Operations, consistency and failover
for multi DC clusters

THE LAST PICKLE

CASSANDRA SUMMIT - SEPTEMBER 2016

Alexander Dejanovski
@alexanderdeja

Consultant
www.thelastpickle.com

Datastax MVP for Apache Cassandra

About The Last Pickle

We help people deliver and improve Apache Cassandra based solutions.

With staff in 5 countries and over 50 years combined experience in Apache Cassandra.


FINEST MENSWEAR
SPRINGFIELD
UNIQUE AND ORIGINAL
1988
BORN IN EUROPE

DC = DataCenter

Why multi DC ?

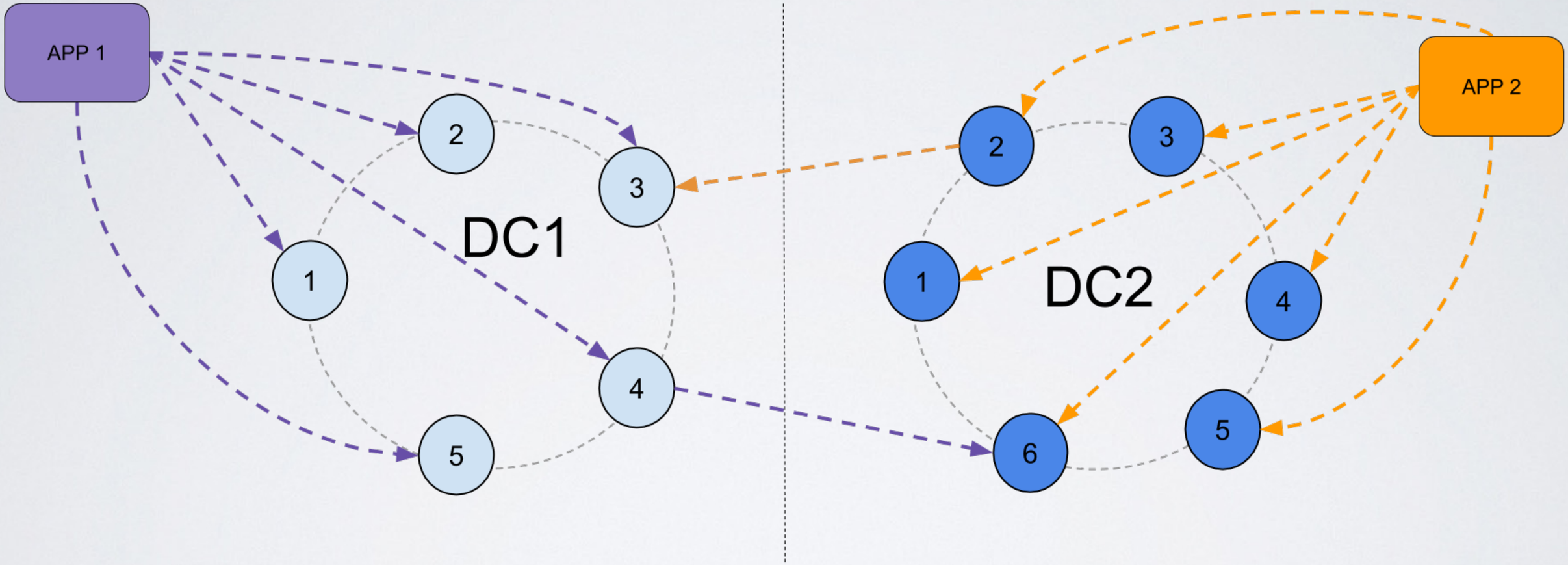
Consistency

Operations

Failover

Load Balancing
&
Disaster recovery

Multi DC use cases



www.thelastpickle.com

Geographical colocation with clients

Multi DC use cases

CloudPing.info

Amazon Web Services™ are available in several regions. Click the button below to estimate the latency from your browser to each AWS™ region.

Region	Latency
US-East (Virginia)	127 ms
US-West (California)	421 ms
US-West (Oregon)	215 ms
Europe (Ireland)	70 ms
Europe (Frankfurt)	80 ms
Asia Pacific (Mumbai)	181 ms
Asia Pacific (Seoul)	348 ms
Asia Pacific (Singapore)	245 ms
Asia Pacific (Sydney)	351 ms
Asia Pacific (Tokyo)	278 ms
South America (São Paulo)	282 ms
China (Beijing)	221 ms

HTTP Ping

If you like this tool, please check out RestBackup.com. We help software makers to add subscription features, reduce support costs, and differentiate their products from the competition.

Mike Leonhard
mike@restbackup.com

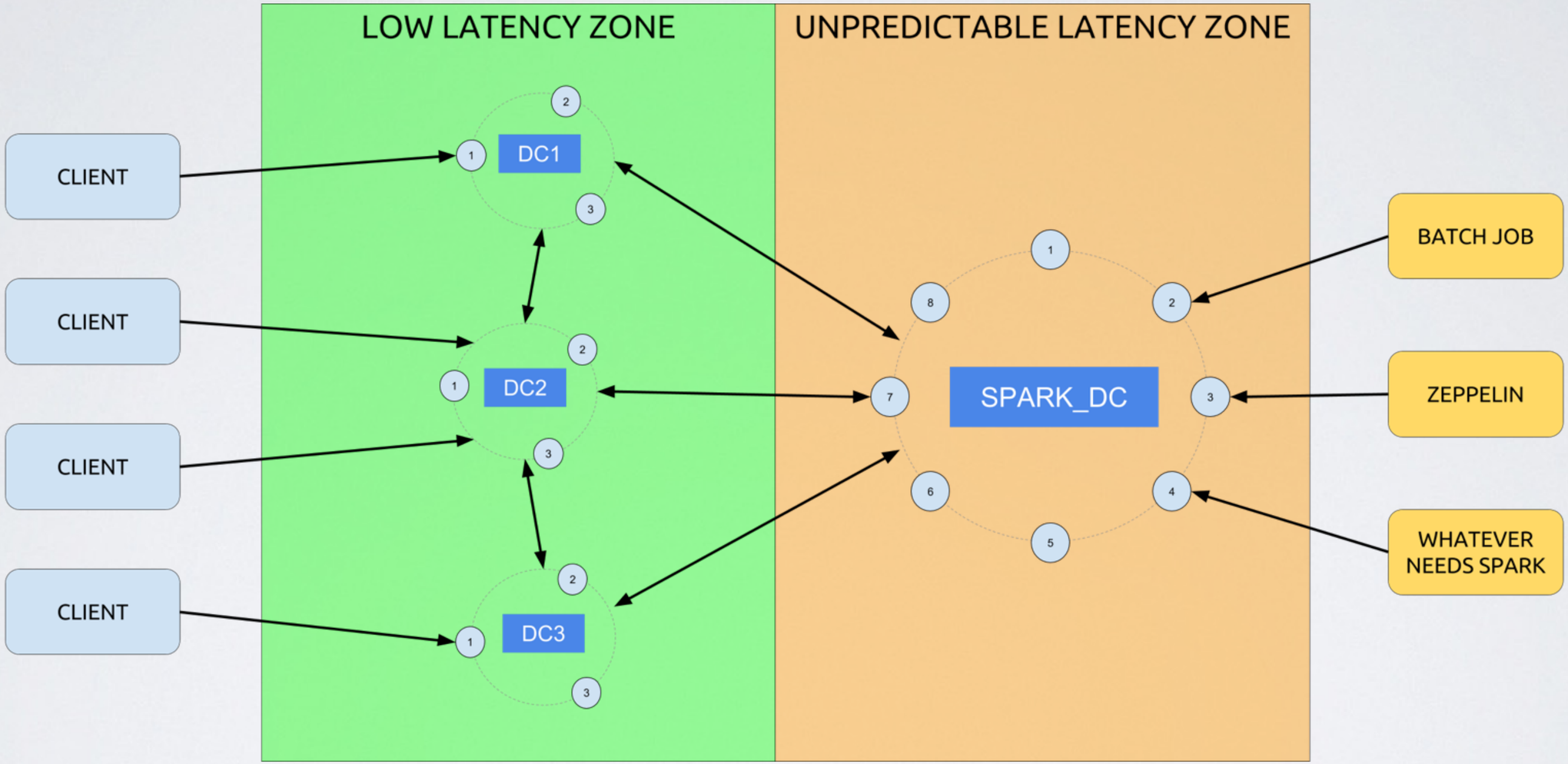
Multi DC use cases

www.thelastpickle.com

Multi DC use cases

Separate operational and analytical workloads

Multi DC use cases



Why multi DC ?

