Hazelcast Overview

Chris Wilson – VP of Sales
chris.wilson@hazelcast.com
Rahul Gupta – Senior Solution Architect
rahul@hazelcast.com

Hazelcast is an operational, in-memory, distributed computing platform that manages data using in-memory storage, and performs parallel execution for breakthrough application speed and scale.

Distributed Computing. Simplified.

Training in Hong Kong – October 27, 2016

Free Instructor Led Training in Hong Kong

October 27, 2016 @ 10am-3pm — Paperclip HK, Nam Wo Hong Building, 148 Wing Lok Street, Sheung Wan, Hong Kong

Details Ready to take the first steps in understanding in-memory data grids? This free session of Hazelcast Essentials walks developers through how to construct Hazelcast clusters and deliver basic caching/compute services. Students will be introduced to the fundamental features of Hazelcast and how they may be applied to solve various use cases. This free course [...]

The Shift to Web-Scale and Real-time In-memory Computing

Real-Time Latency

Do things up to 1000x faster than a database

Web Scale

Scale up and out to support the largest use cases

Business Moments

Spot transient opportunities to exploit dynamically

Fast Big Data

Big Data with low latencies for batch and streaming

Situational Awareness

Decision makers' instant business-state understanding

Company Snapshot

- Founded in 2008, 75 staff
- Commercial Open Source Business Model
- Gartner “Market Guide for IMDG” 2015,
  Leader in Forrester IMDG Wave Report 2015
- Headquarters in Palo Alto with offices in London, New York, Istanbul

BOARD MEMBERS

Greg Luck
CEO
Terracotta
Ehcache

Fuad Malikov
Founder & VP Technical Ops
IBM

Roland Manger
Earlybird Venture

Salil Deshpande
Bain Capital Ventures

Ali Kutay
CEO Striim
CEO Golden Gate

Rod Johnson
CEO SpringSource

MANAGEMENT TEAM

Greg Luck
CEO
Terracotta
Ehcache

Fuad Malikov
Founder & VP Technical Ops
IBM

Enes Akar
VP Engineering

Chris Wilson
VP Sales
Oracle
Skytree

Morgan Dioli
VP Finance
Twitter
Terracotta

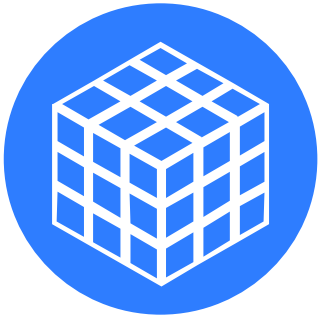
Kevin Cox
VP Marketing
SAP
EXASOL

Hazelcast Use Cases

High-Density Caching

- High-Density Memory Store, client and member
- Full JCache support
- Elastic scalability
- Super fast
- High availability
- Fault tolerance

In-Memory Data Grid



- Simple, modern APIs
- Distributed Data Structures
- Distributed Compute
- Distributed Clustering
- Object-oriented and non-relational
- Elastic and scalable
- Transparent database integration
- Cluster Management

Microservices Infrastructure

- Isolation of Services with many, small clusters
- Service registry
- Network discovery
- Inter-process messaging
- Fully Embeddable
- Spring Cloud, Boot Data Services

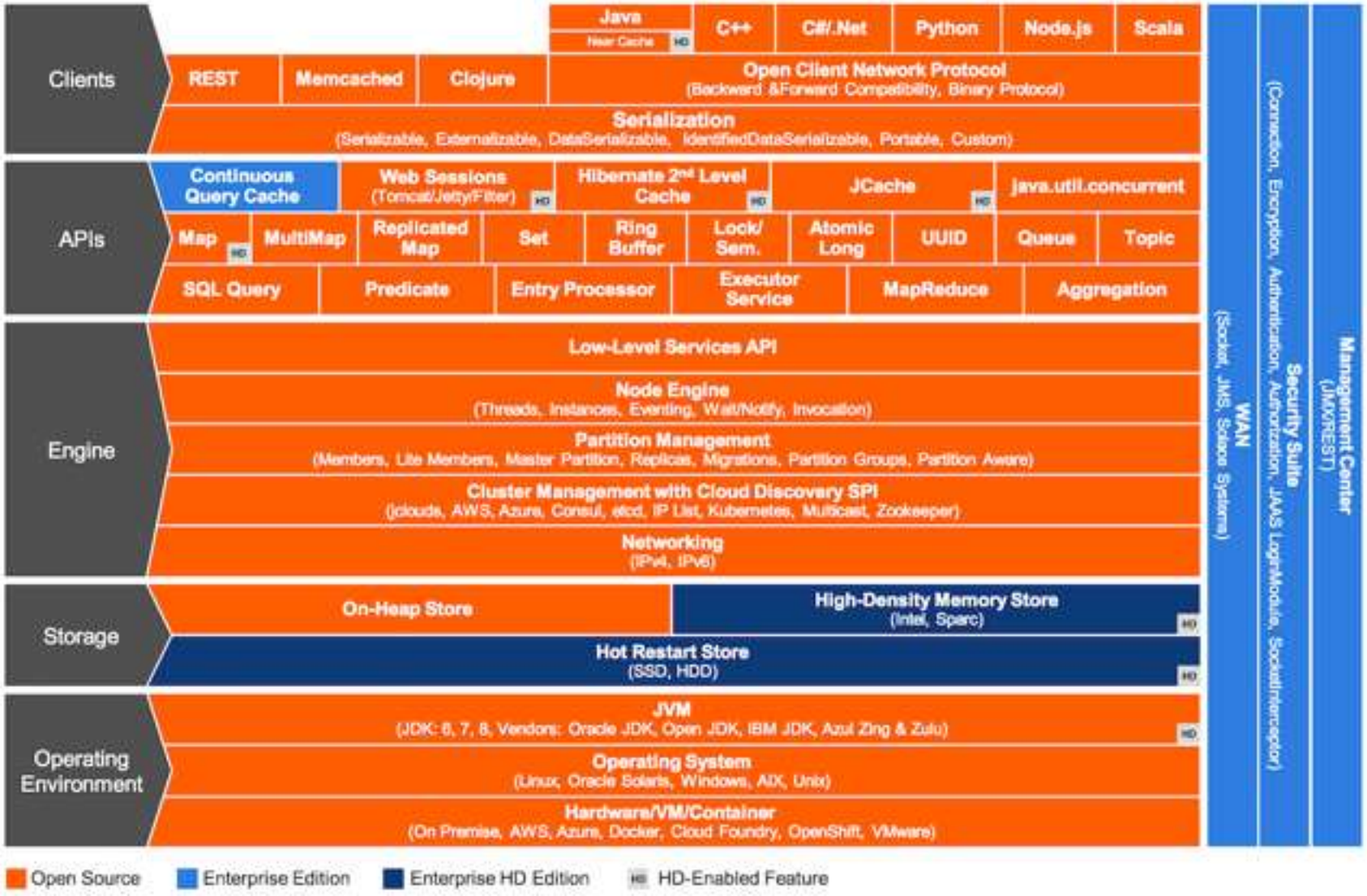
Web Session Clustering

- Seamless failover between user sessions
- High performance
- No application alteration
- Easy scale-out
- Fast session access
- Off load to existing cluster
- Tomcat, Jetty + any Web Container
- Works efficiently with large session objects using delta updates

Analyst Reports

Gartner	Hazelcast reviewed in Gartner “Market Guide for In-Memory Data Grids” [subscription required] https://www.gartner.com/doc/3092924/market-guide-inmemory-data-grids
OVUM	**“On the Radar: An open-source in-memory data grid platform for Java”** [subscription required] https://www.ovumkc.com/Products/IT/Infrastructure-Solutions/On-the-Radar-Hazelcast/Summary
FORRESTER®	**Hazelcast Inc cited as Leader by Independent Research Firm** [subscription required] https://www.forrester.com/The+Forrester+Wave+InMemory+Data+Grids+Q3+2015/quickscan/-/E-RES120420

Hazelcast In-memory Platform



Why Hazelcast?

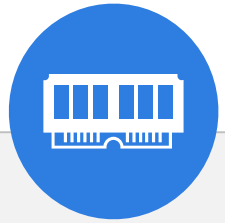
- **Scale-out Computing** enables cluster capacity to be increased or decreased on-demand

- **Resilience** with automatic recovery from member failures without losing data while minimizing performance impact on running applications

- **Programming Model** provides a way for developers to easily program a cluster application as if it is a single process

- **Fast Application Performance** enables very large data sets to be held in main memory for real-time performance

