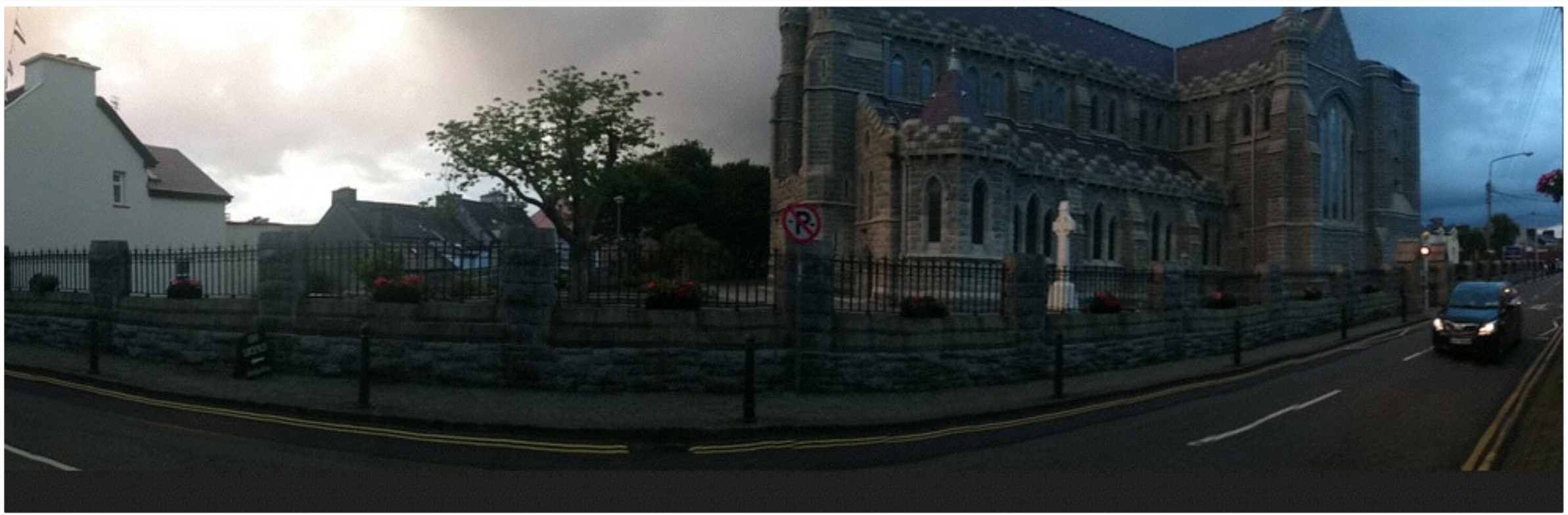


Sometimes (most times) down and out is better than slow.

Adaptive Availability *for* Quality *of* Service

A new world order

Slow ≅ Byzantine

In most modern systems, users perceive:

"slow is the new down."

In most distributed systems:

