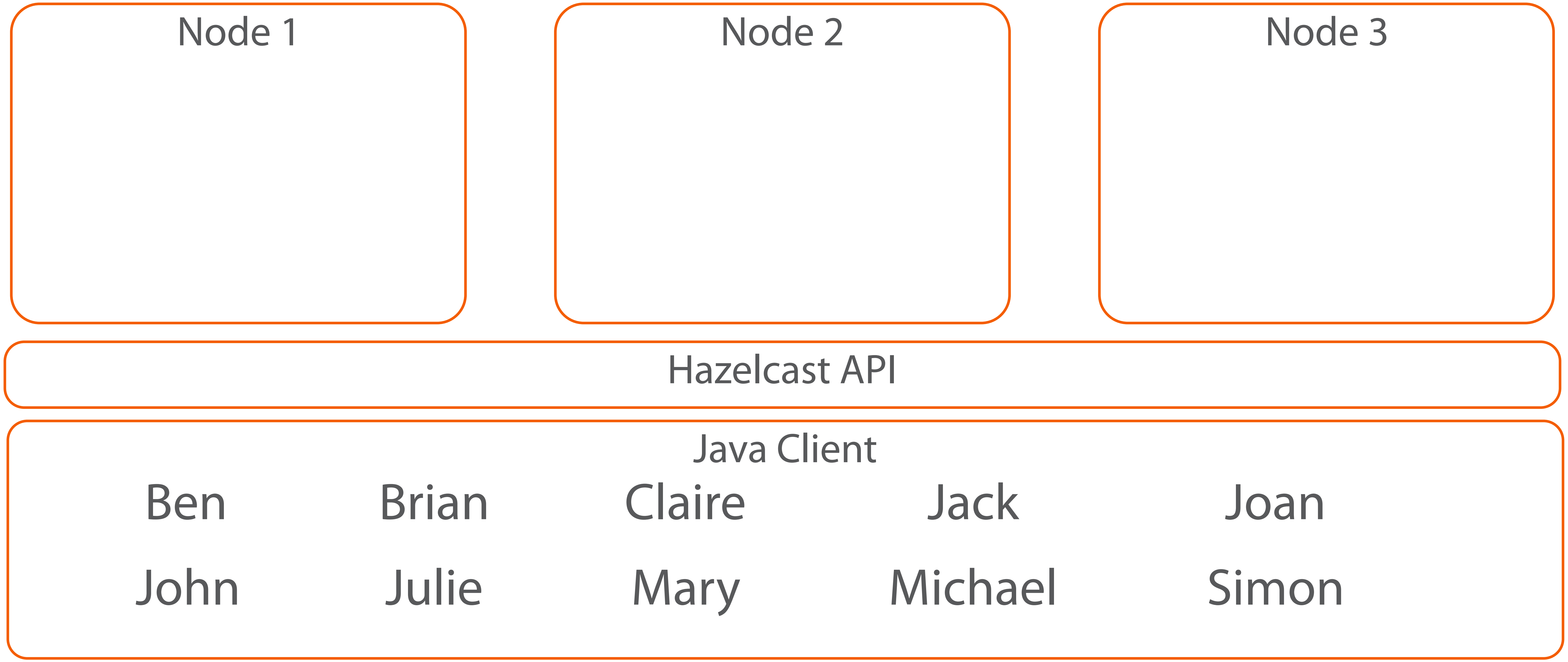
Basic Map Usage

Grant Little
http://www.grantlittle.me
grant@grantlittle.me

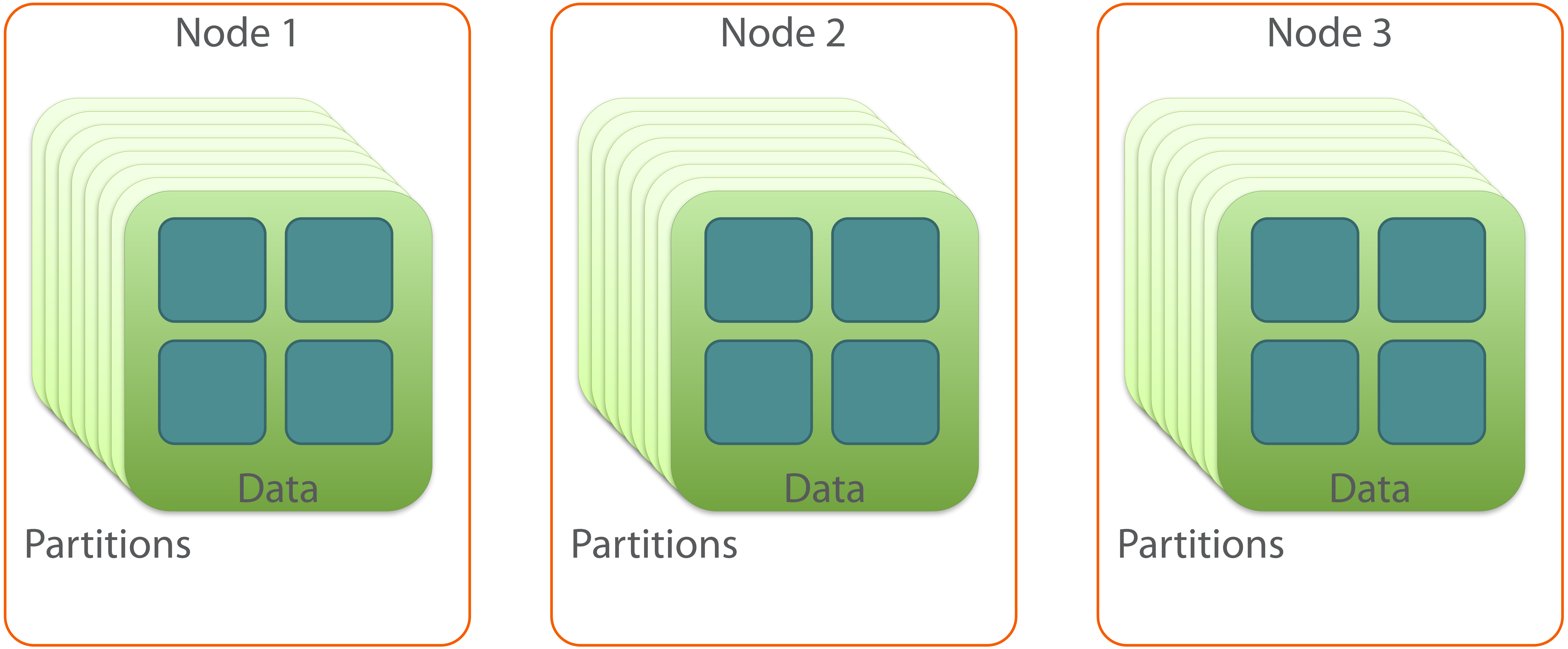
Overview

- What is a distributed map
- Adding data to a map
- Resiliency
- Persistent Storage
- Searching Maps and Predicates
- Indexing