In Memory Data Grid

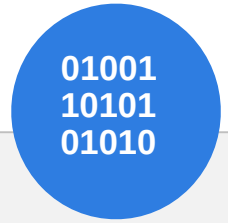


In Memory Data Storage

+

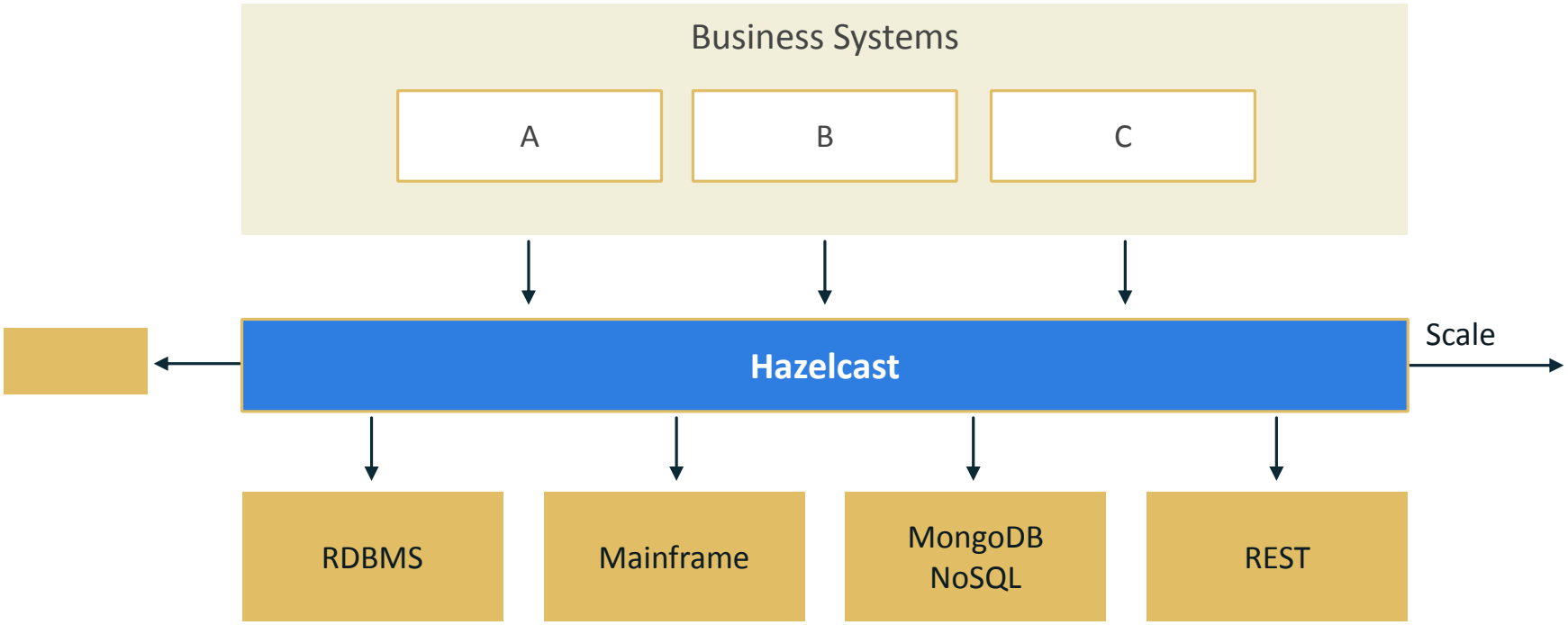
In Memory Data Computing

+

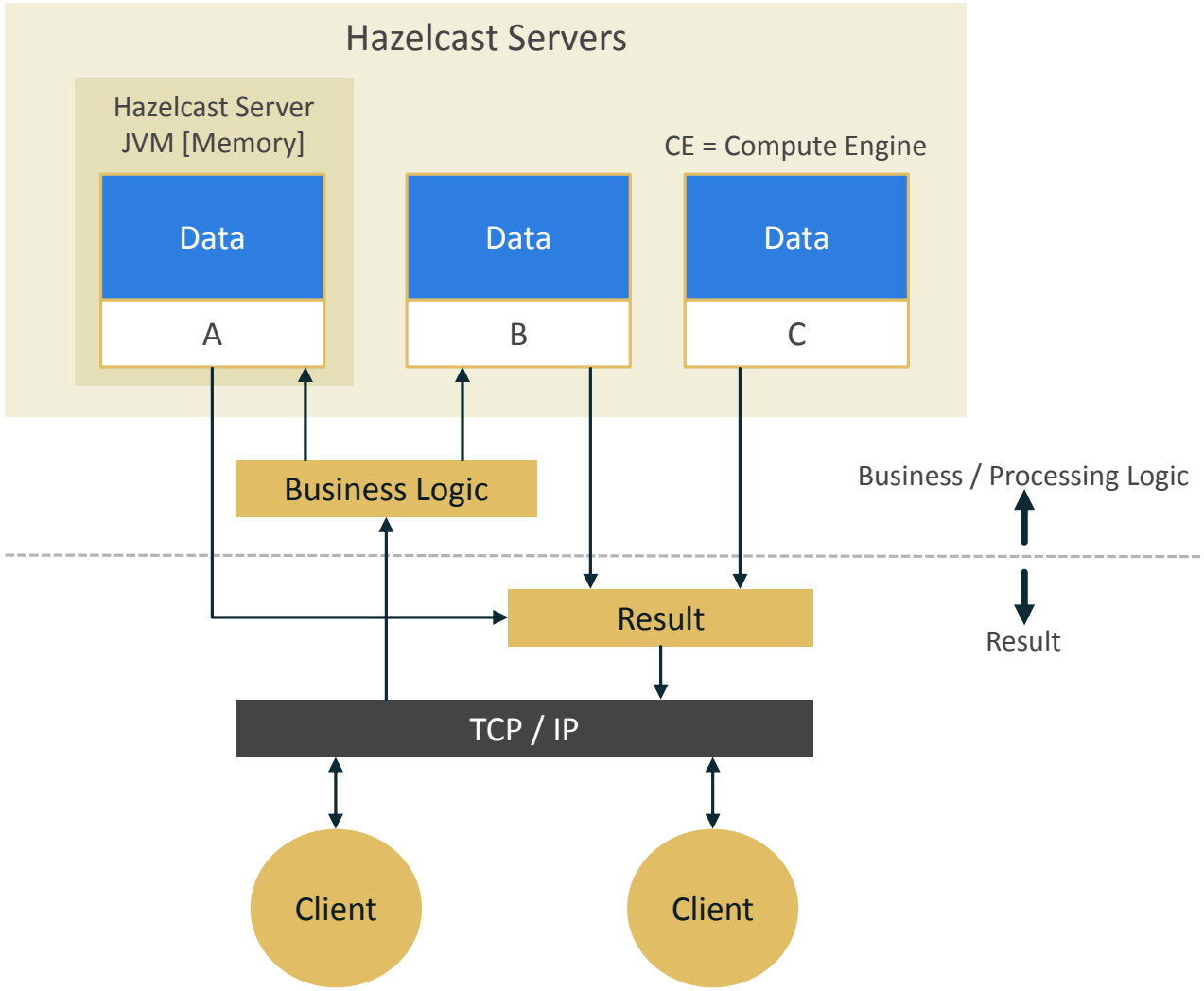


In Memory Data Messaging

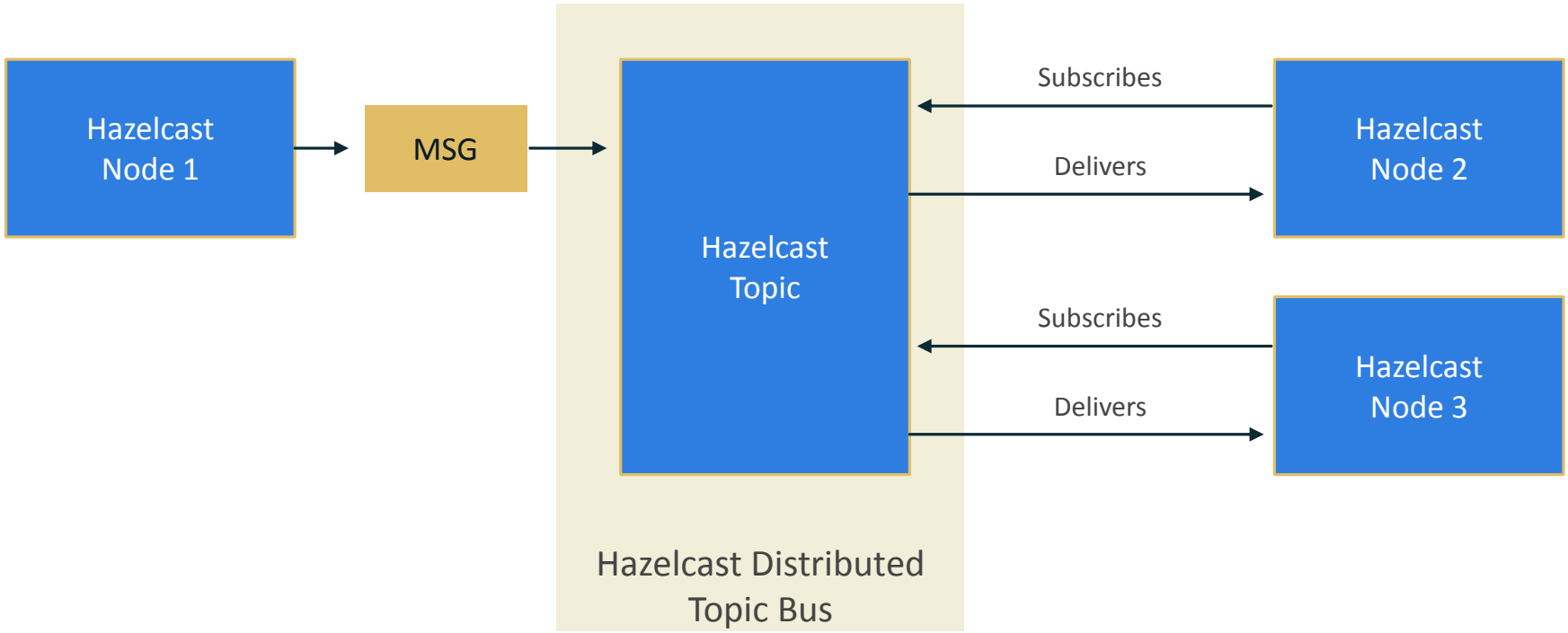
In-Memory Caching



Distributed Computing



Distributed Messaging



Data Distribution and Resilience

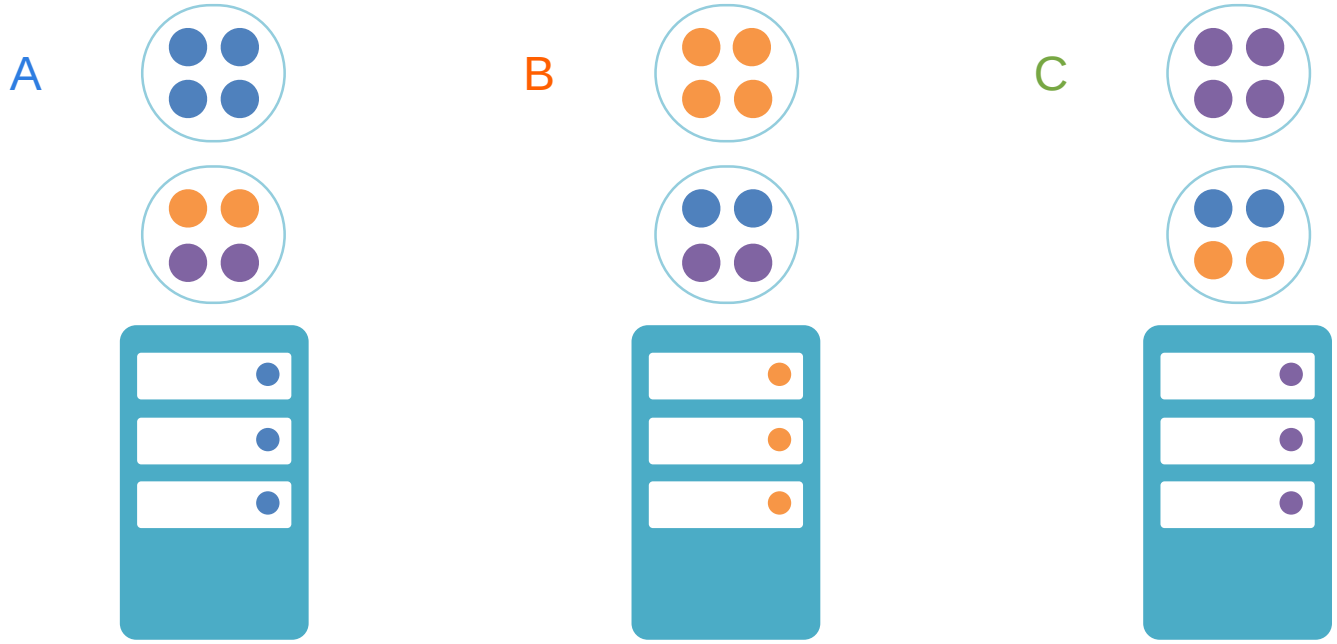
Distributed Maps

Fixed number of partitions (default 271)

