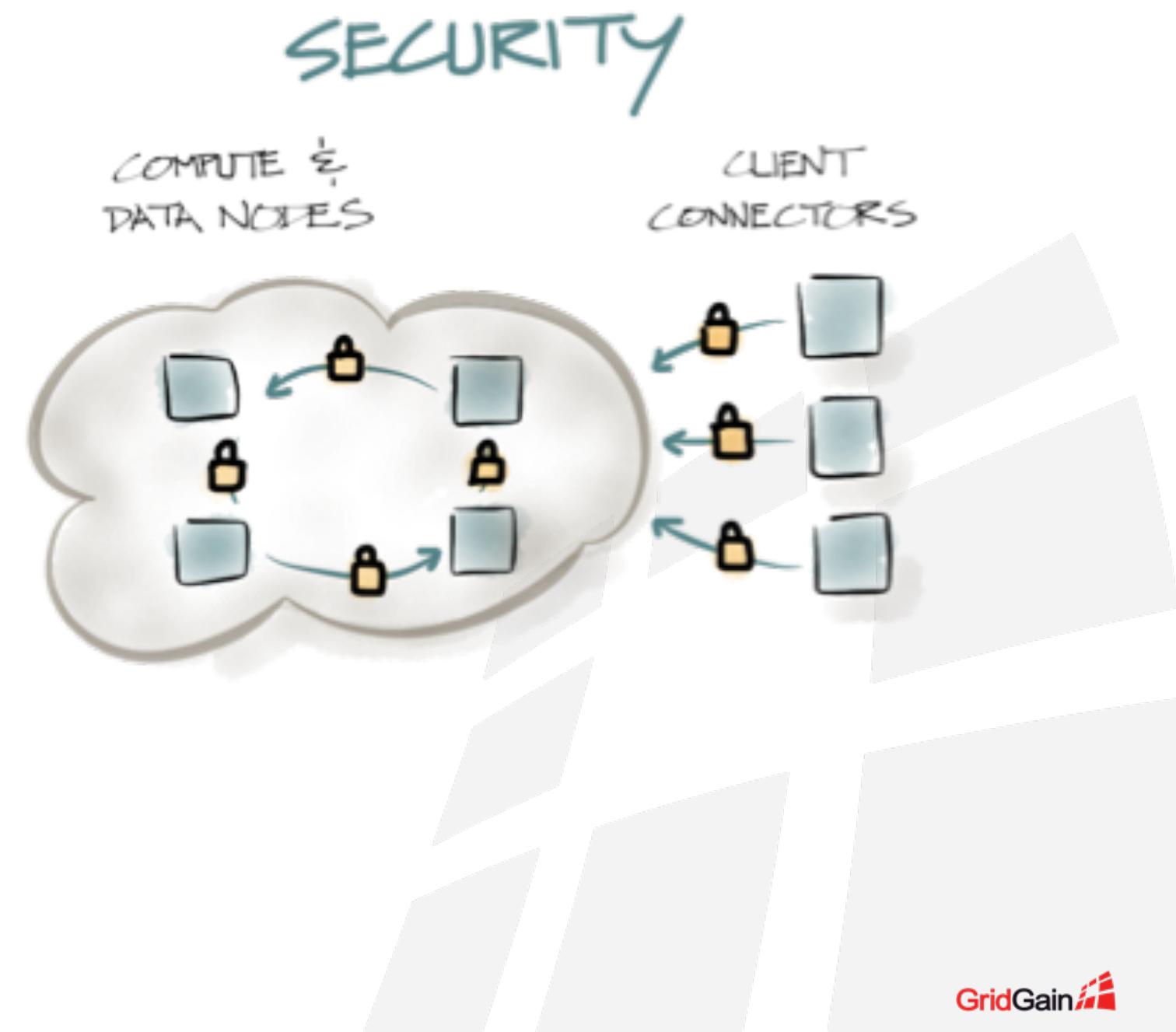
# Local Restartable Store

- Persistent On-Disk Store
  - Fast Recovery
  - Local Data Reload
    - Eliminate Network and Db impacts when reloading in-memory store
- \* *Enterprise Edition Only*



# In-Memory Data Fabric: Security

- Pluggable Auth & Auth
    - JAAS, LDAP, JNDI, Kerberos
  - In-Cluster Node Authentication
  - Client Authentication
  - Secure Client Sessions
  - Fine-Grained Authorization
  - Comprehensive Auditing
    - Who? What? When?
- \* *Enterprise Edition Only*





# THANK YOU!



[www.ignite.apache.org](http://www.ignite.apache.org)



#apacheignite