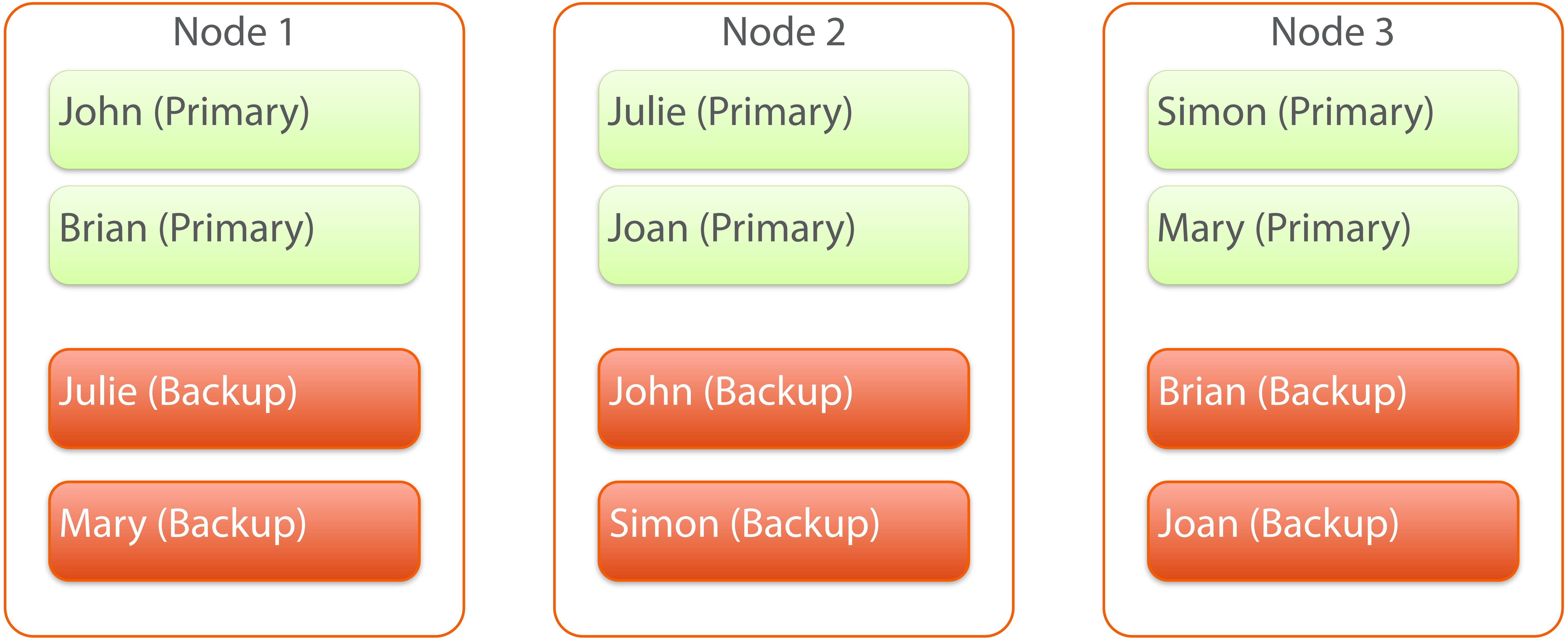
Distributed Map



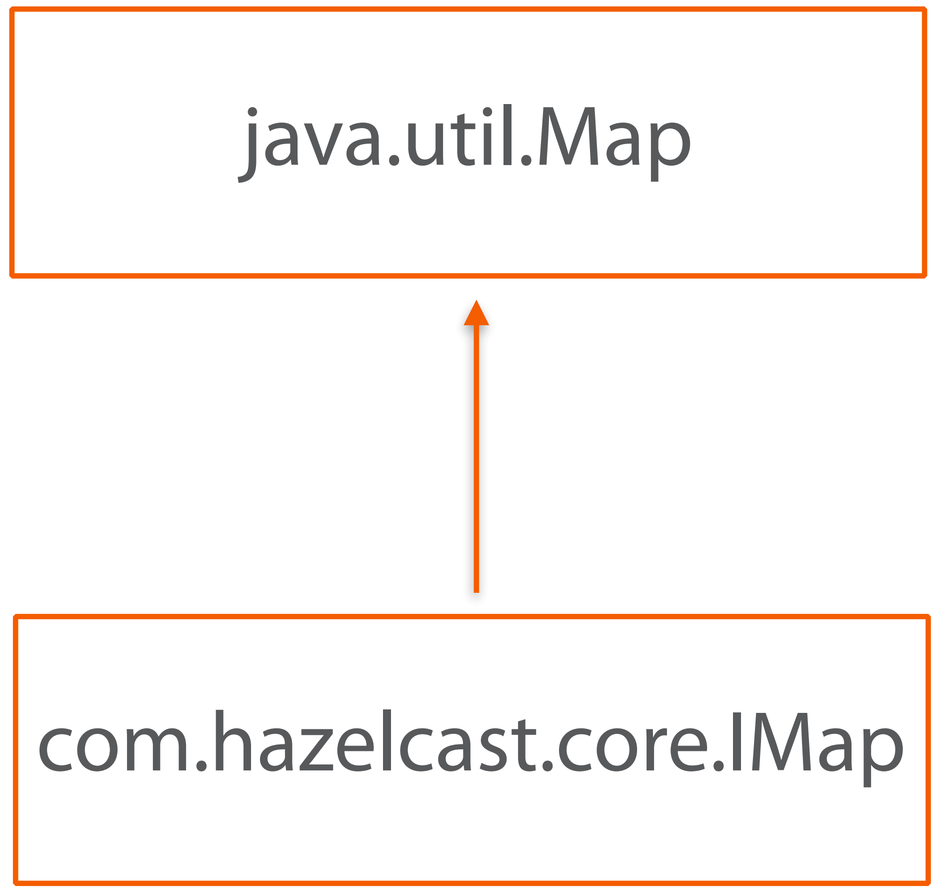
Data Distribution



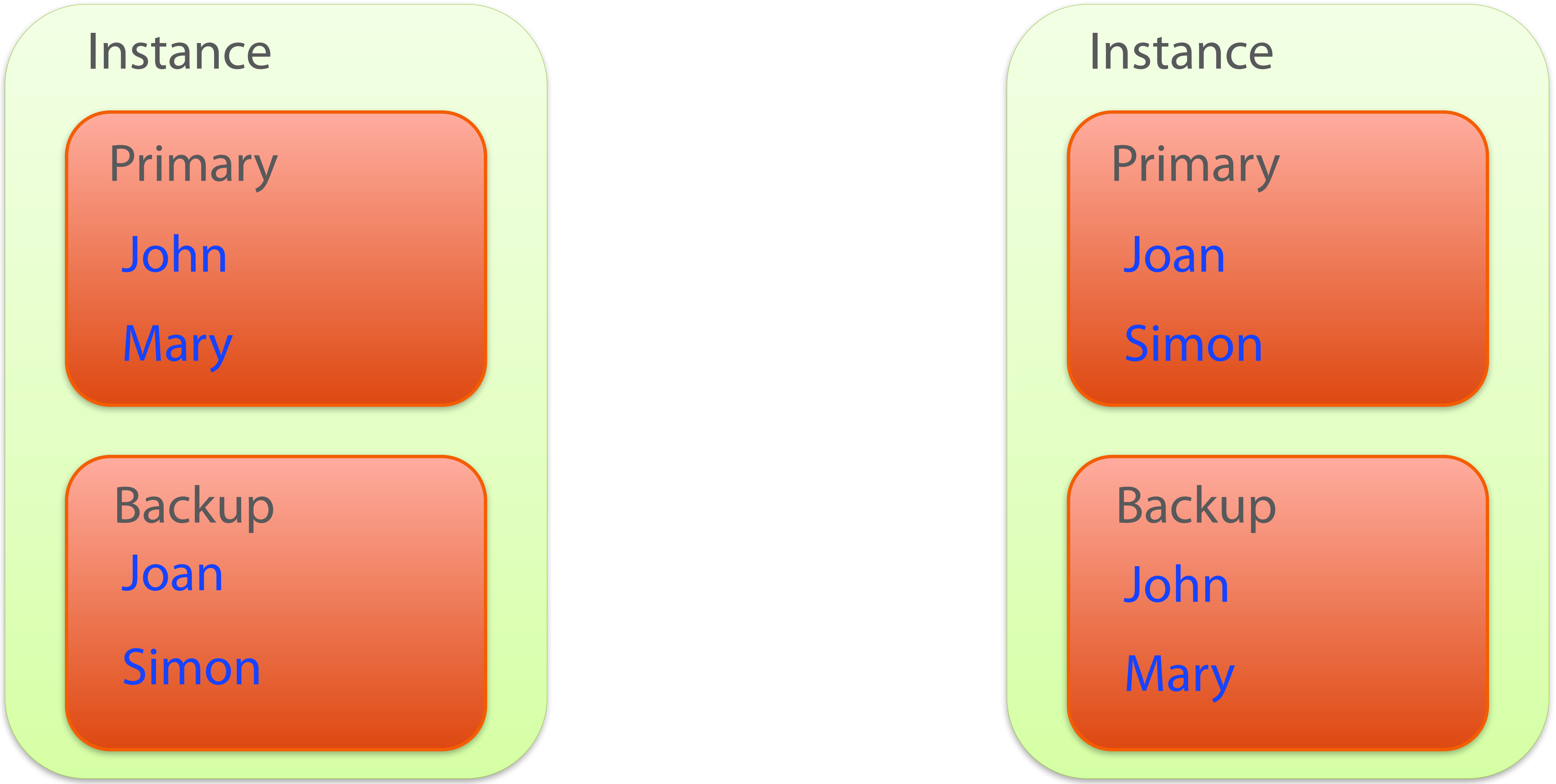
Data Backups



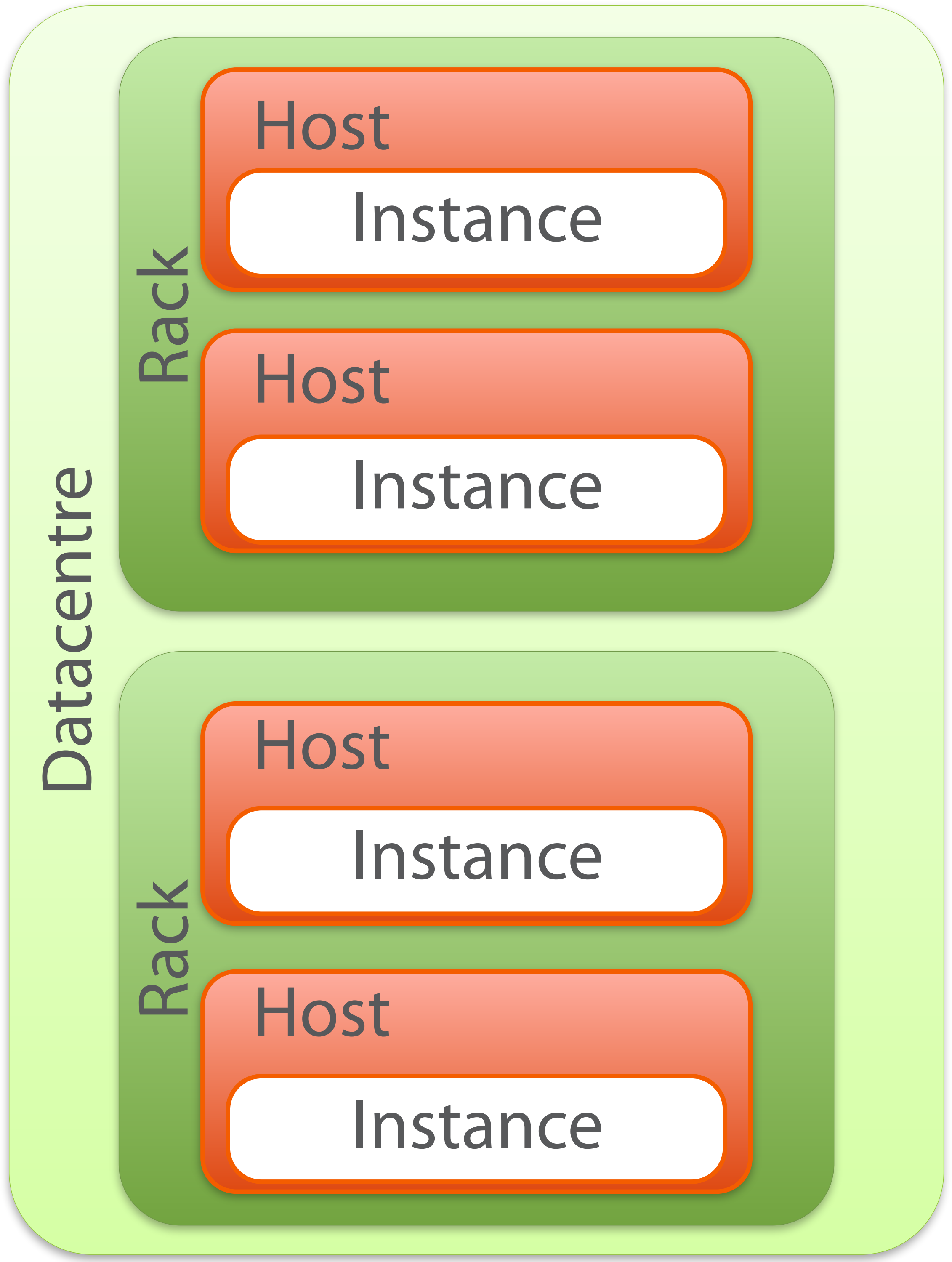
IMap Interface



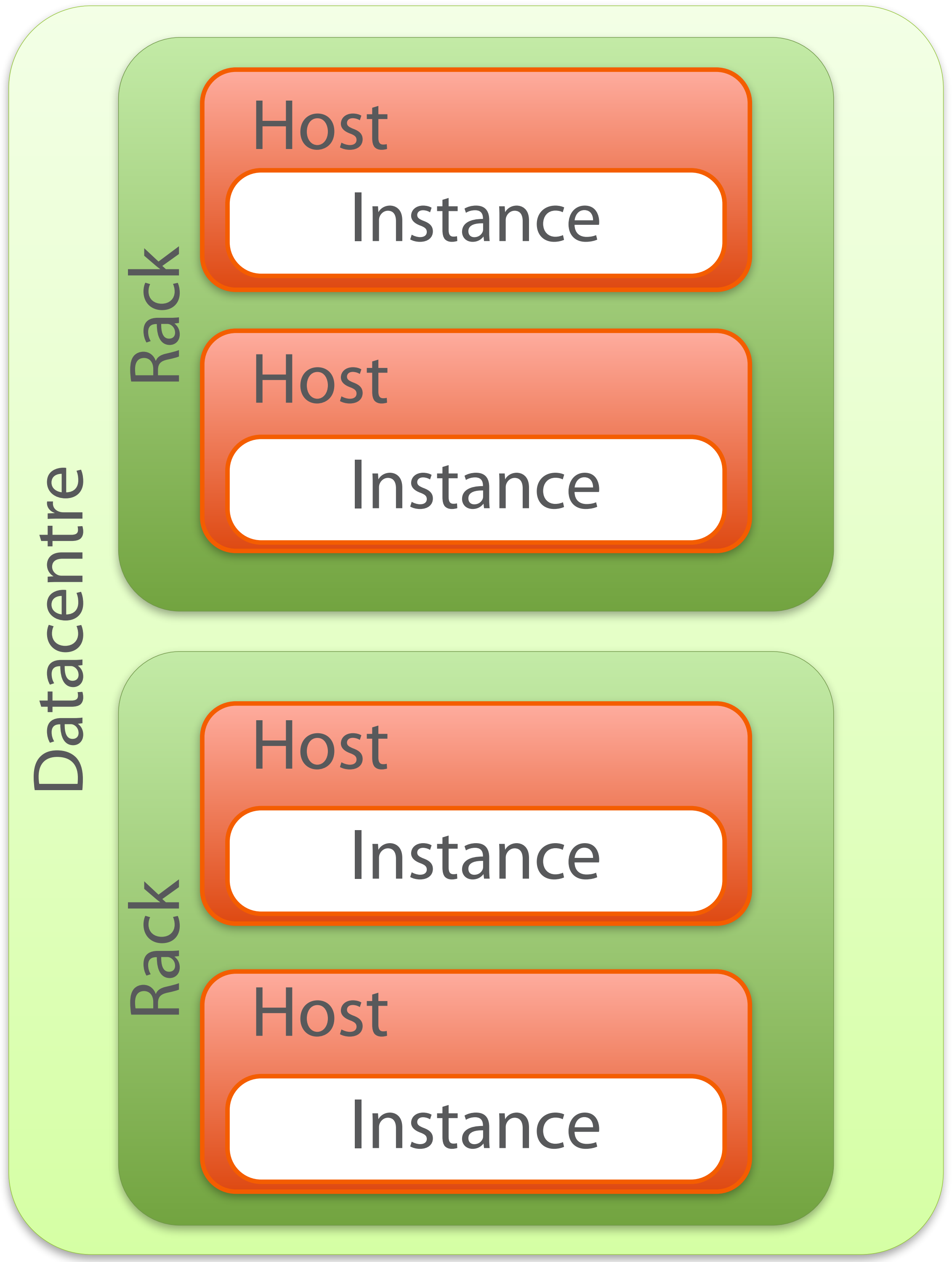
Activation of Backup Data



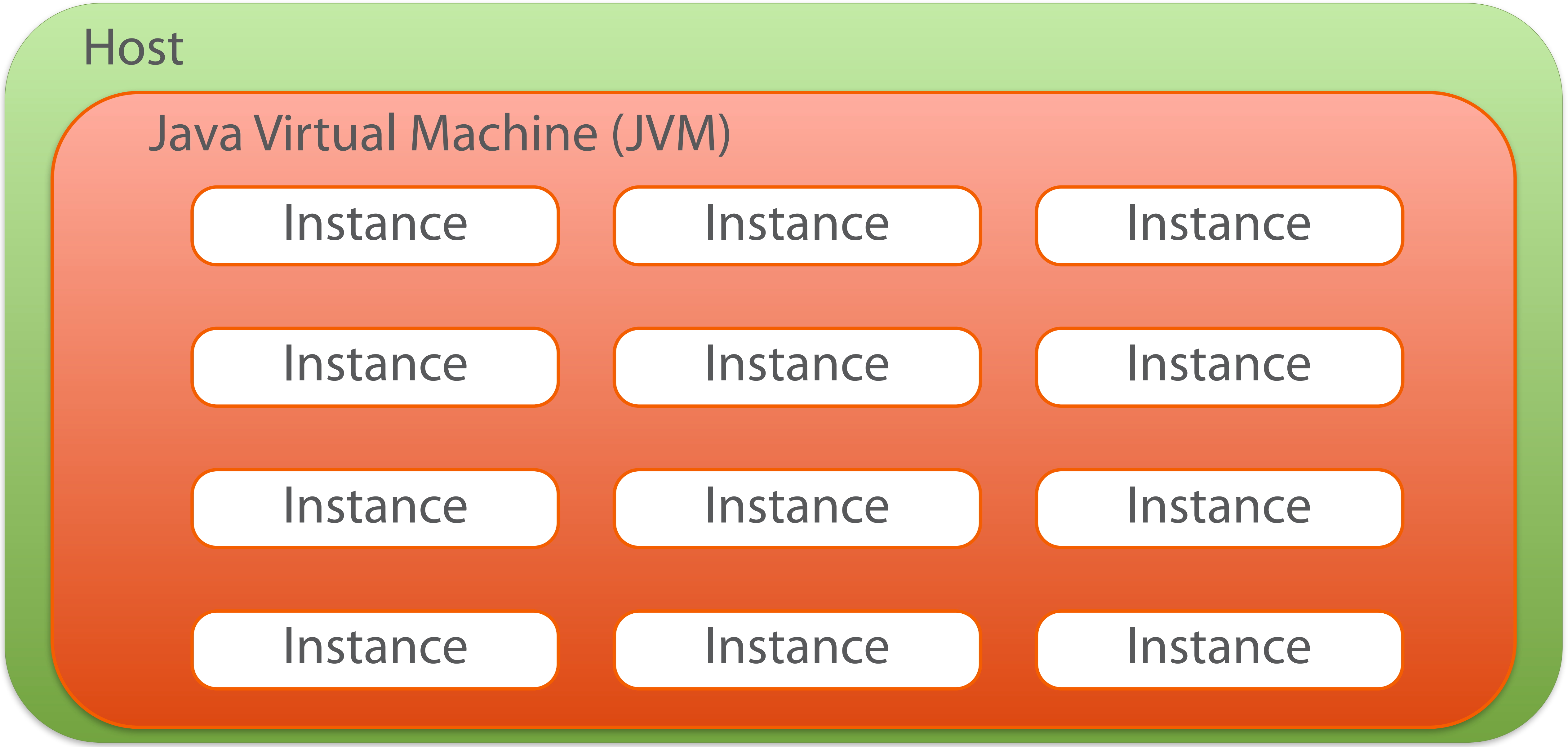
Possible Production Setup



1 Hazelcast Instance
=
1 Storage Node
=