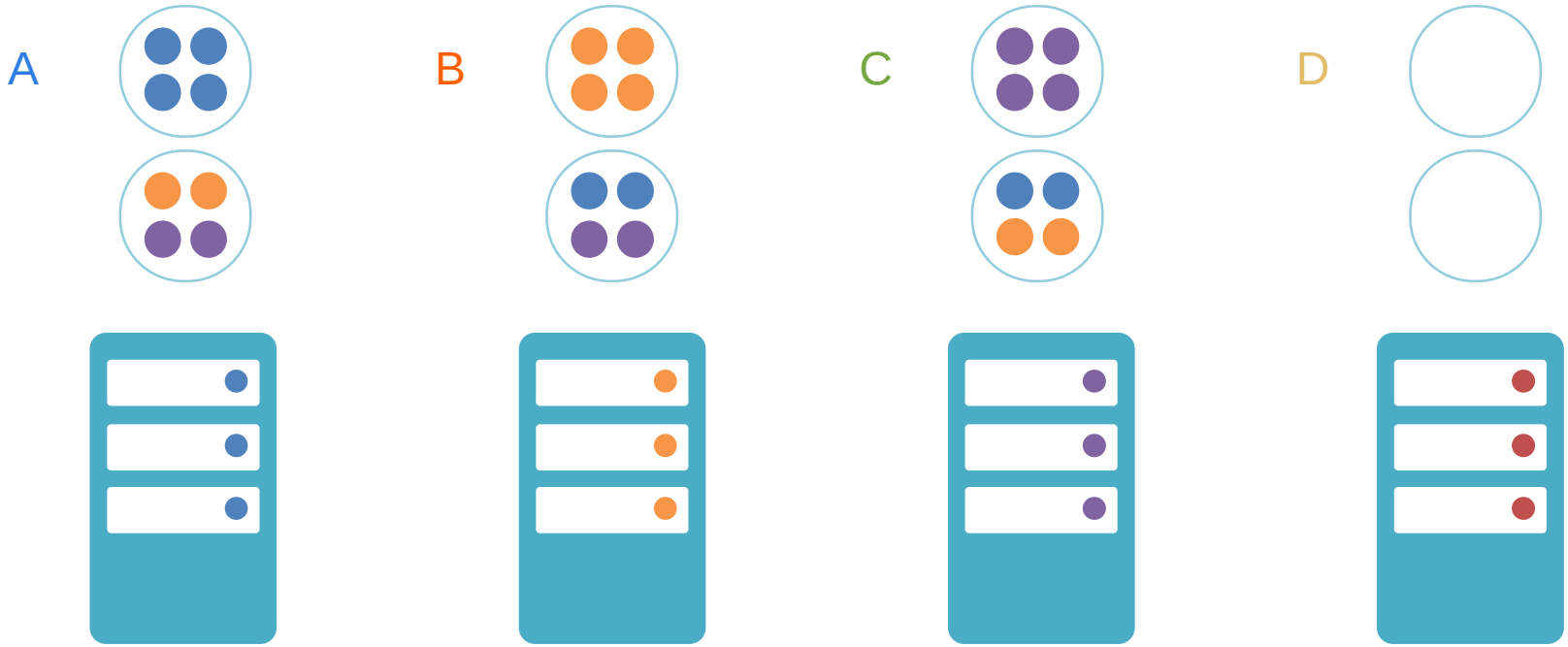
Each key falls into a partition

partitionId = hash(keyData)%PARTITION_COUNT

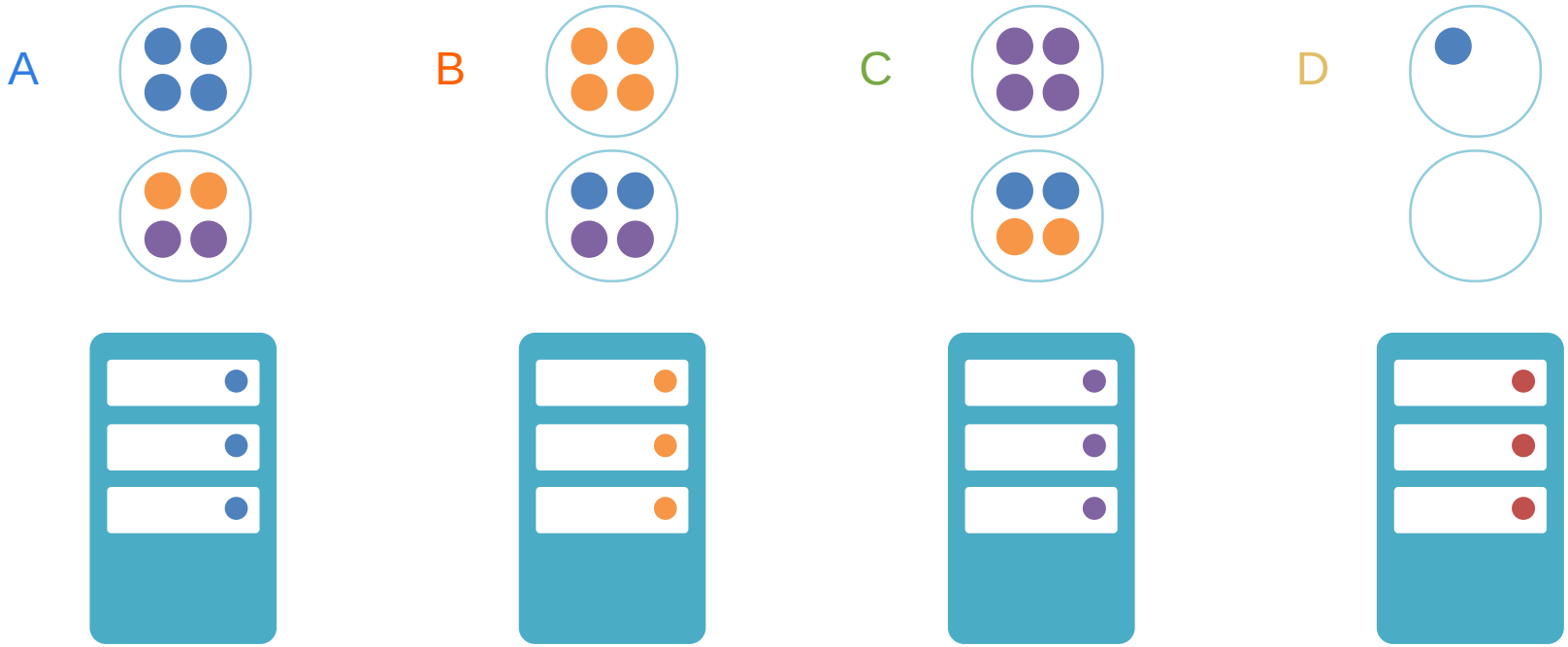
Partition ownerships are reassigned upon membership change