Consistency

Operations

Failover

Strongly consistent clusters

Low latency between DCs
and At least 3 DCs
and No search/analytical DC

Eventually consistent clusters

High latency between DCs

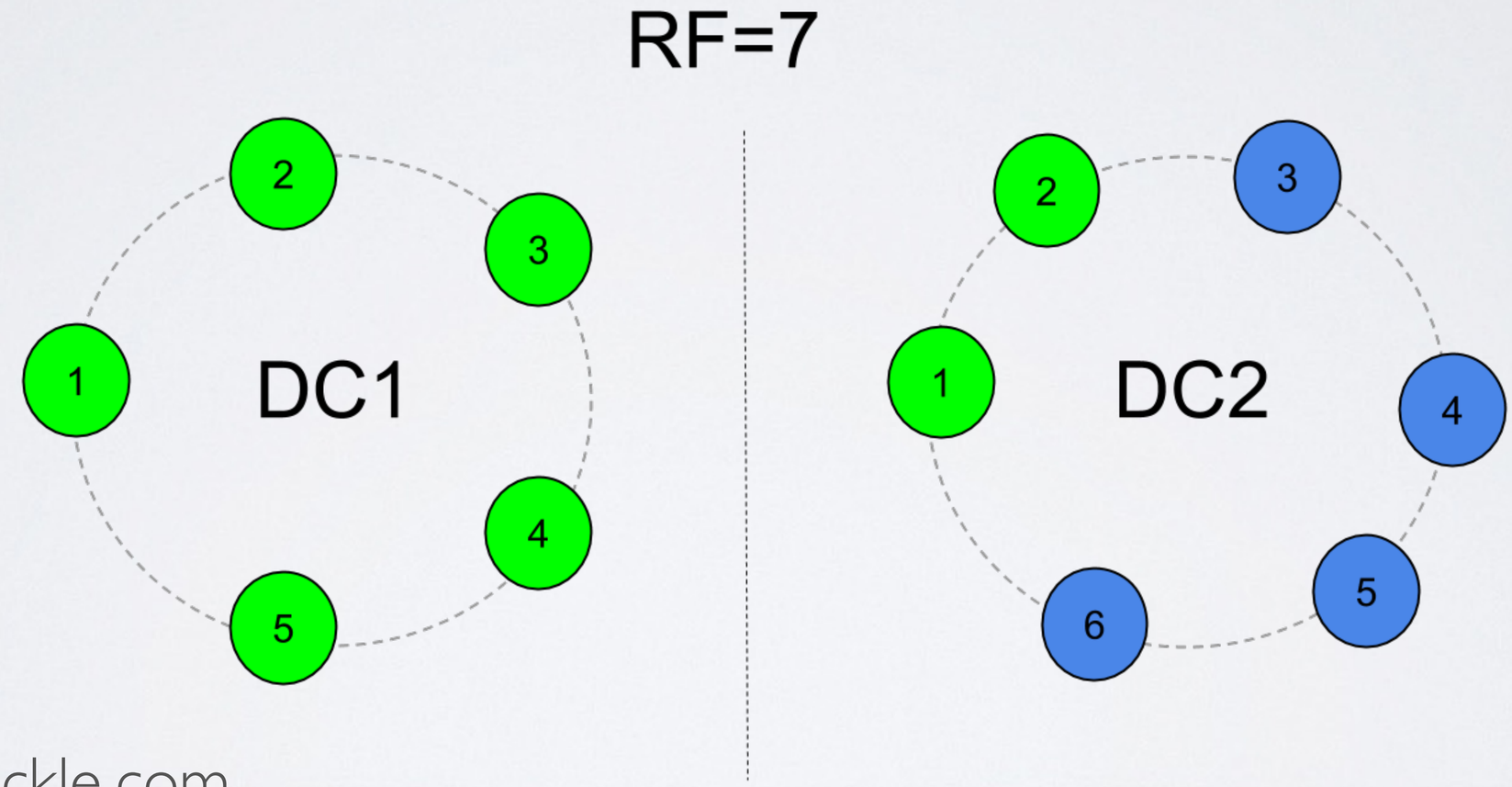
or **exactly 2 DCs**

or **at least 1 search/analytical DC**

Create a keyspace for a single DC cluster :

```
CREATE KEYSPACE ks1
WITH replication =
        {
         'class':'SimpleStrategy',
         'replication_factor' : 7
        };
```