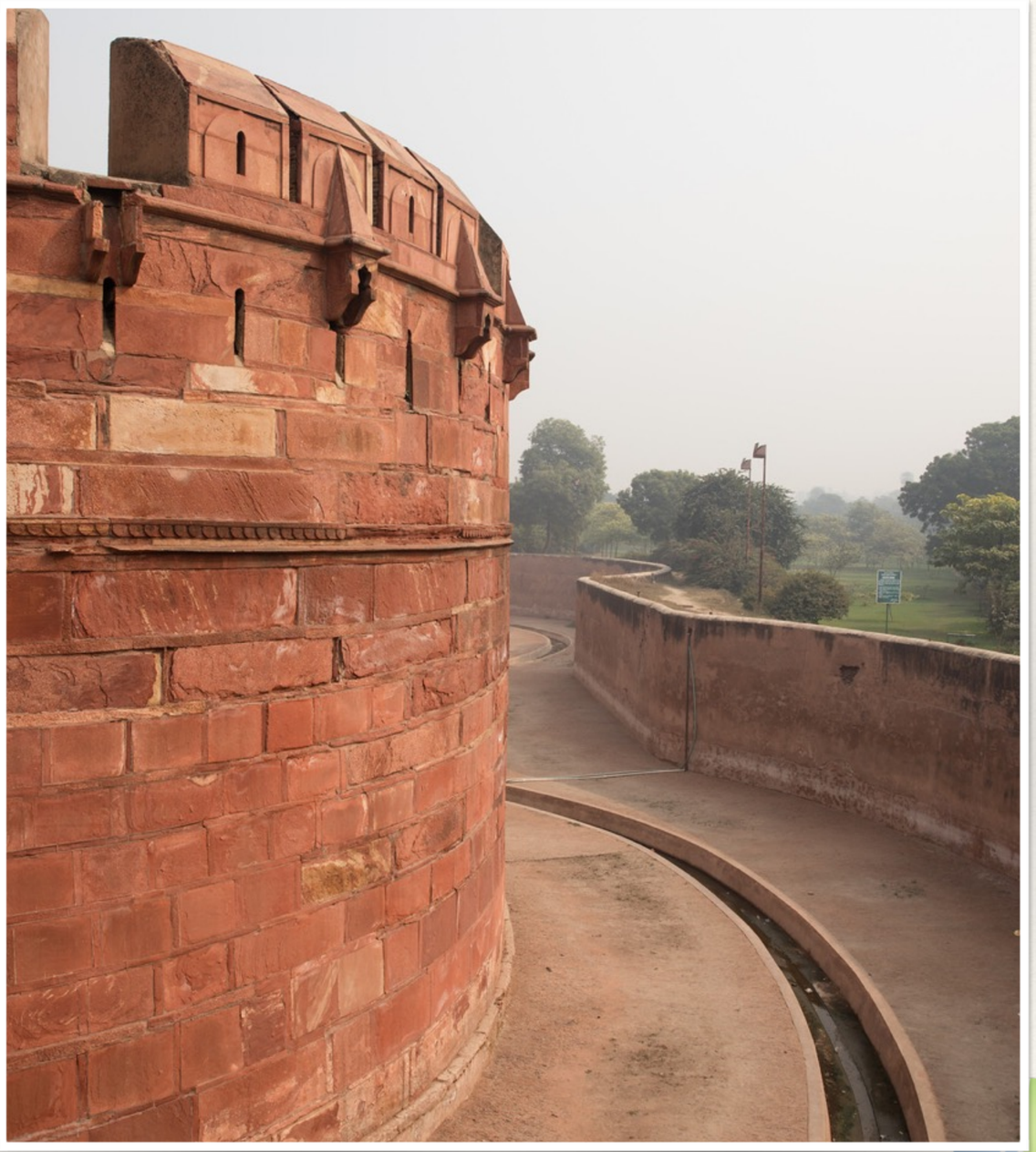
"slow is indistinguishable from byzantine operations."

We had to be "very sure" in the

Days *of* Failover

Primary : Replica system **usually** have non-zero operational costs in performance failover.

- dataloss (in asynchronous systems)
- operational downtime
- operational rebuild time (reversing the flows)



For well-designed, available systems,

Constraints Have Changed

Deciding to fail a node is no longer a

“last resort” decision.

What do I mean by

well-designed?