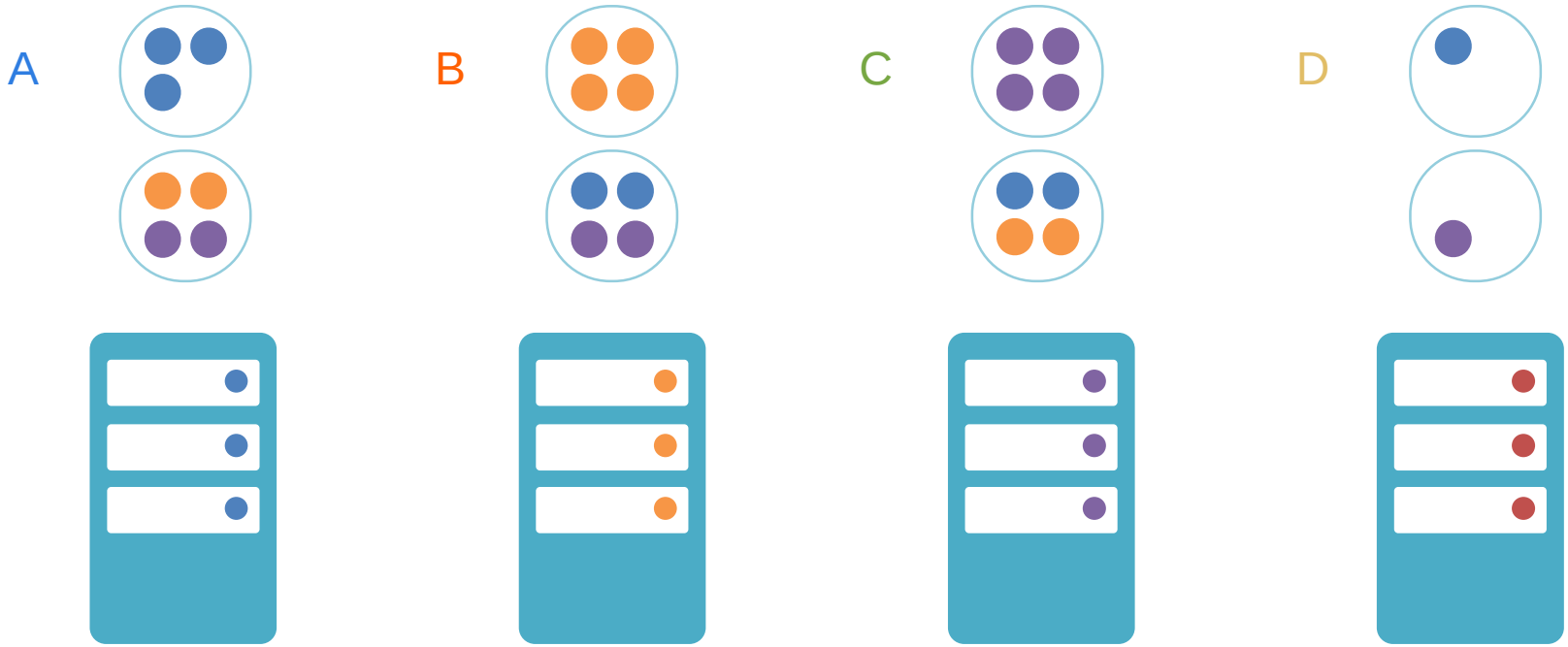
New Node Added



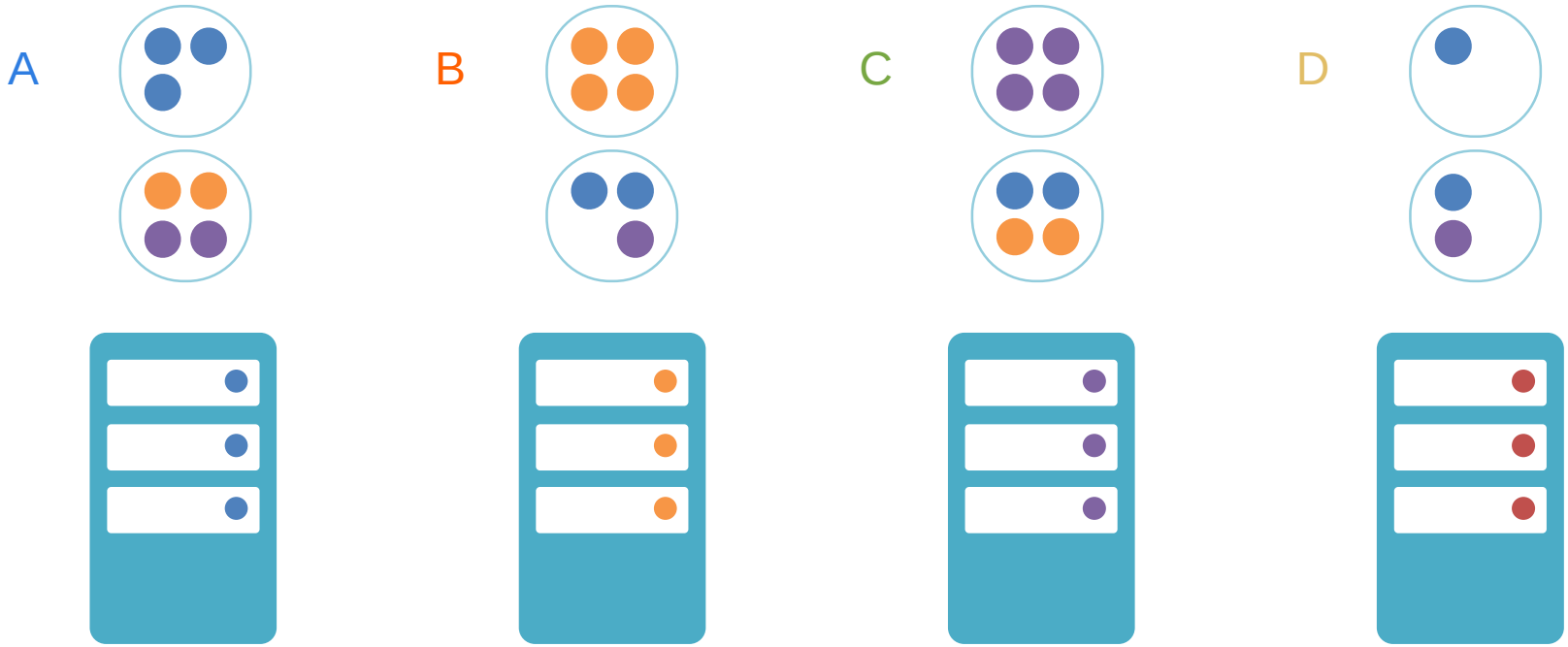
Migration



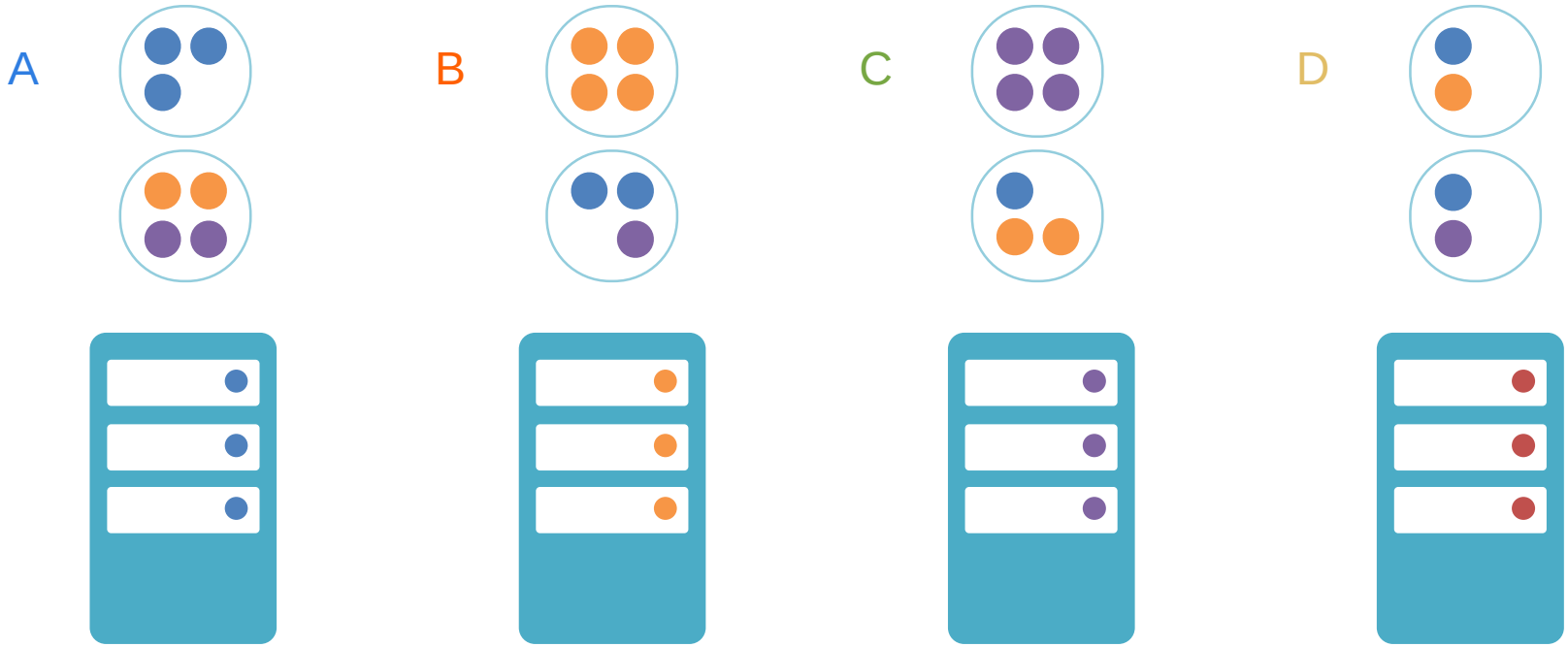
Migration



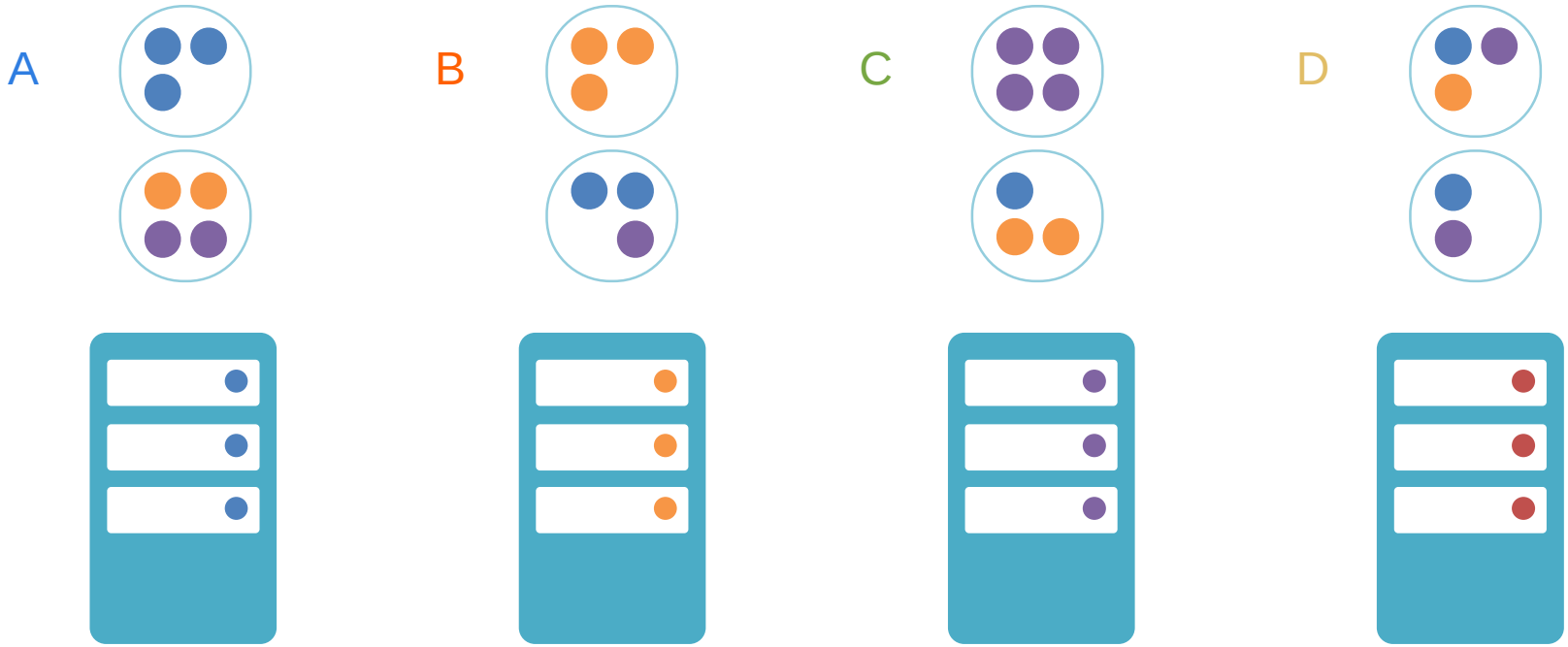
Migration



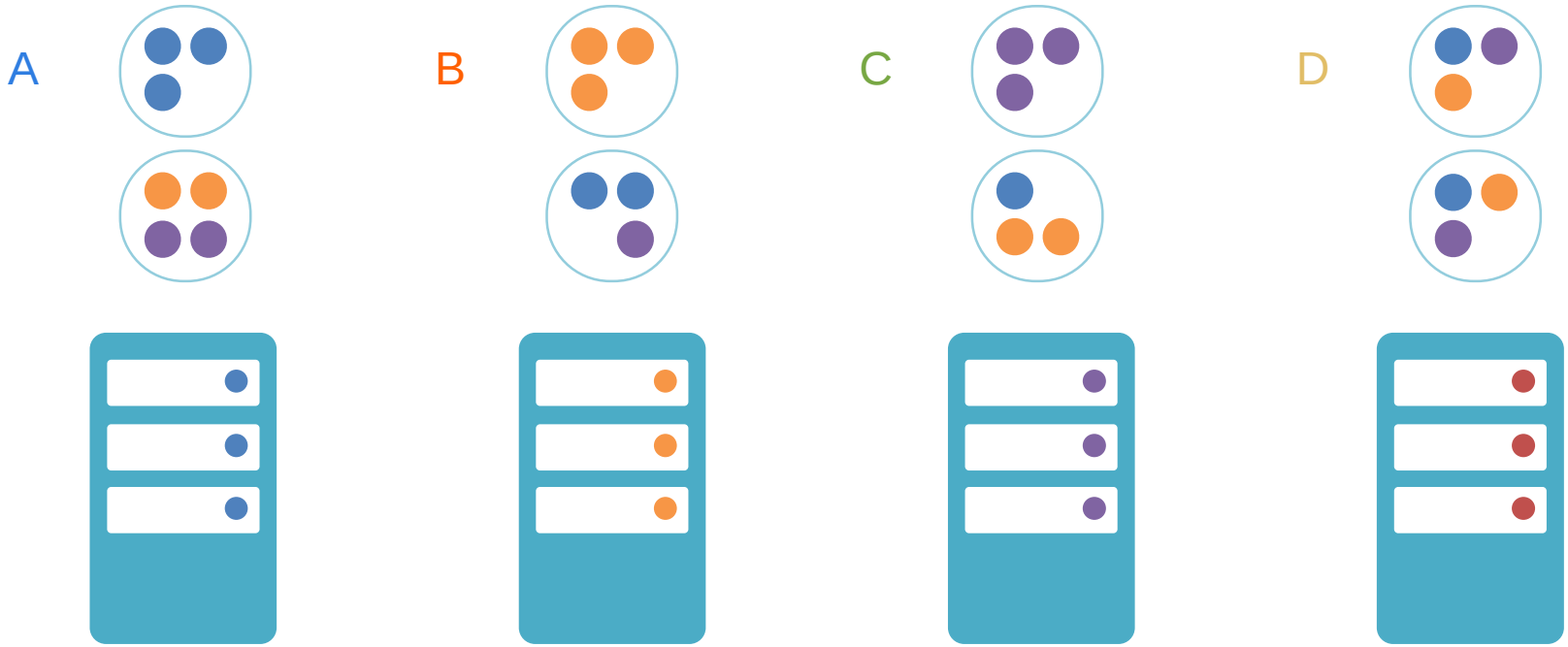
Migration



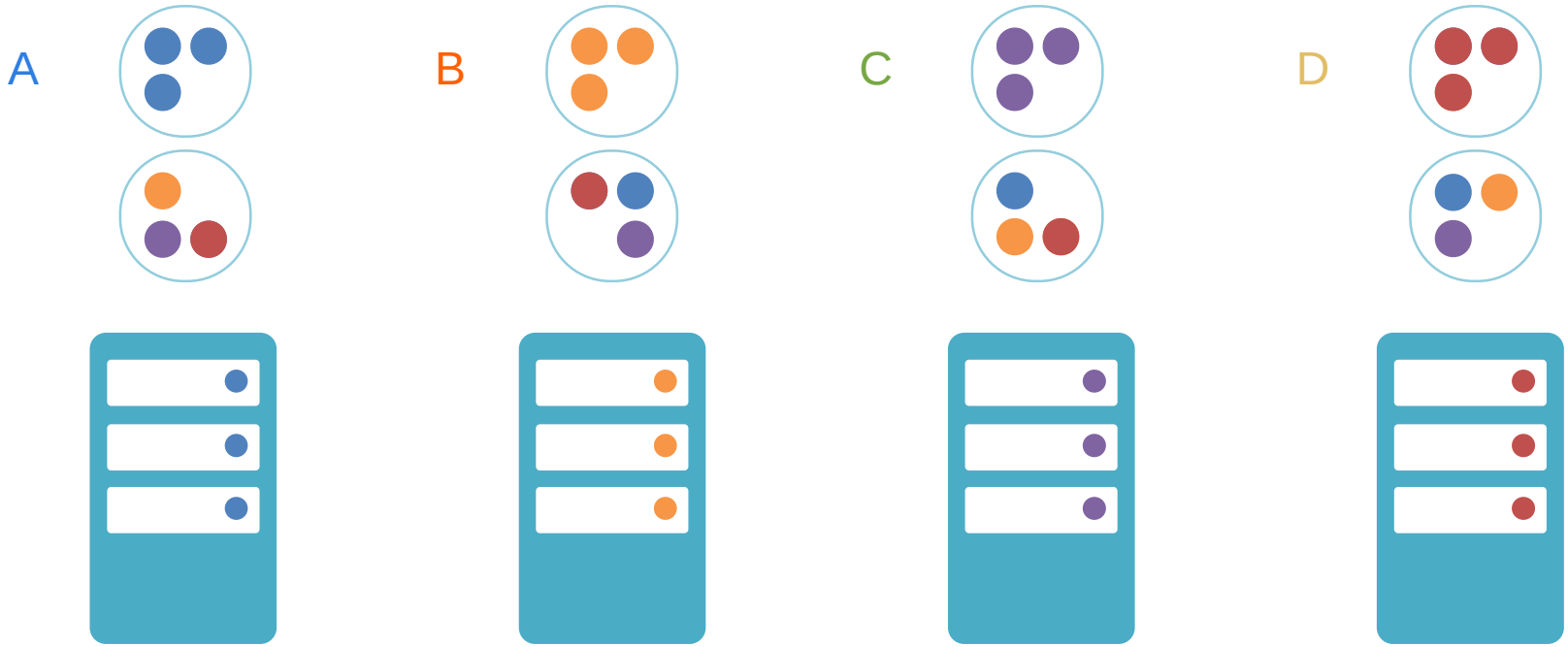
Migration



Migration

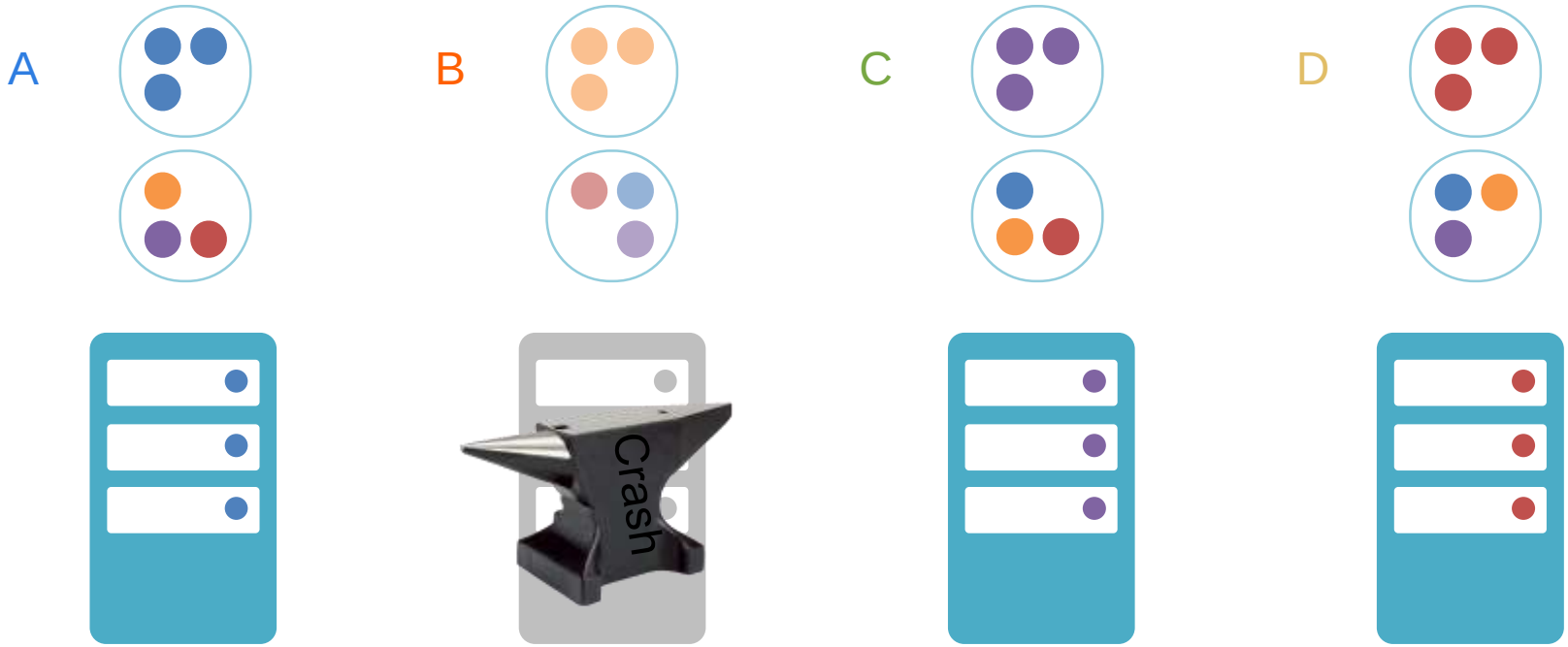


Migration Complete

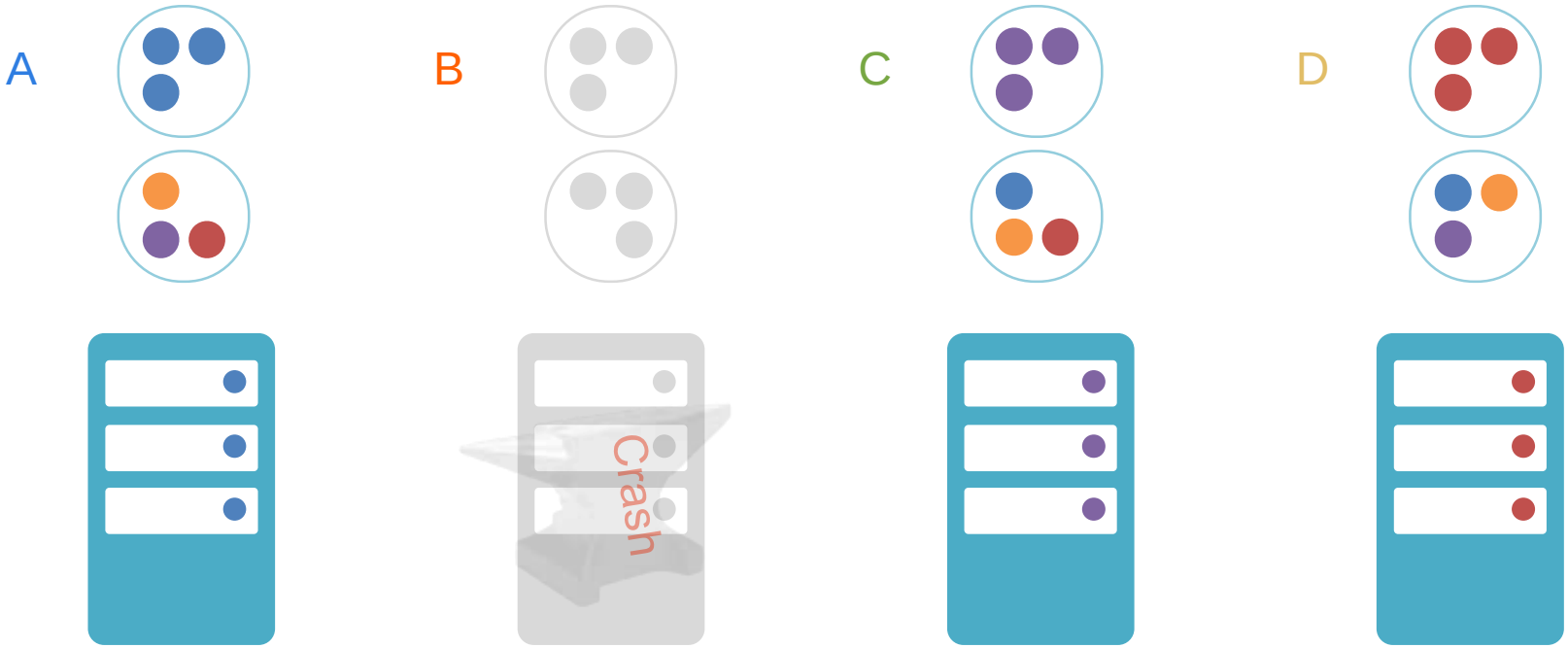


Data Safety on Node Failure

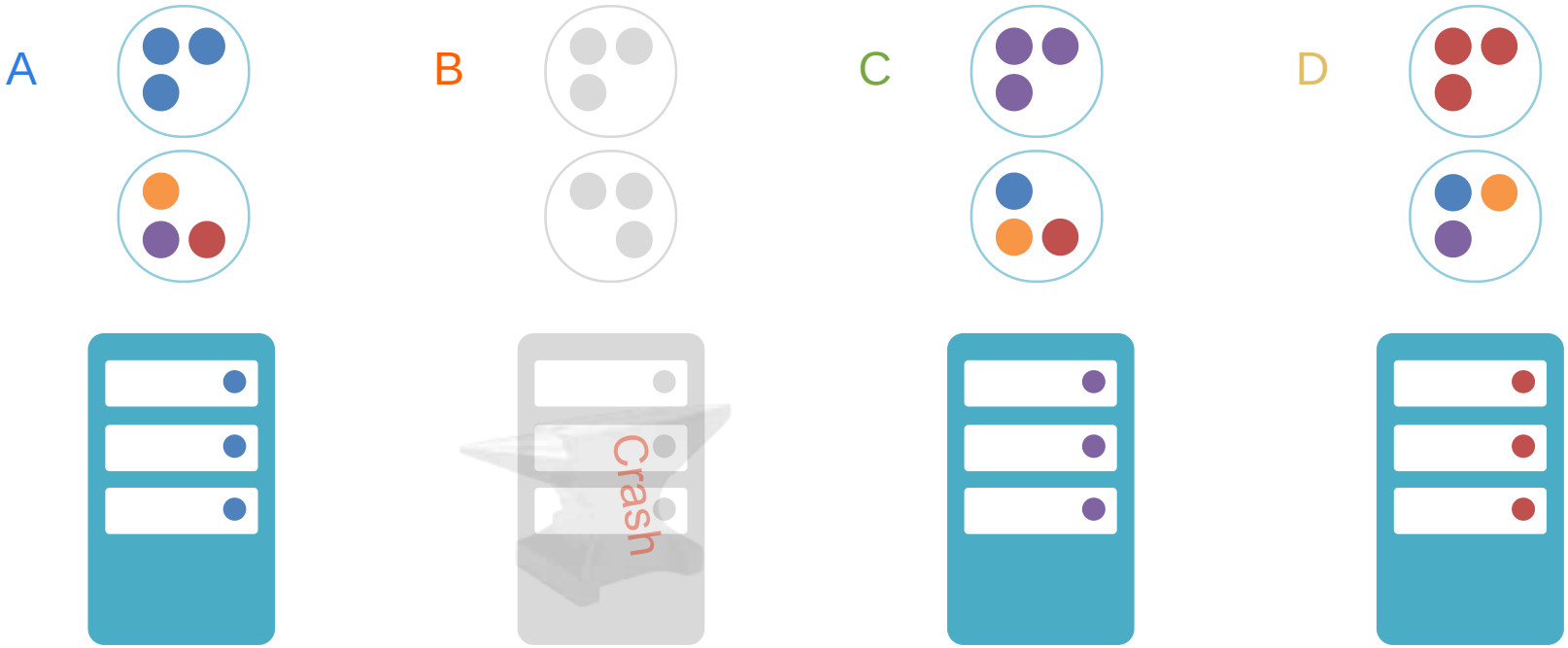
Node Crashes



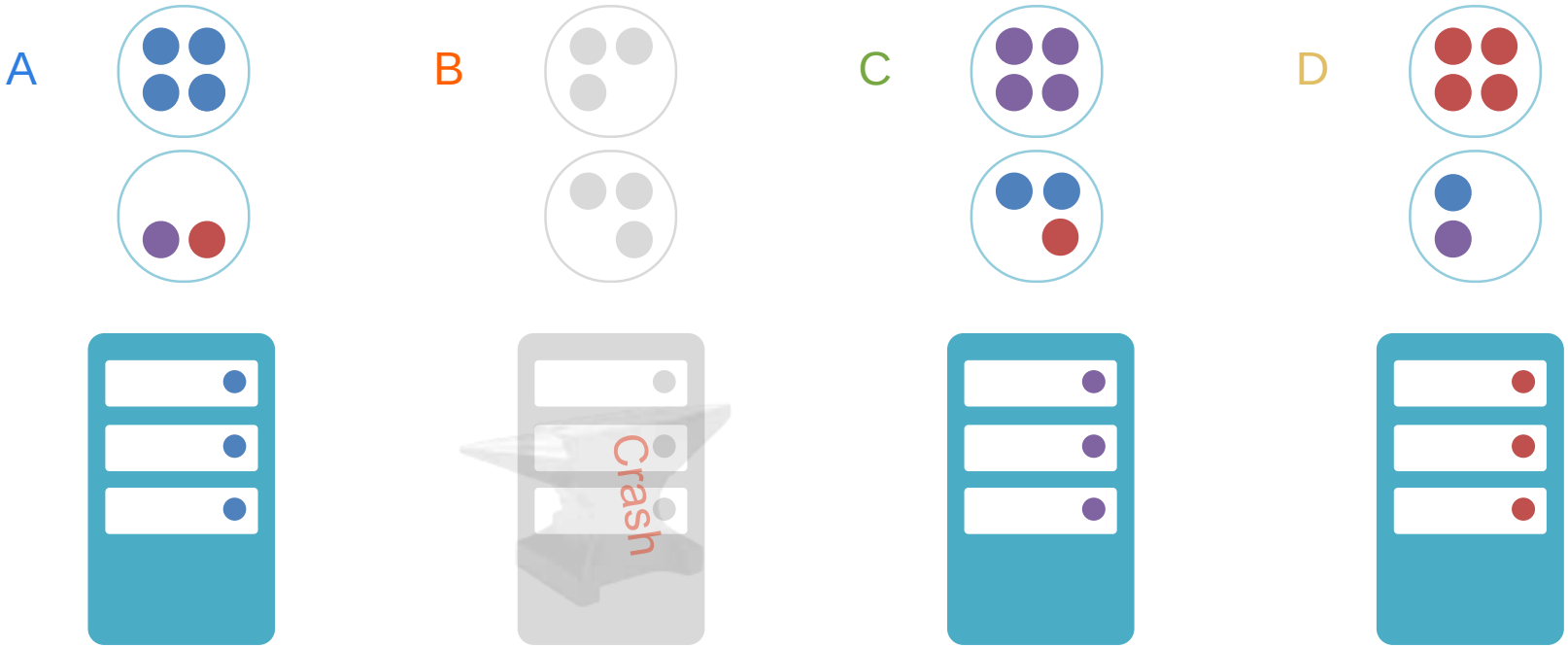
Backups Are Restored



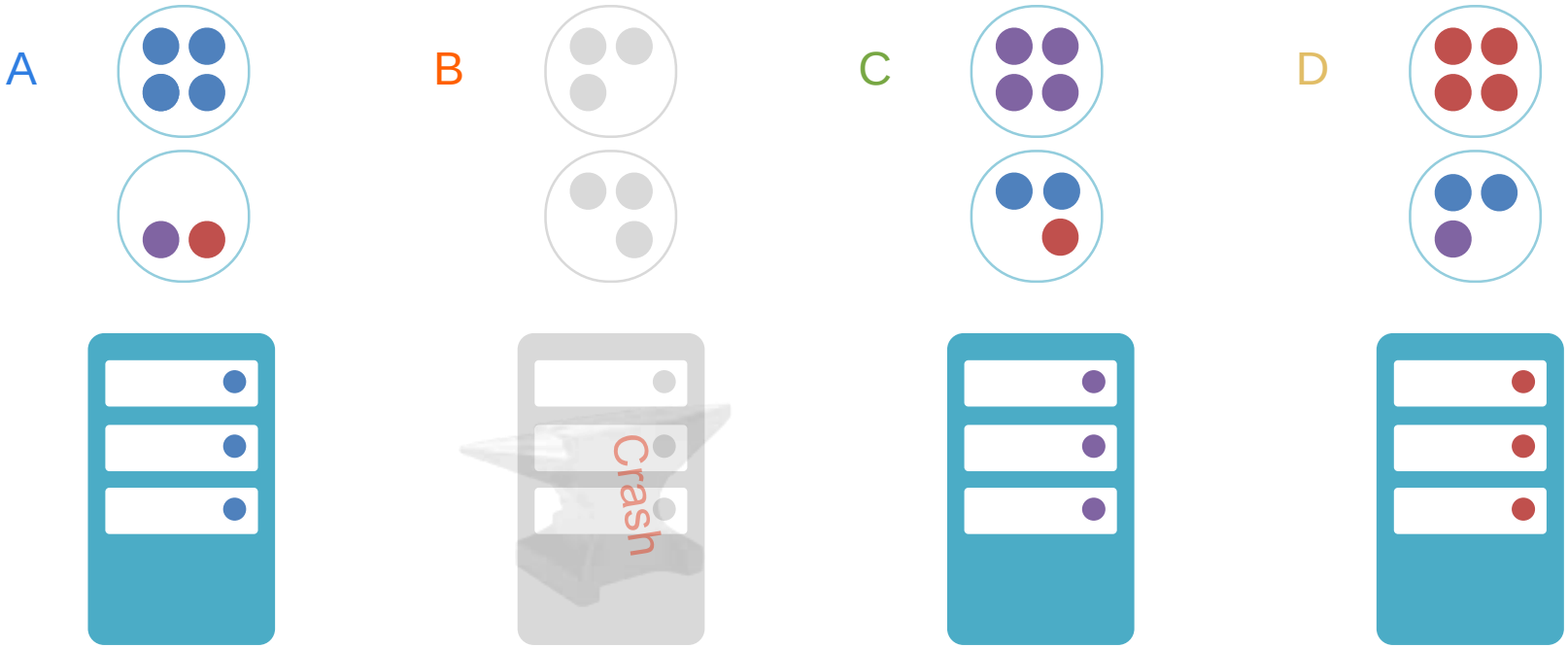
Backups Are Restored



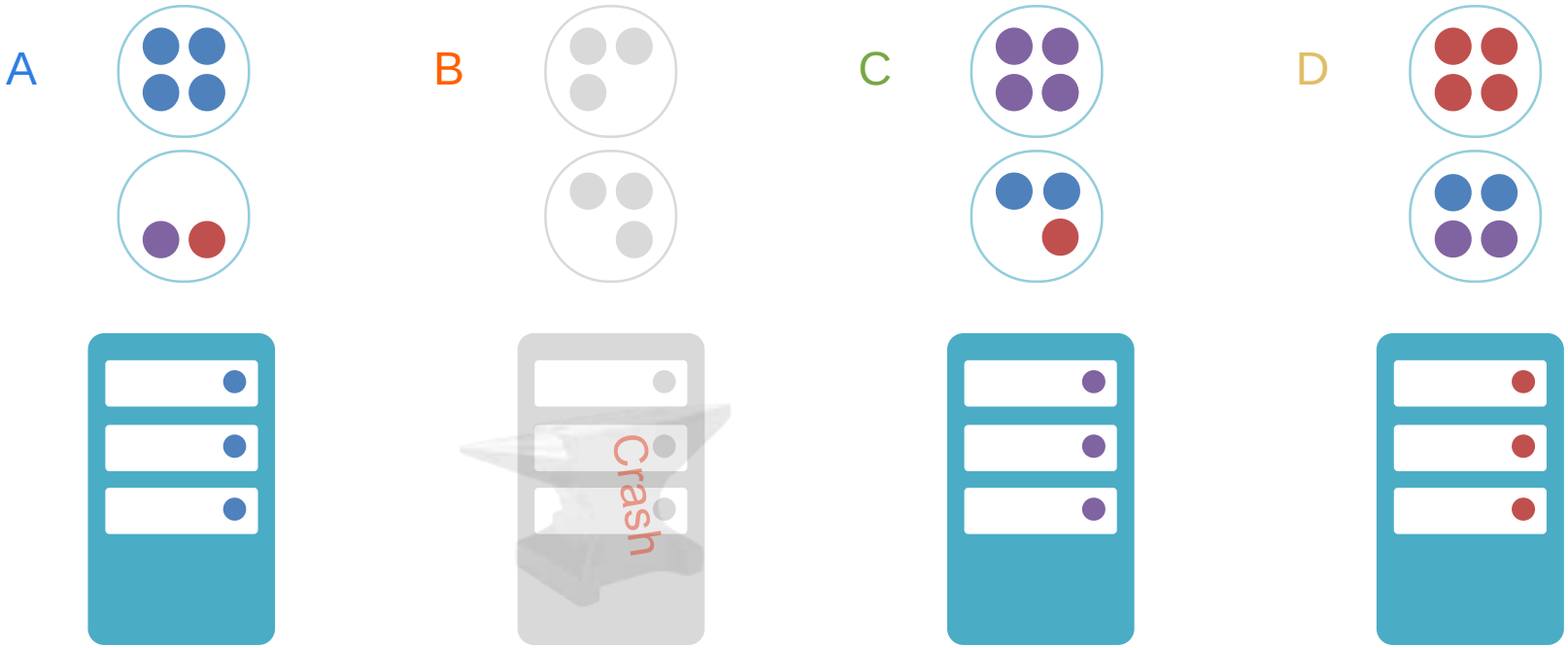
Backups Are Restored



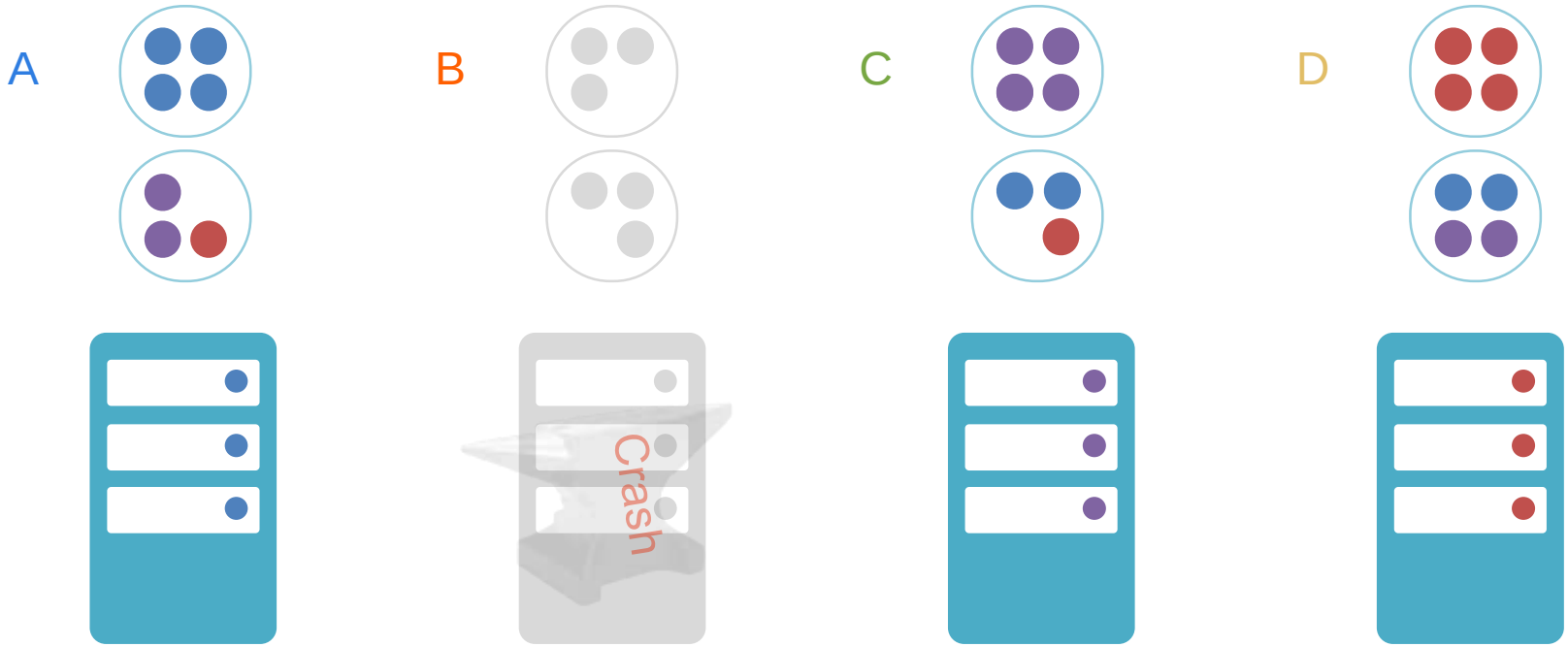
Backups Are Restored



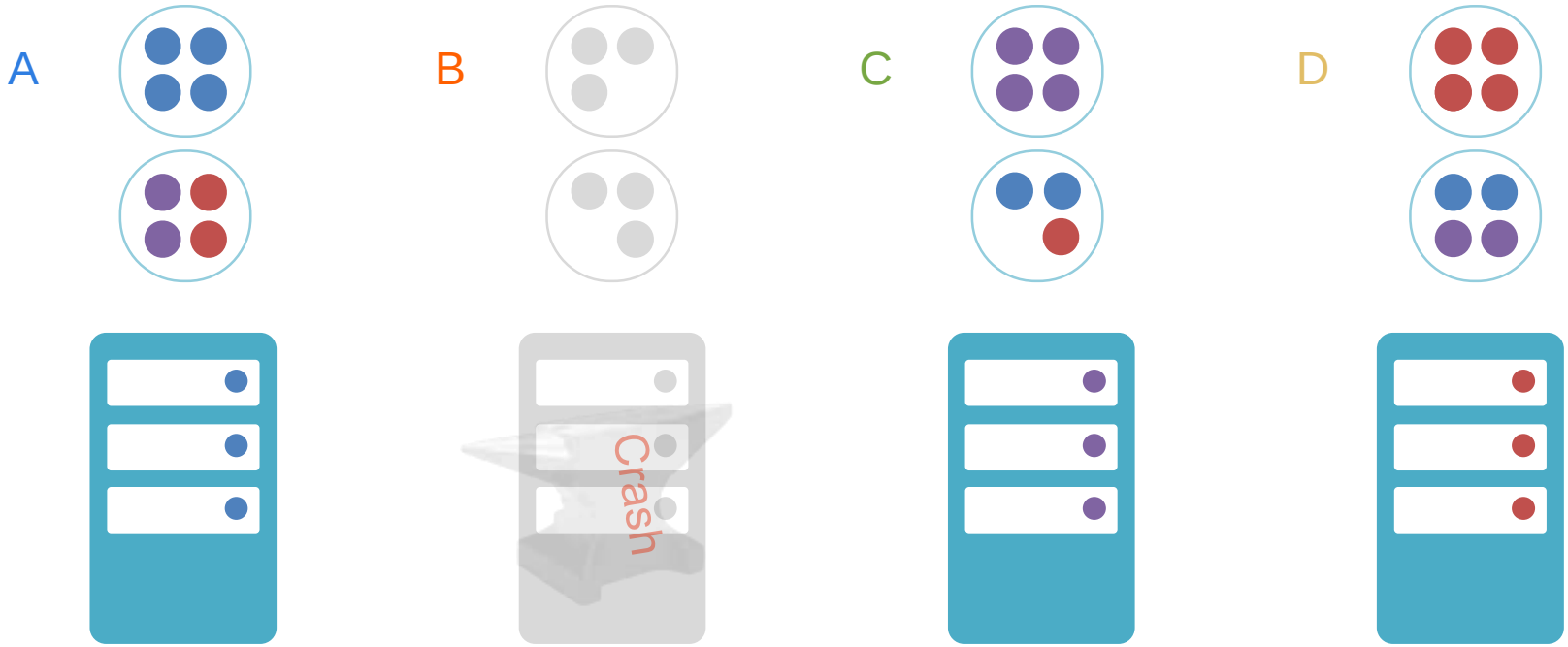
Backups Are Restored



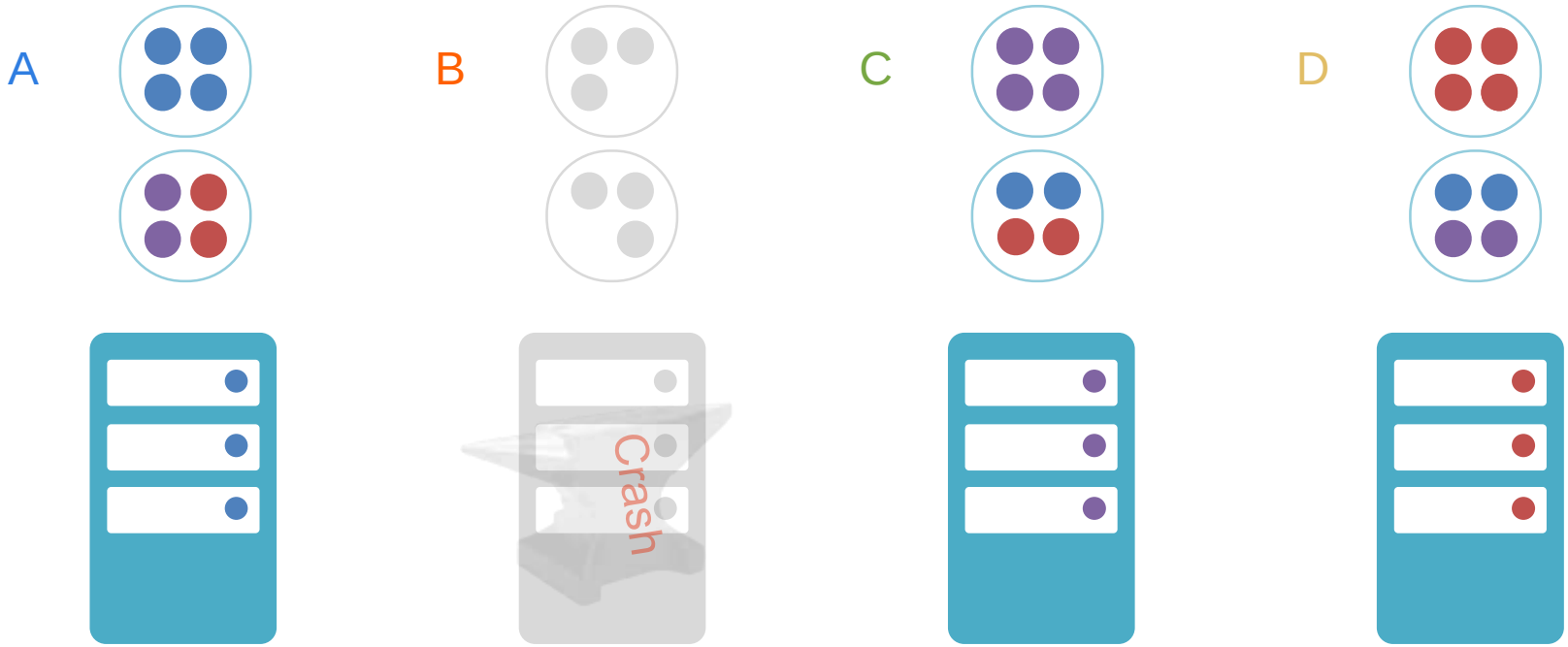
Backups Are Restored



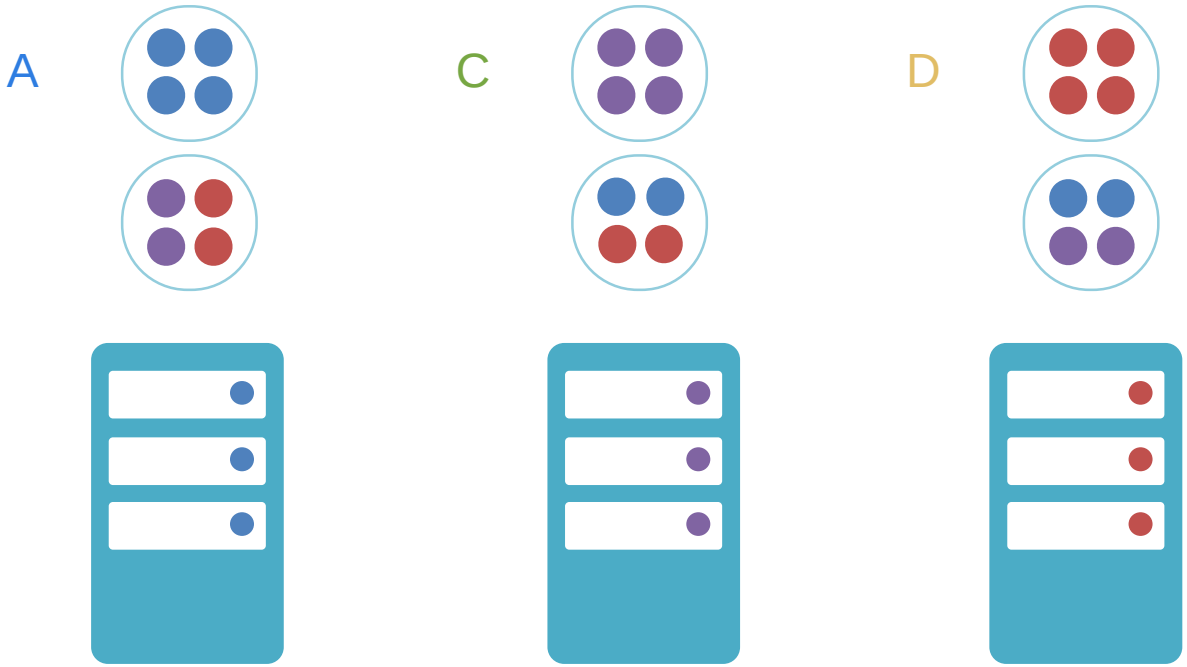
Backups Are Restored



Backups Are Restored

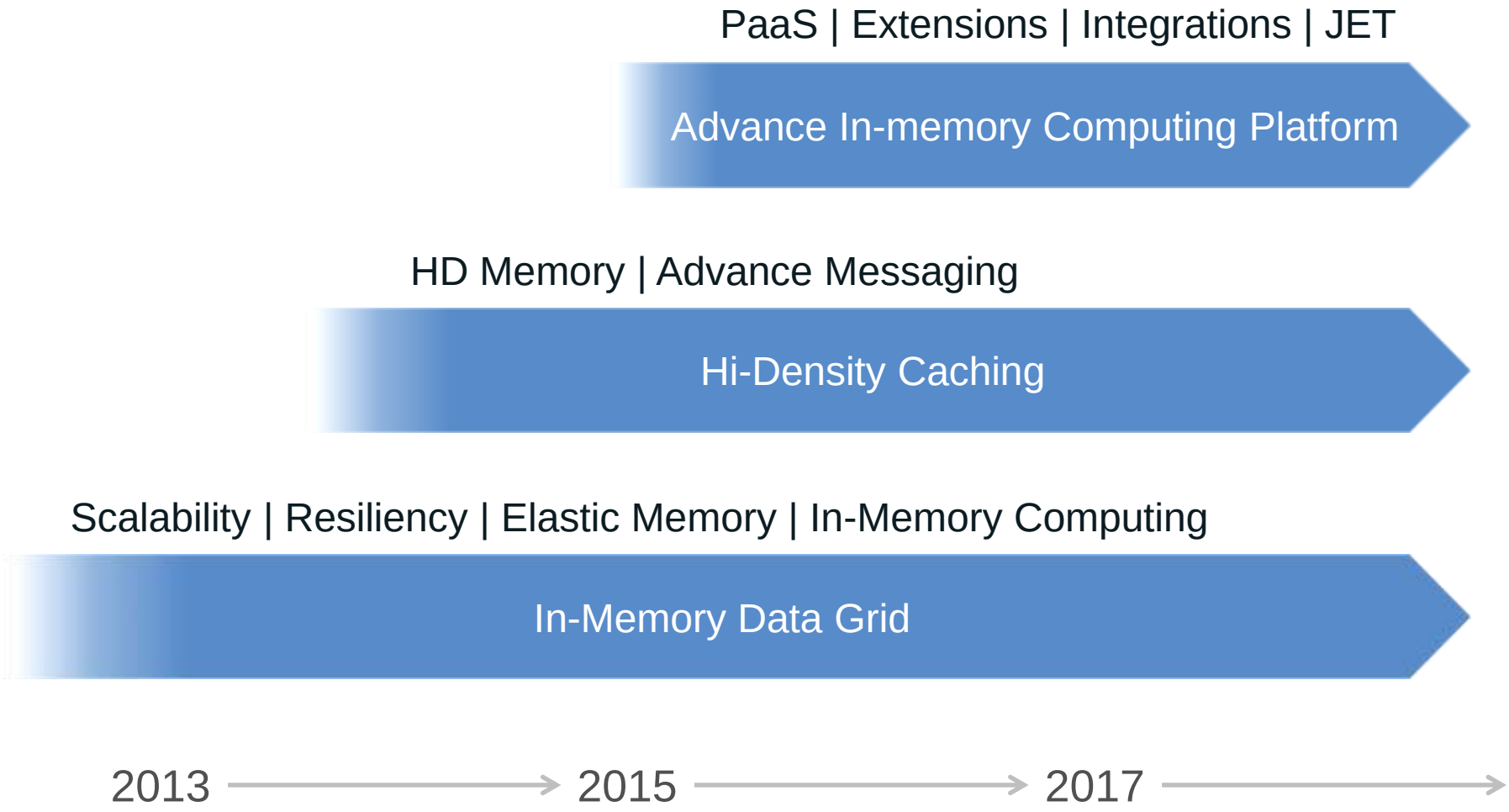


Recovery Is Complete

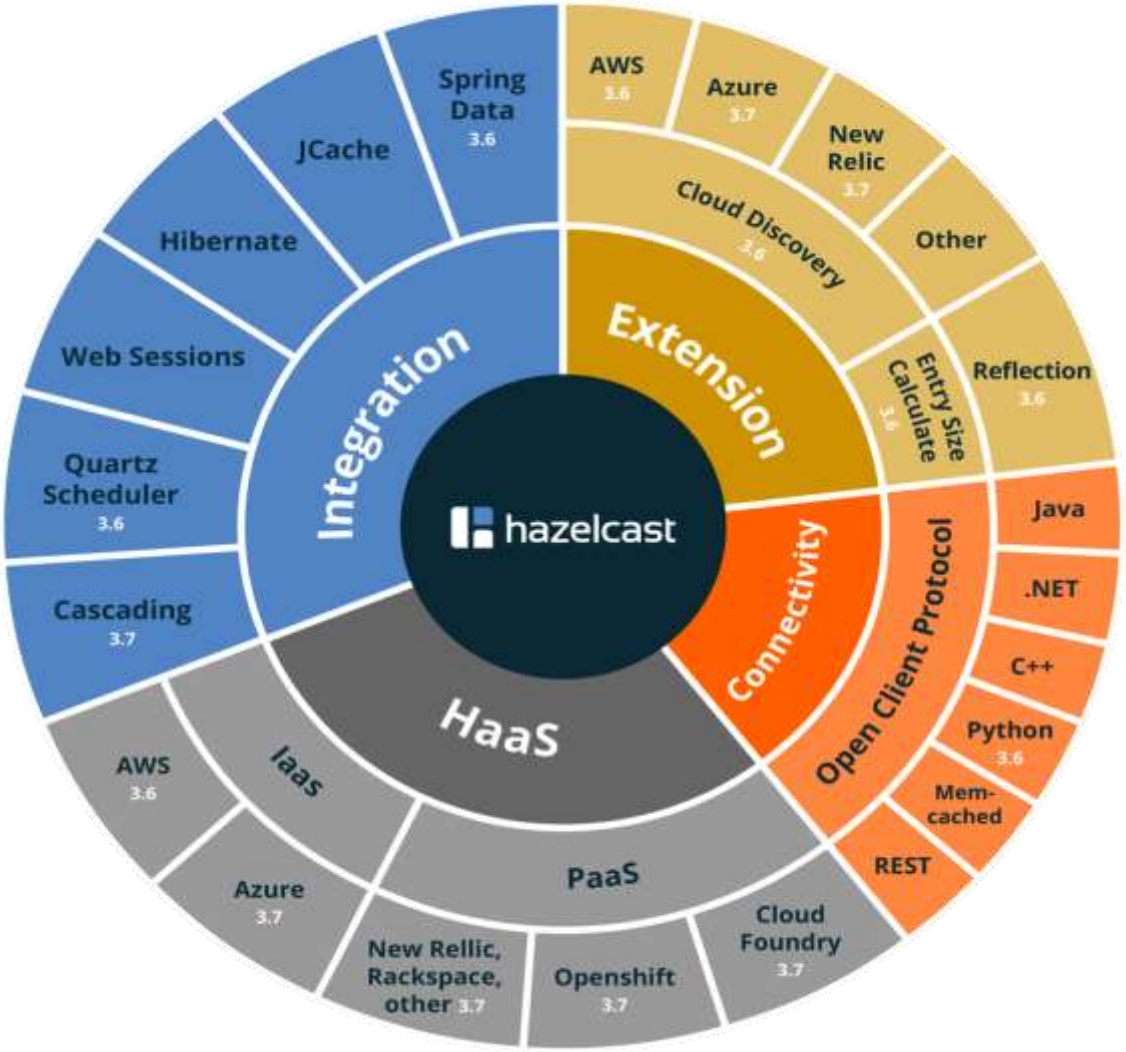


Roadmap and Latest