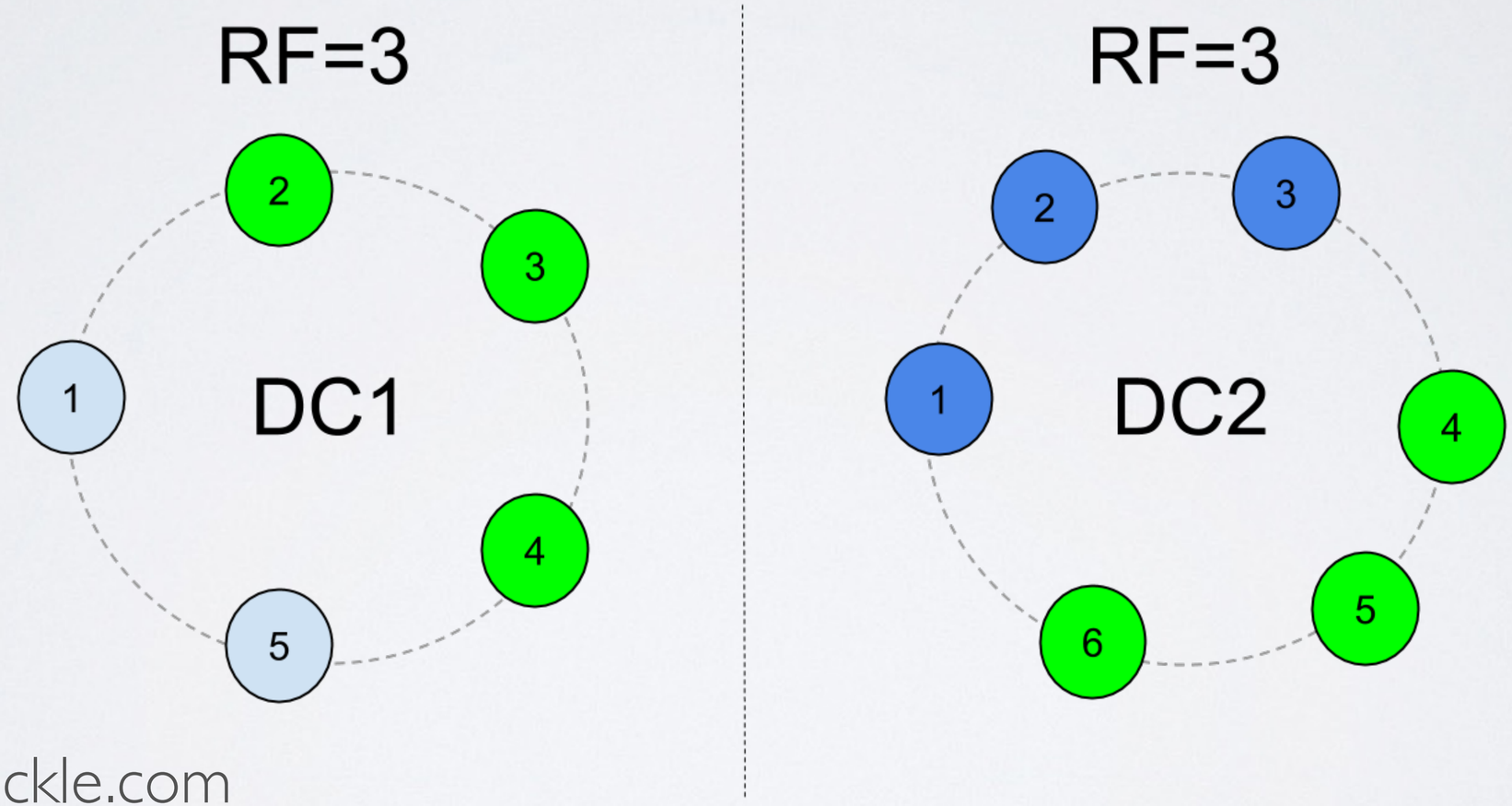
SimpleStrategy on a multi DC cluster



Create a keyspace for a multi DC cluster :

```
CREATE KEYSPACE ks1
WITH replication =
        {'class':'NetworkTopologyStrategy',
         'dc1' : 3,
         'dc2' : 3
        };
```

NetworkTopologyStrategy on a multi DC cluster



Configuring DC on nodes

With **GossipingPropertyFileSnitch** :

conf/cassandra-rackdc.properties

dc=DC2

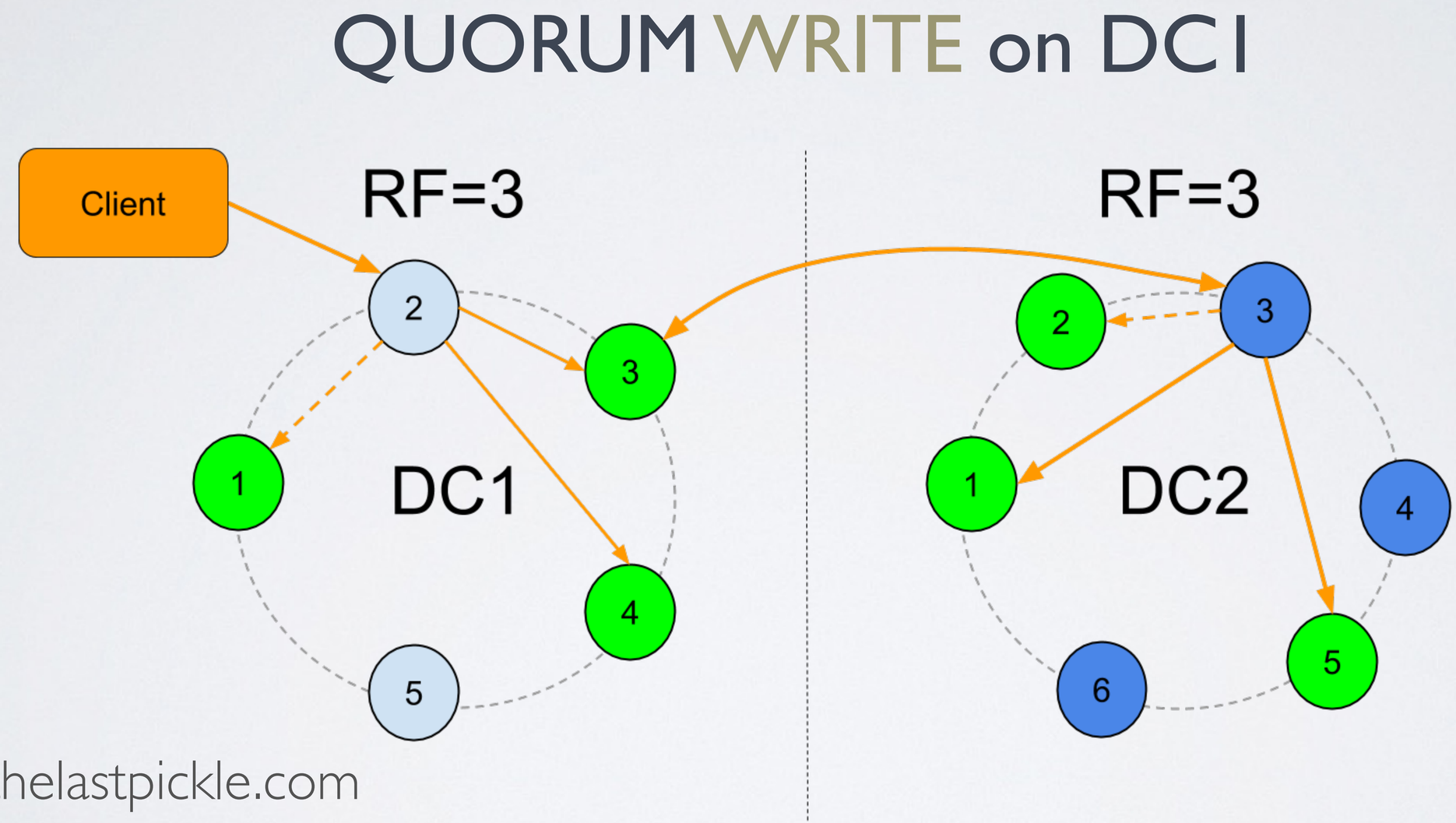
Non DC-aware Consistency Levels

ONE (default)
(TWO, THREE)
QUORUM
ALL

DC-aware Consistency Levels

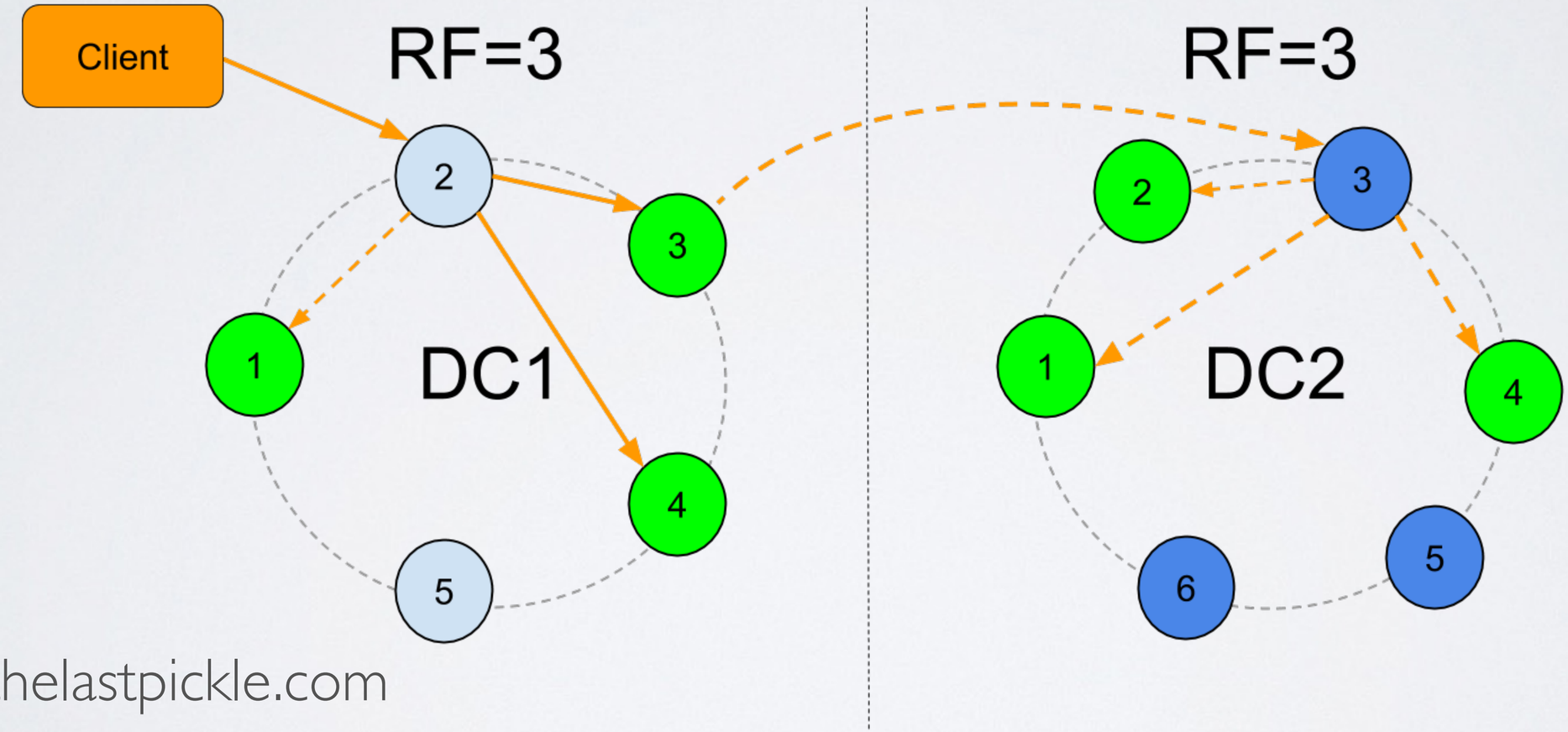
LOCAL_ONE (default)
LOCAL_QUORUM
EACH_QUORUM