The failure of a node does not cause

- service interruption
- significant performance regressions

The recovery of a node does not cause

- unnecessary work (only minimal replay)
- significant performance regressions

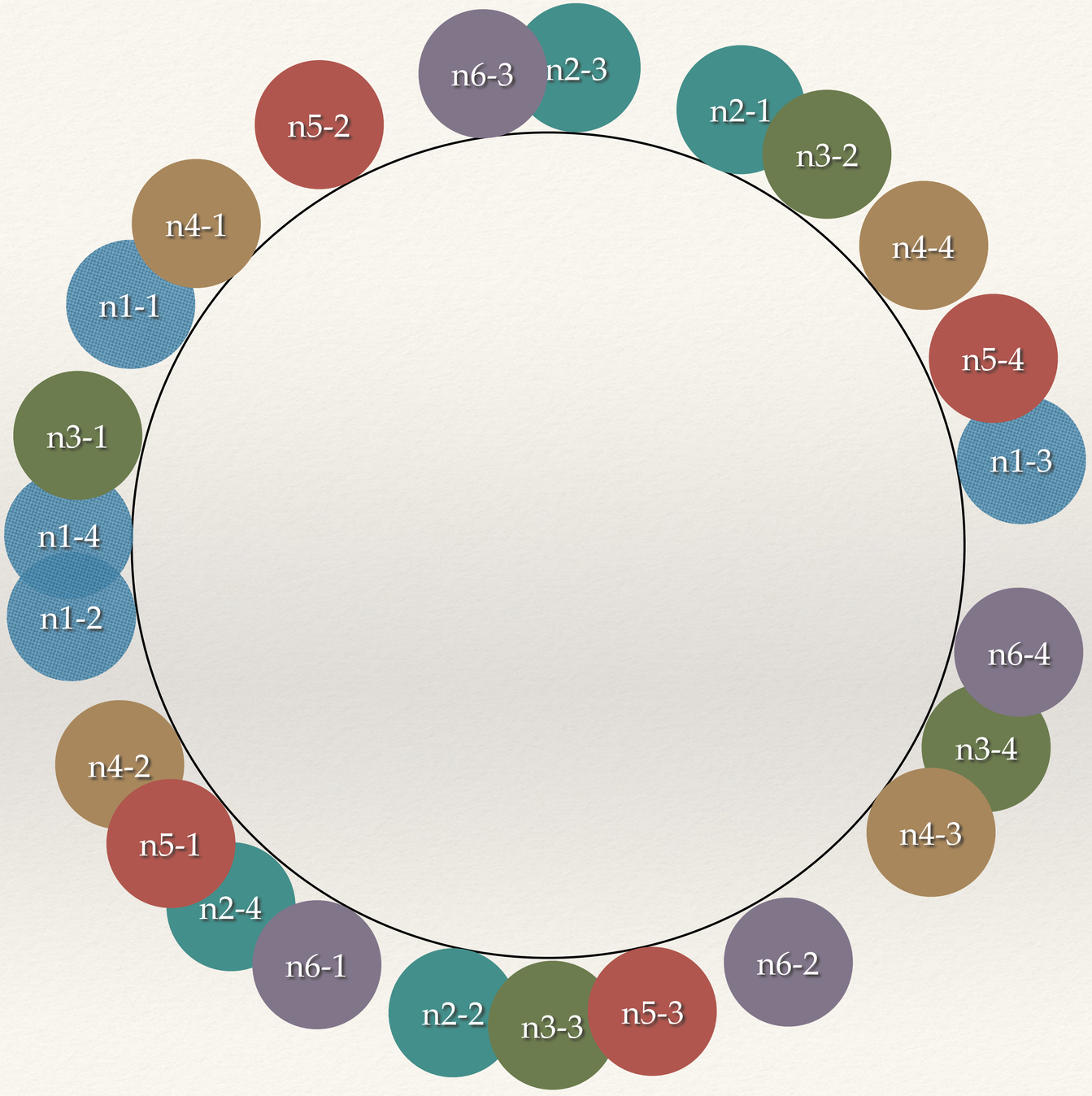
A brief tangent on an

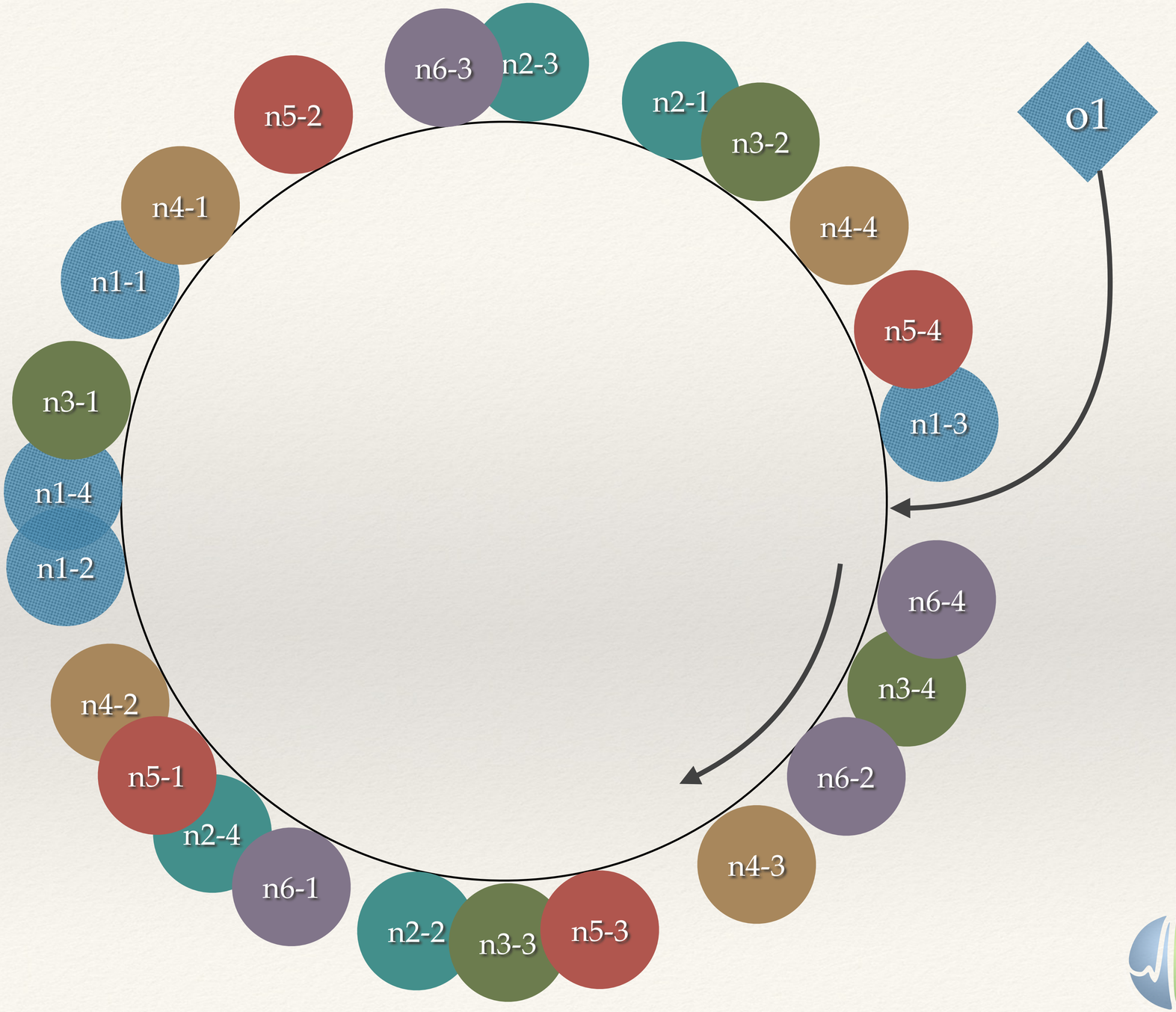