Hazelcast High Level Roadmap



Hazelcast Platform: Hazelcast Everywhere

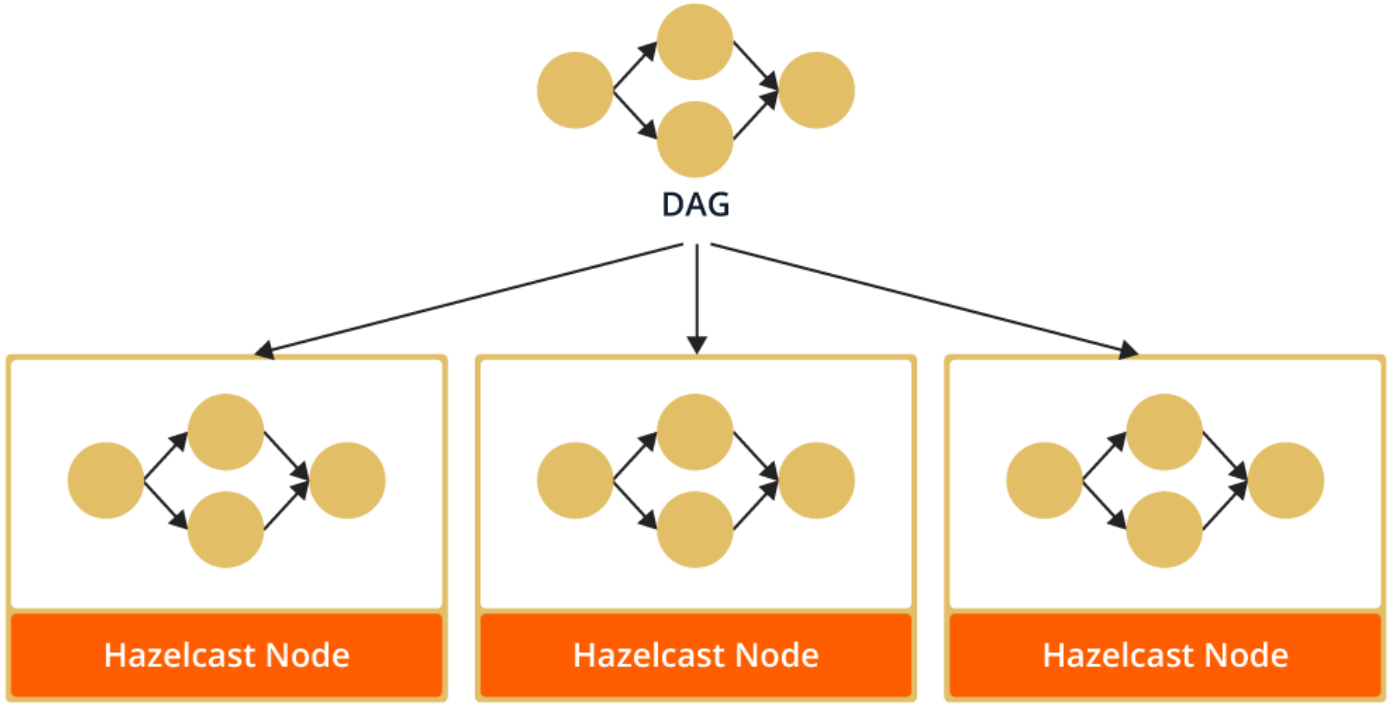



hazelcast JET

What’s Hazelcast Jet?

- General purpose distributed data processing framework
- Based on Direct Acyclic Graph to model data flow
- Built on top of Hazelcast
- Comparable to Apache Spark or Apache Flink

Job Execution



Hazelcast 3.7 Release

New Hazelcast 3.7 Features

Features	Description
Modularity	In 3.7, Hazelcast is converted to a modular system based around extension points. So clients, Cloud Discovery providers and integrations to third party systems like Hibernate etc will be released independently. 3.7 will then ship with the latest stable versions of each.
Redesign of Partition Migration	More robust partition migration to round out some edge cases.
Graceful Shutdown Improvements	More robust shutdown with partition migration on shutdown of a member
Higher Networking Performance	A further 30% improvement in performance across the cluster by eliminating notifyAll() calls.
Map.putAll() Performance Speedup	Implement member batching.

New Hazelcast 3.7 Features

Features	Description