Read & write path

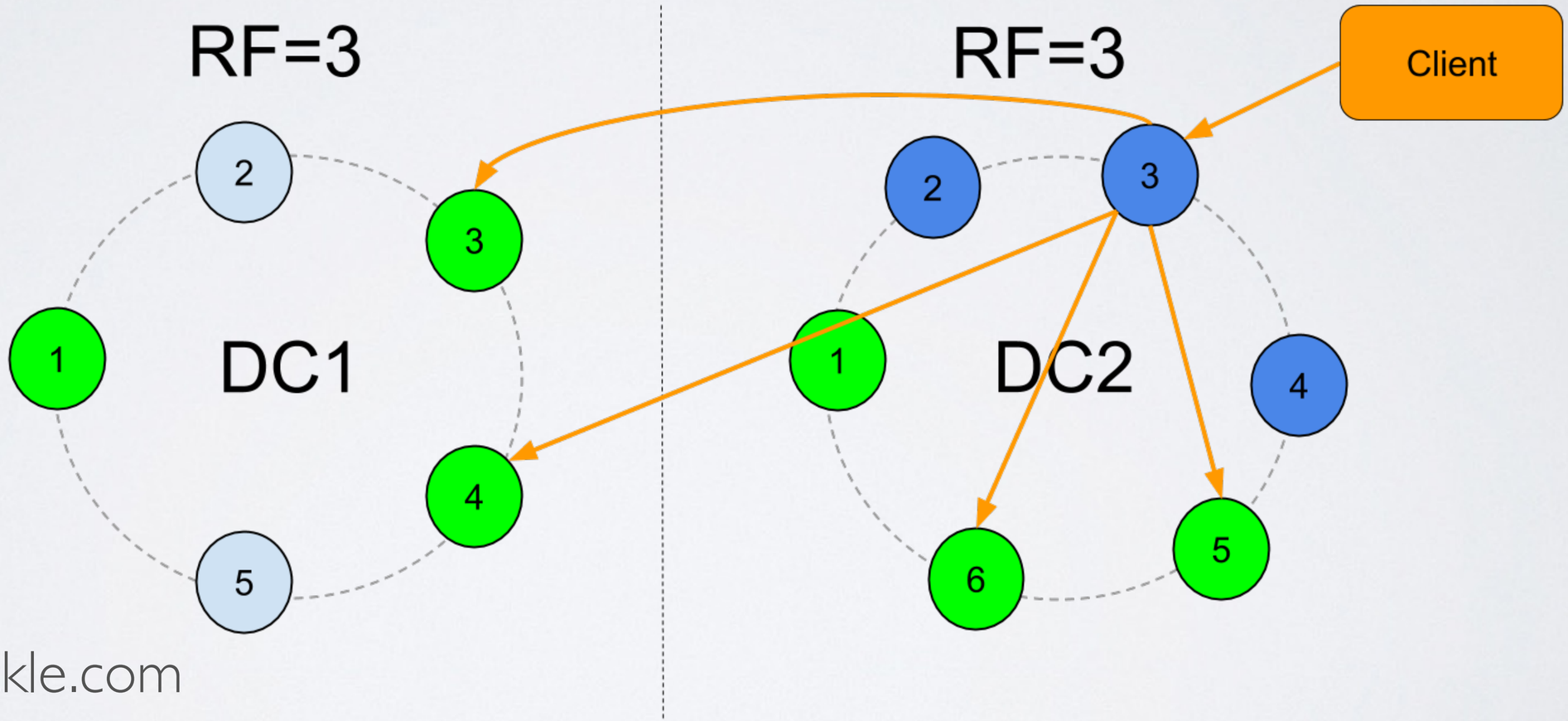


Read & write path

LOCAL_QUORUM WRITE on DC1

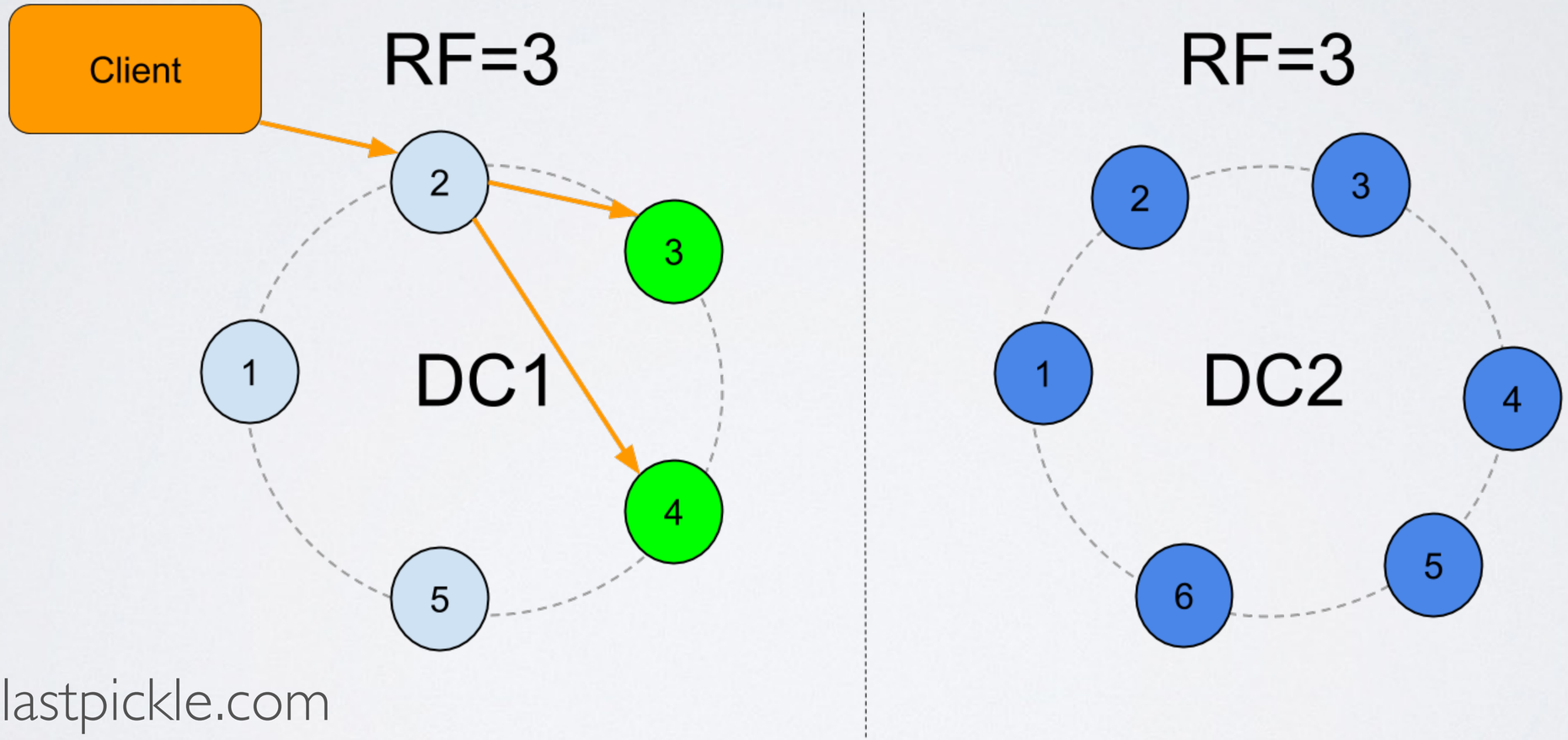


Read & write path



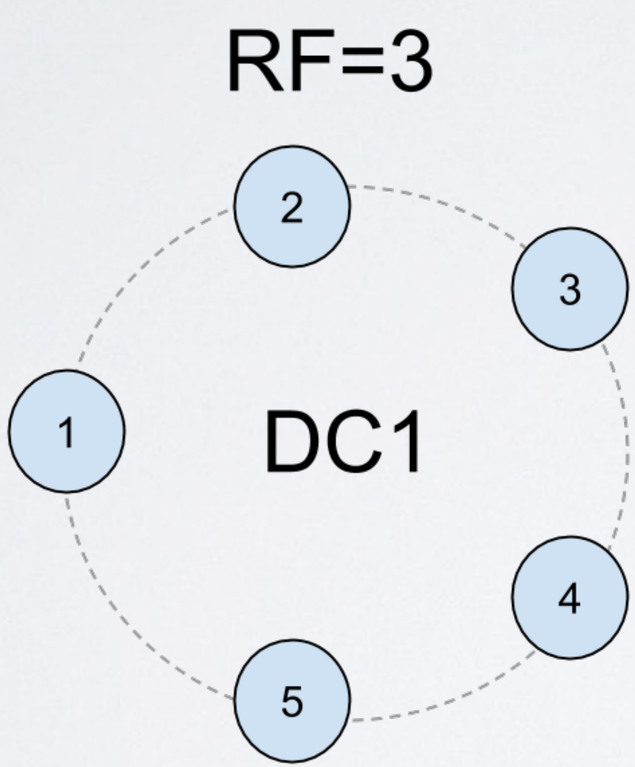
Read & write path

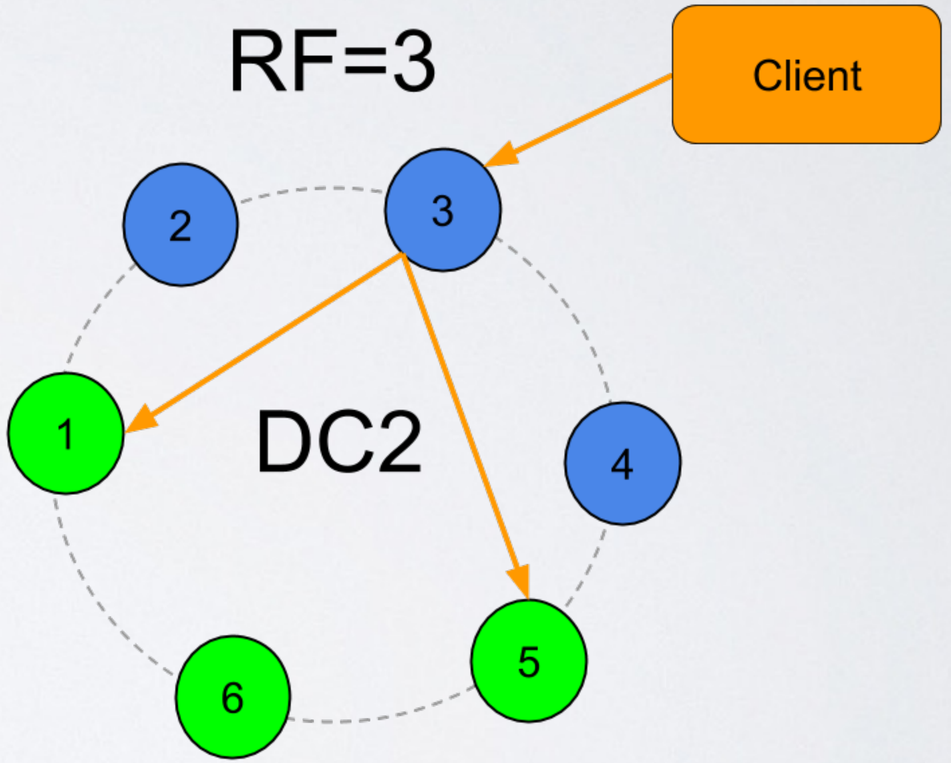
LOCAL_QUORUM READ on DC1



Read & write path

LOCAL_QUORUM READ on DC2





Why multi DC ?

Consistency

Operations

Failover

Specific configurations for multi DC clusters in conf/cassandra.yaml

Specific throttling for inter DC streaming throughput :

```
inter_dc_stream_throughput_outbound_megabits_per_sec
```

Defaults to 200 Mbps (25 MB/s)

Internode encryption can be activated for inter DC communications only

```
server_encryption_options:
    internode_encryption: dc
```

Internode compression can be activated for inter DC communications only

```
internode_compression: dc
```

Operations & configuration

Reduce the TCP overhead in async DCs by setting :

```
inter_dc_tcp_nodelay: true
```

Larger but fewer TCP packets

Adding a new DC to an existing cluster

Migrate all your SimpleStrategy KS to NetworkTopologyStrategy

Operations - adding a new DC

```
ALTER KEYSPACE ks1
WITH replication =
        {'class':'NetworkTopologyStrategy',
         'dc1' : 3
        };
```

Operations - adding a new DC

Disable auto bootstrap on new nodes

Operations - adding a new DC

Disable auto bootstrap on new nodes
(not mandatory, but safer…)

Operations - adding a new DC

Add this in conf/cassandra.yaml :

```
auto_bootstrap: false
```

Start new nodes

At this point, Nodes in the new DC are :

- empty
- not involved in reads nor writes

Change strategy params to add replicas on the new DC

You might want to make sure traffic is restricted to DC1 before you move on…

(unless you're using QUORUM)

Operations - adding a new DC

```
ALTER KEYSPACE ks1
WITH replication =
        {'class':'NetworkTopologyStrategy',
         'dc1' : 3,
         'dc2' : 3
        };
```