1 Cluster Member



Possible Development Setup



1 Hazelcast Instance = 1 Storage Node = 1 Cluster Member

Map Store Persistence

- Maps store all data in memory
- When all storage nodes are shutdown the data is lost
- Use Map Stores for more persistent storage
  - Create a class that implements com.hazelcast.core.MapStore
  - Configure Hazelcast to use the MapStore for a particular map

Synchronous vs Asynchronous MapStore

- Synchronous
  - Every Map update results in the store or delete method on the MapStore being called
- Asynchronous
  - Every few seconds the Map Store is passed a batch of updates via storeAll or deleteAll methods on the MapStore.
  - Possible risk of data not being saved by the MapStore if the storage node is lost

Searching for Data

- Customer object has a “dob” (date of birth) attribute

```
private Date dob;

public Date getDob() { return dob; }
public void setDob(Date dob) { this.dob = dob; }
```

- Search for Customers who date of birth falls within a range

Main Predicates Available

- and
- between
- equal
- greaterEqual
- greaterThan
- ilike
- in
- lessEqual
- lessThan
- like
- not
- notEqual
- or
- regex

Or implement your own by implementing com.hazelcast.query.Predicate

SQL Predicates

Finding Data Using a SQL Like Language

SQL Predicates

- Possible to write queries in a SQL like language
  - name = ‘Grant’
  - name != ‘Grant’
  - email NOT LIKE ‘%@pluralsight.com’
  - transactionAmount >= 10.00 and transactionAmount < 20.00
  - country IN (‘US’, ‘AU’, ‘IRE’, ‘NZ’, ‘UK’)
  - currency NOT IN (‘AUD’, ‘EUR’, ‘GBP’, ‘USD’)
  - transactionAmount BETWEEN (10.00, 20.00)

Indexing

- Hazelcast stores map objects in their serialised form
- When searching:-
  - Each object is deserialised
  - The predicate is applied to the deserialised data
- Avoid the deserialisation by indexing

Review

- Distributed Map to store objects
- Insert, update and delete from a Map
- Hazelcast distributes data among the cluster members
- MapStore for persistent storage of map data
- Searching using Predicates and the SqlPredicate
- Indexes to improve search performance

Next "Working with Map Data"