Rule Based Query Optimizer	Make queries significantly faster by using static transformations of queries.
Azul Certification	Run Hazelcast on Azul Zing for Java 6, 7 or 8 for less variation of latencies due to GC.
Solaris Sparc Support	Align HD Memory backed data structure's layouts so that platforms, such as SPARC work. Verify SPARC using our lab machine.
New Features for JCache	Simple creation similar to other Hazelcast Data Structures. E.g.
Command Line Interface	New command line interface for common operations performed by Operations.
Non-blocking Vert.x integration	New async methods in Map and integration with Vert.x to use them.

New Hazelcast 3.7 Clients and Languages

Features	Description
	Scala integration for Hazelcast members and Hazelcast client. Implements all Hazelcast features. Wraps the Java client for client mode and in embedded mode uses the Hazelcast member directly.
Node.js	Native client implementation using the Hazelcast Open Client protocol. Basic feature support.
Python	Native client implementation using the Hazelcast Open Client protocol. Supports most Hazelcast features.
Clojure	Clojure integration for Hazelcast members and Hazelcast client. Implements some Hazelcast features. Wraps the Java client for client mode and in embedded mode uses the Hazelcast member directly.

New Hazelcast 3.7 Cloud Features

Features	Description
Azure Marketplace	Ability to start Hazelcast instances on Docker environments easily. Provides Hazelcast, Hazelcast Enterprise and Management Center.
Azure Cloud Provider	Discover Provider for member discovery using Kubernetes. (Plugin)
AWS Marketplace	Deploy Hazelcast, Hazelcast Management Center and Hazelcast Enterprise clusters straight from the Marketplace.
Consul Cloud Provider	Discover Provider for member discovery for Consul (Plugin)
Etcd Cloud Provider	Discover Provider for member discovery for Etcd (Plugin)
Zookeeper Cloud Provider	Discover Provider for member discovery for Zookeeper (Plugin)
Eureka Cloud Provider	Discover Provider for member discovery for Eureka 1 from Netflix. (Plugin)
Docker Enhancements	Docker support for cloud provider plugins

Hazelcast Services

Service Offerings

Hazelcast (Apache Licensed)

- Professional Subscription – 24x7 support*

Hazelcast Enterprise Support