At this point, your new DC accepts both reads and writes

But nodes on the new DC
are still desperately empty

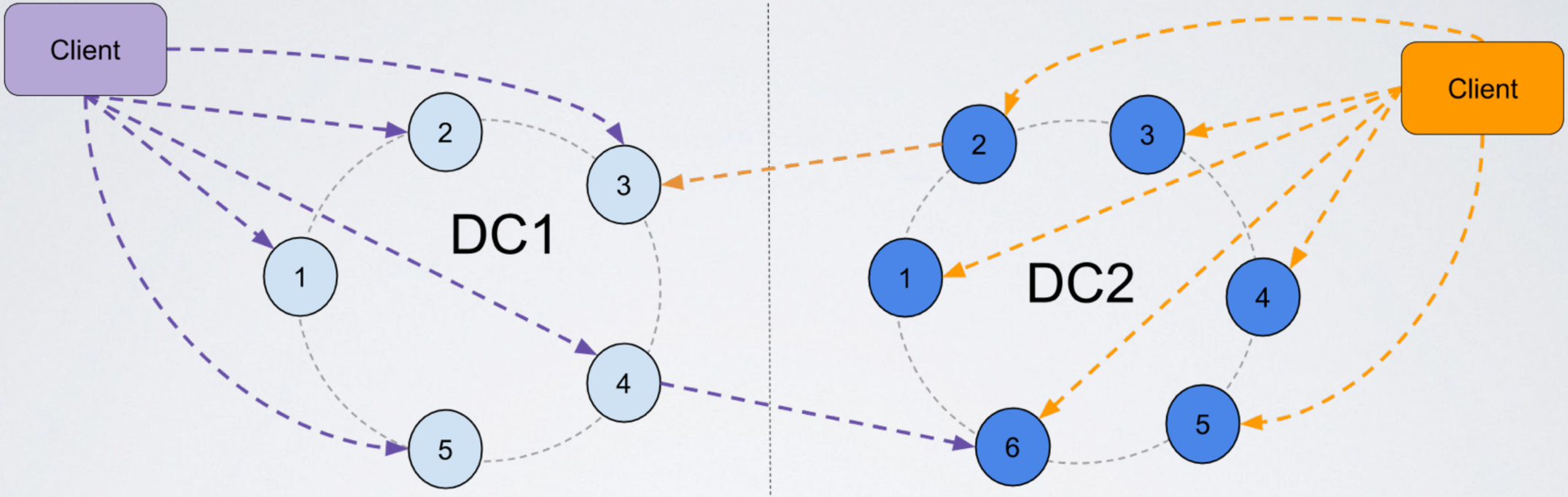
Routing traffic on a specific DC
is a dev task

Operations - adding a new DC

1/ Pick a coordinator in a specific DC :

use DCAwareRoundRobinPolicy in your <paste your language> Datastax driver

Operations - adding a new DC




```
Cluster.builder().withLoadBalancingPolicy(
    new DCAwareRoundRobinPolicy("DC1")
);
```

```
Cluster.builder().withLoadBalancingPolicy(
    new DCAwareRoundRobinPolicy("DC2")
);
```

2/ tell the coordinator to work with nodes in its own DC only :

use a LOCAL_* CL

Operations - adding a new DC

Consistency level can be modified on the fly through feature flips for example

Load balancing policy cannot…

Fill your new nodes
with data taken from `dc1` :

Run a rolling « `nodetool rebuild dc1` »
on all nodes in dc2

Operations - adding a new DC

Your new DC is now
fully ready to rock

How to remove DC2 from the cluster

Switch all traffic to DC1

You may want to run repair

You may want to run repair
(hopefully you've seen my talk yesterday)

Operations - removing a DC

```
ALTER KEYSPACE ks1
WITH replication =
        {'class':'NetworkTopologyStrategy',
         'dc1' : 3,
         'dc2' : 3
        };
```

Decommission all nodes from dc2

Run « `nodetool decommission` »
on all nodes in dc2

Anti-entropy repair

Merkle trees are requested
from all replicas in all DCs by default

Specific switch to run repair
in the local DC:

```
nodetool repair -local
```

Should you run repair on all DCs ?

Hints & repair

Yes, if :

SimpleStrategy KS

or

-local switch

or

KS not replicated to all DCs

Otherwise no

Try to avoid « over-repairing »
your cluster

Each token range needs a single pass…

Hints work between DCs like they do between nodes in a single DC

Hints can be disabled on specific DCs in conf/cassandra.yaml :

```
hinted_handoff_disabled_datacenters:
    - DC1
    - DC2
```

This means DC1 and DC2 won't receive hints

(use this wisely)

Advice for hints in multi DC clusters :

raise `max_hints_delivery_threads` to `4`

Why multi DC ?
Consistency
Operations
Failover

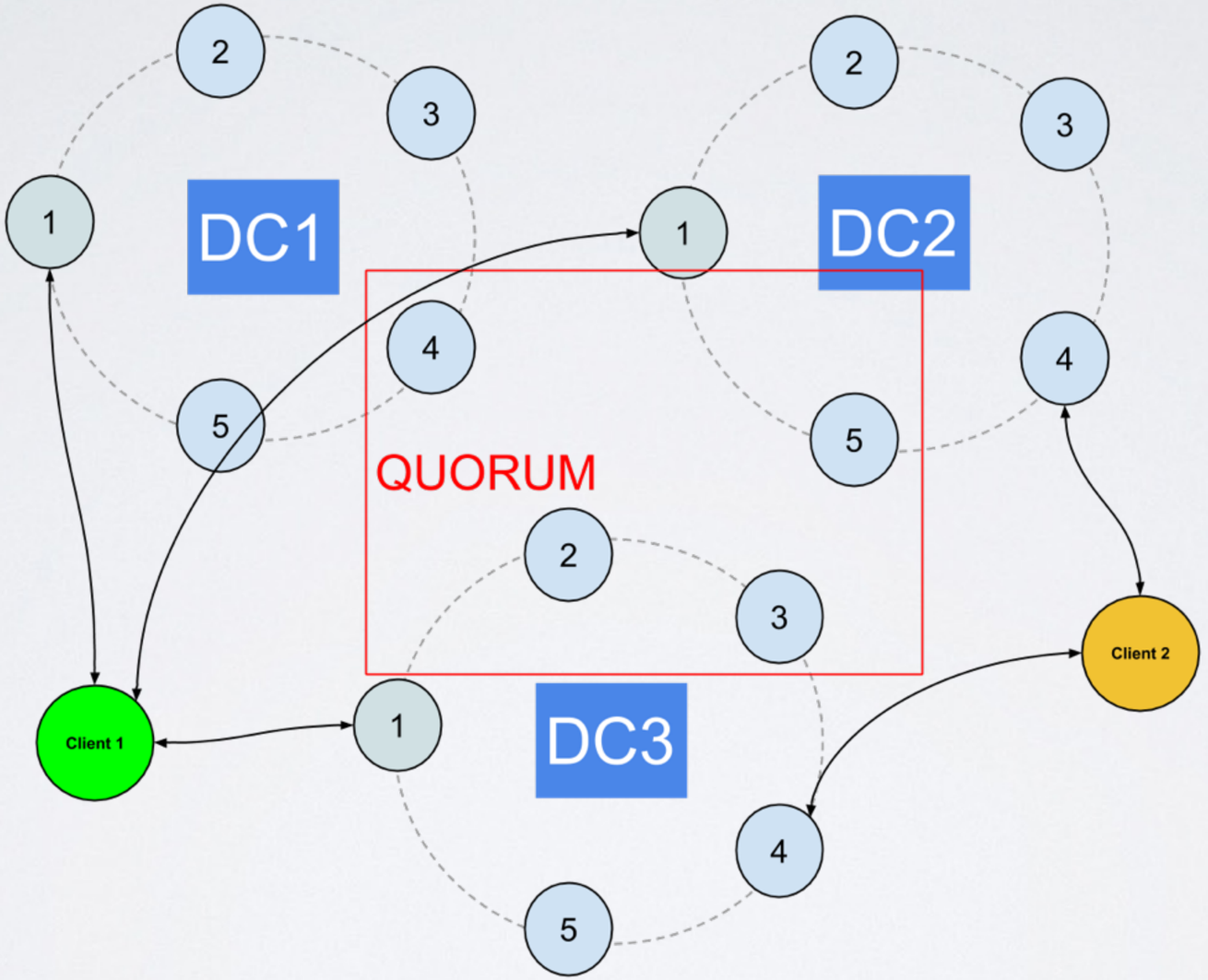
Failover in single DC clusters

Use CL ONE or QUORUM

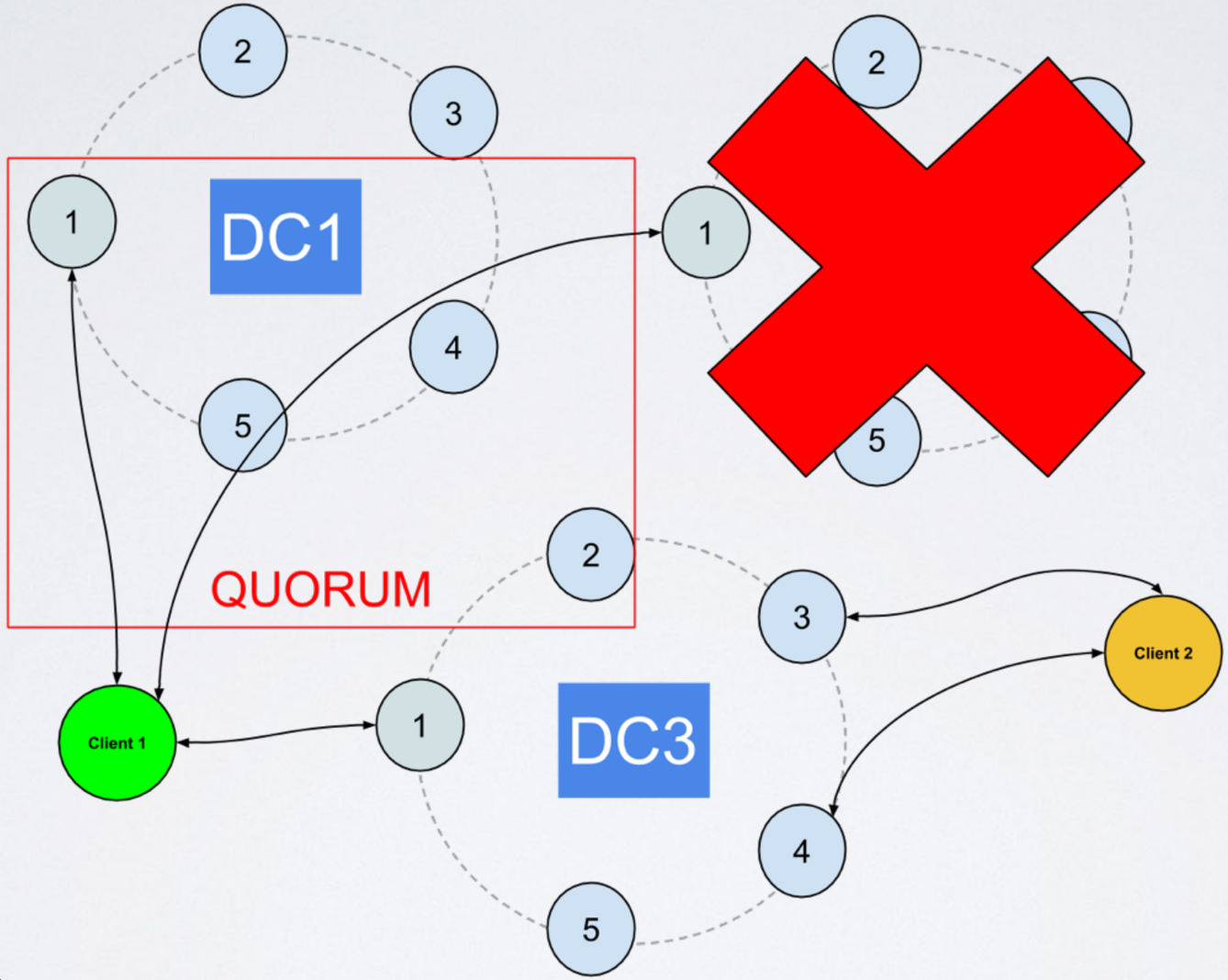
Failover in multi DC strongly consistent clusters