- Available with Hazelcast Enterprise software subscription - 24x7 support*

Additional Services

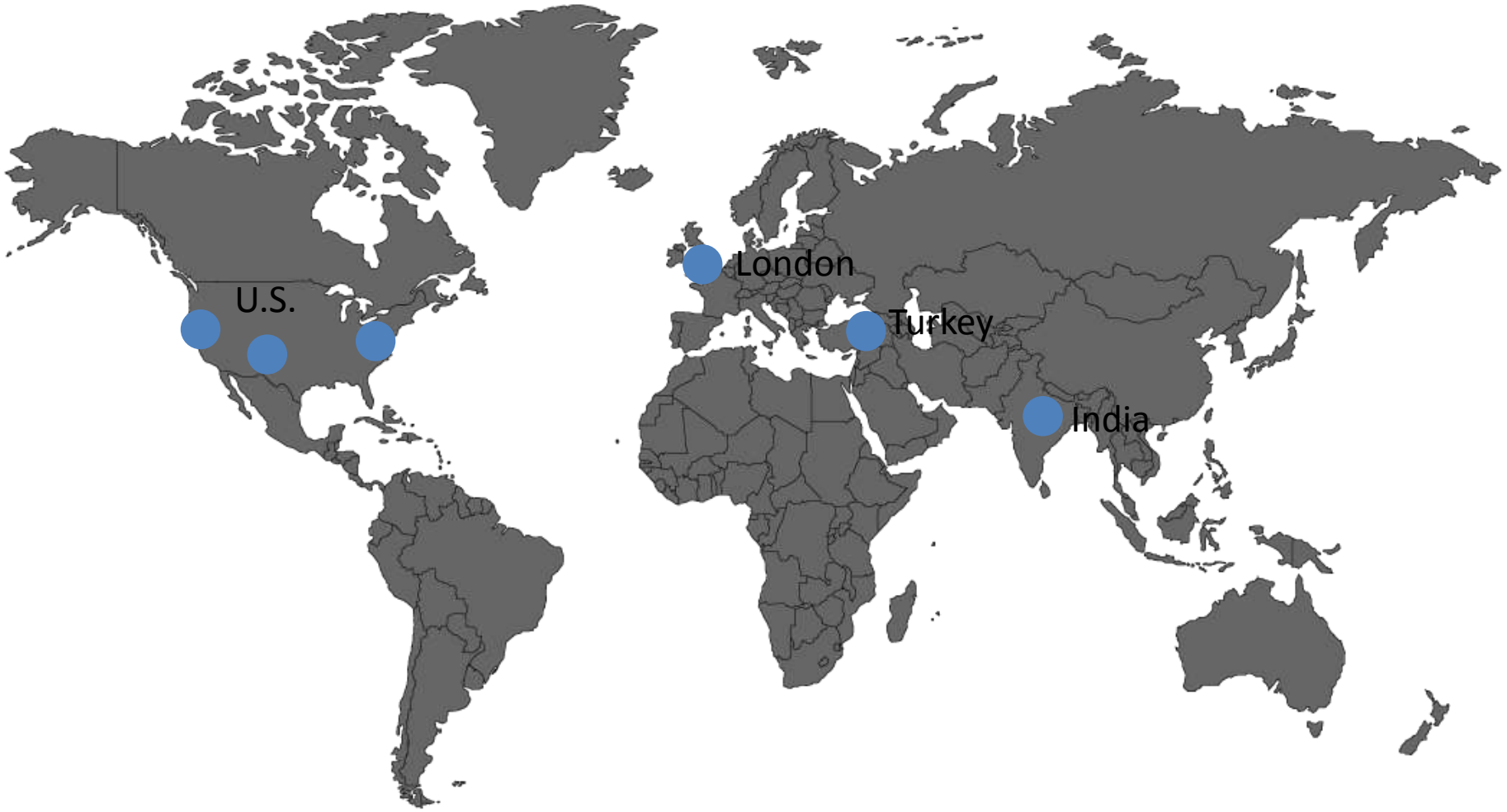
- Development Support Subscription – 8x5 support*
- Simulator TCK
- Training
- Expert Consulting
- Development Partner Program

* All subscriptions include Management Center

Best In Class Support

- Support from the Engineers who wrote the code
- SLA Driven – 100% attainment of support response time
- Follow the Sun
- Portal, Email and Phone access
- Go Red, Go Green. Reproduction of issues on Simulator. Proof of fix on Simulator.
- Periodic Technical Reviews
- Meet your production schedule and corporate compliance requirements
- Ensure the success of your development team with training and best practices

Hazelcast Support Coverage



Release Lifecycle

- **Regular Feature release** each 4-5 months, e.g. 3.3, 3.4, 3.5
- **Maintenance release** approximately each month with bug fixes based on the current feature release, e.g. 3.4.1
- **For older versions**, patch releases made available to fix issues
- **Release End of Life** per support contract

Thank you

rahul@hazelcast.com
chris.wilson@hazelcast.com