Use CL ONE or QUORUM

Failover



Failover

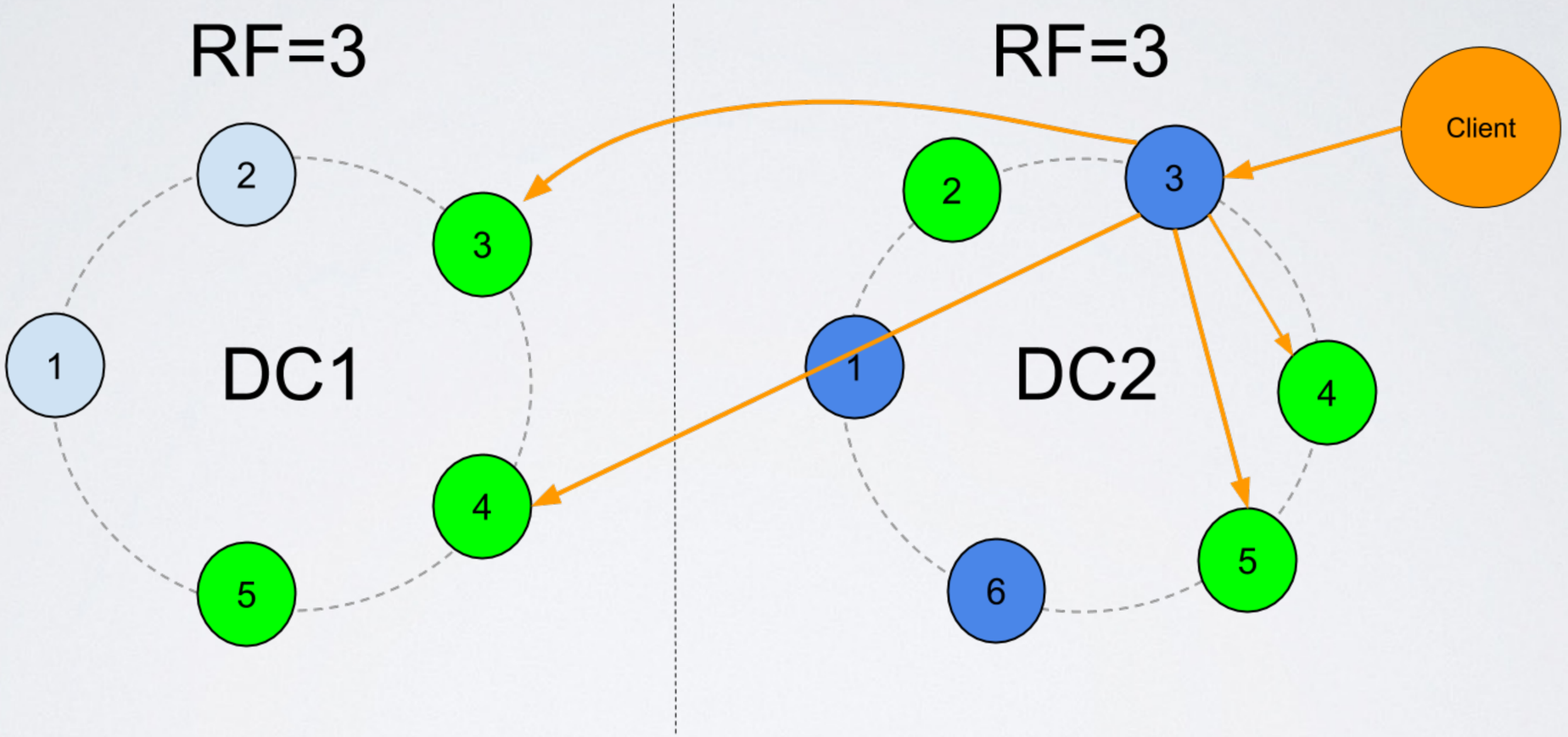


www.thelastpickle.com

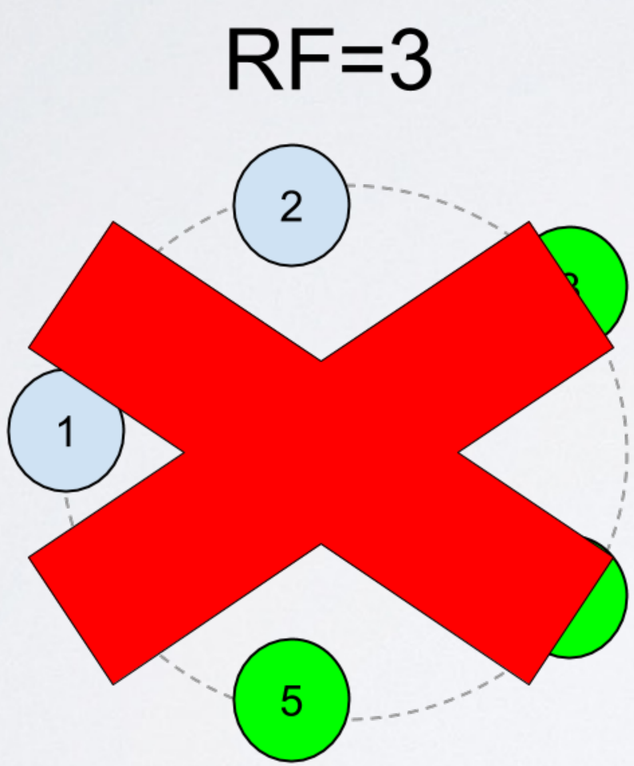
Failover in multi DC eventually consistent clusters

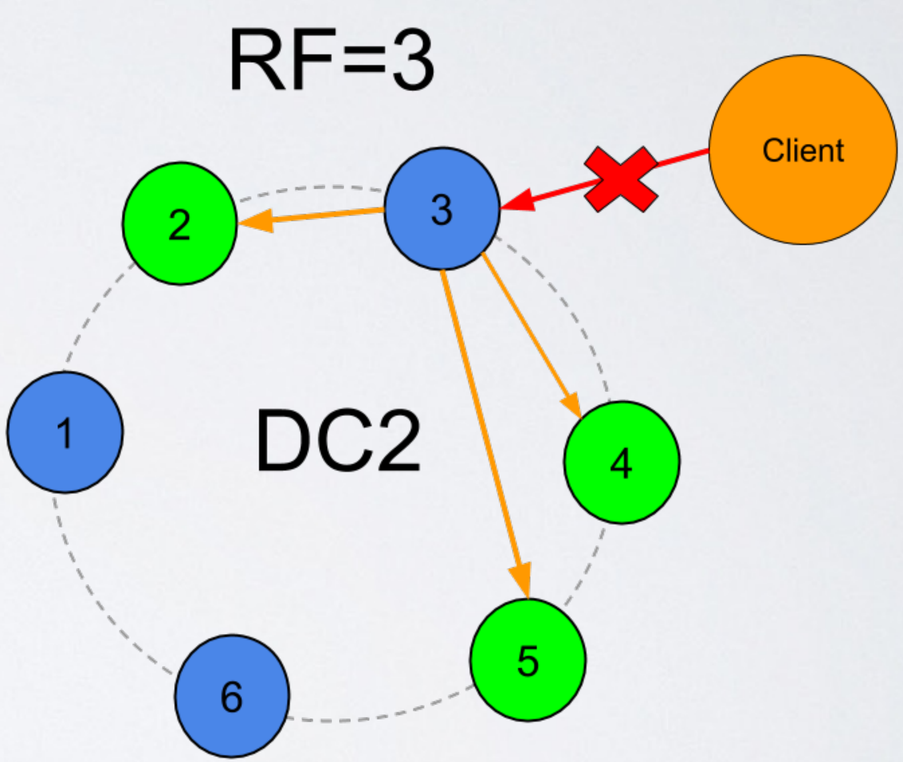
What if my local DC loses QUORUM ?

Failover



Quorum at RF 6 is 4





You have to build inter DC failover

detect failure

metrics, token range monitoring, …

Failover

switch traffic

network, app or driver level

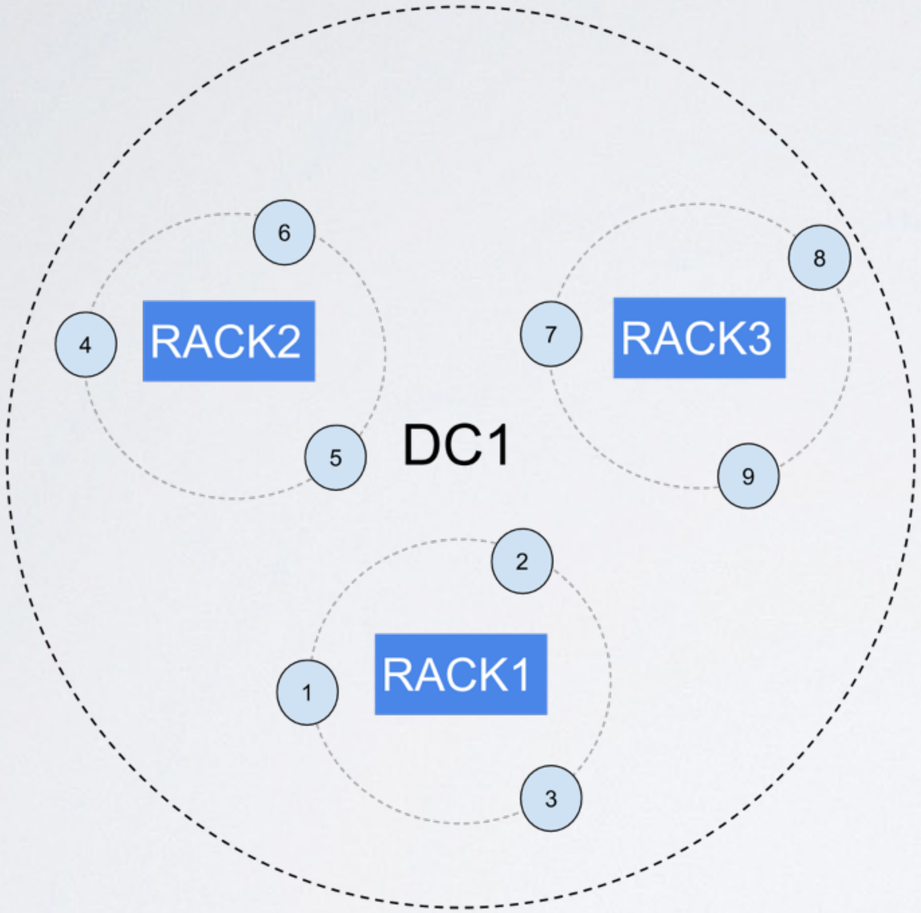
prevent premature back switch
inconsistencies

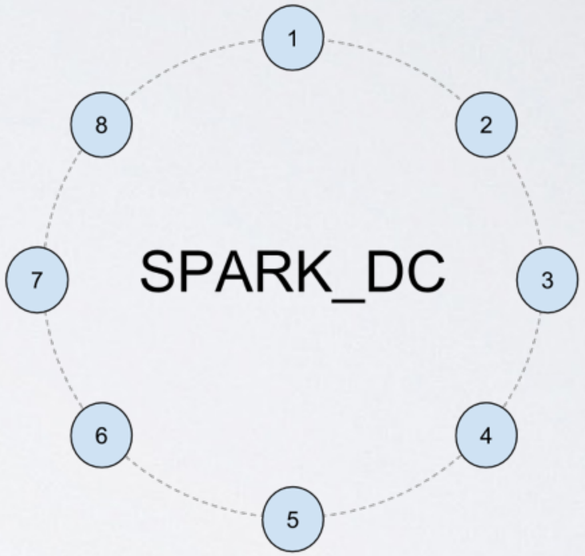
Build your own load balancing policy
on top of the DCAwareRoundRobinPolicy

If you have an analytical DC
and want synchronous operational DCs

Use racks

Failover





Thanks!

@alexanderdeja