Anecdotal Design

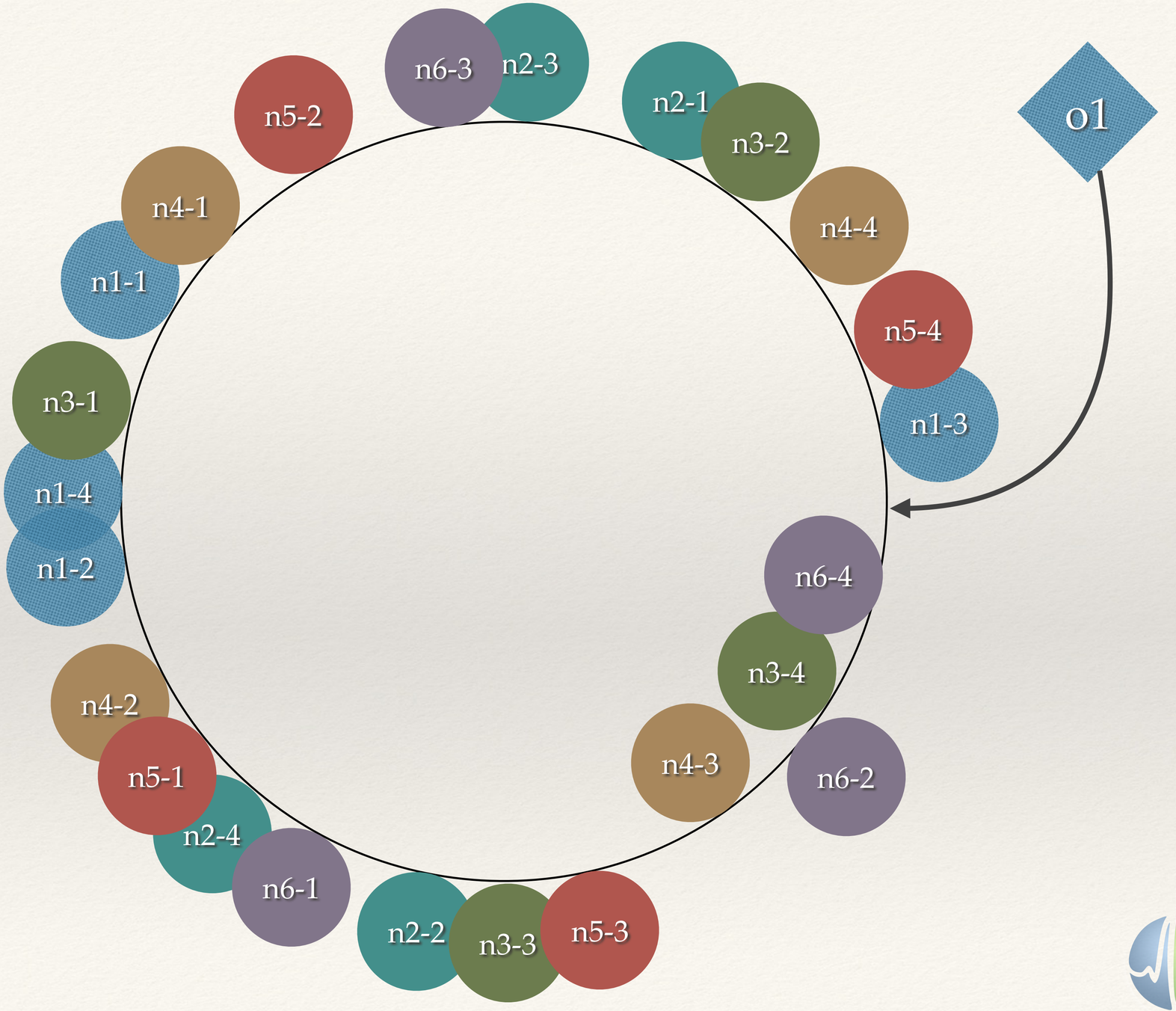
Active feedback on replay performance

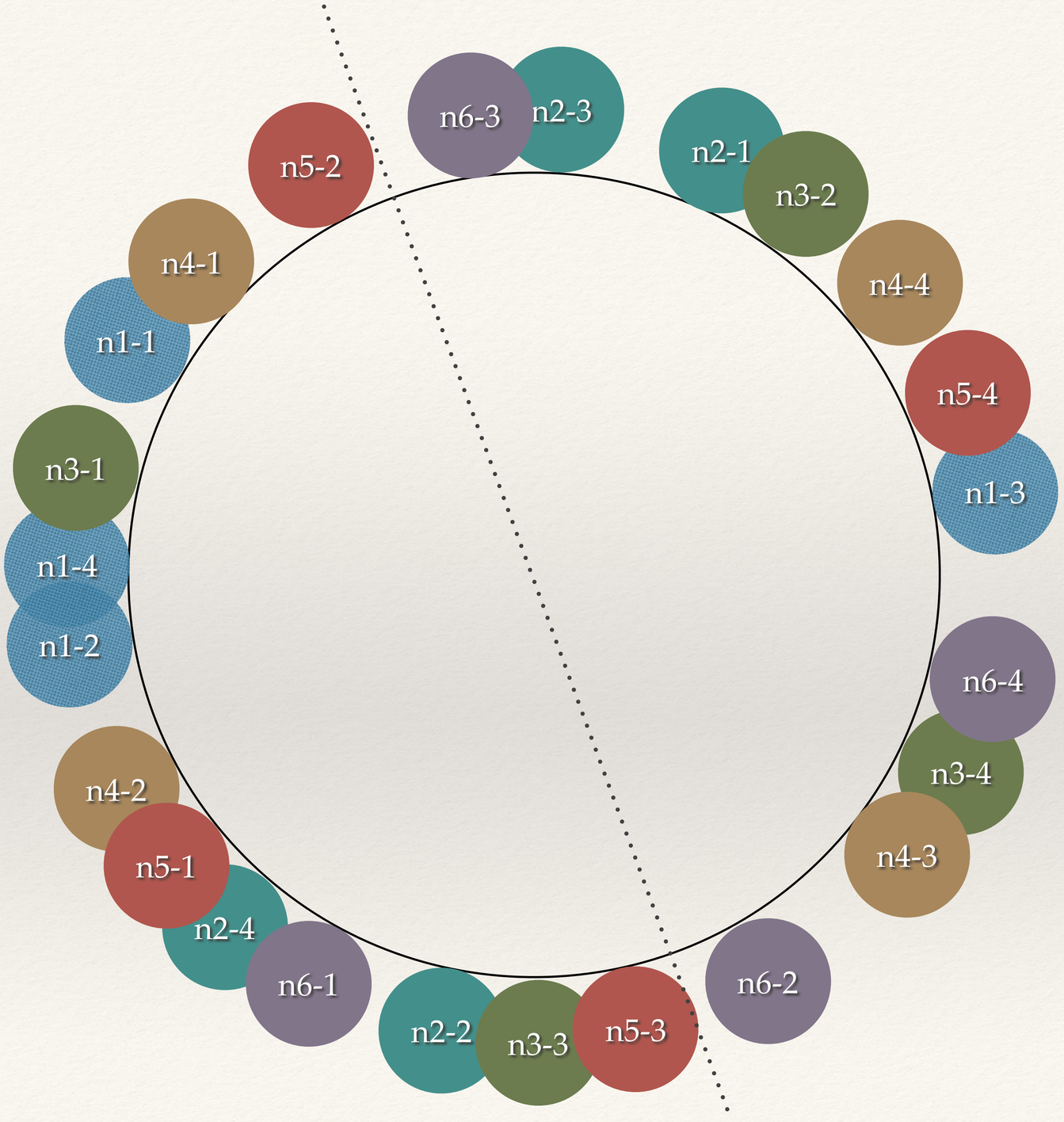
Snowth design

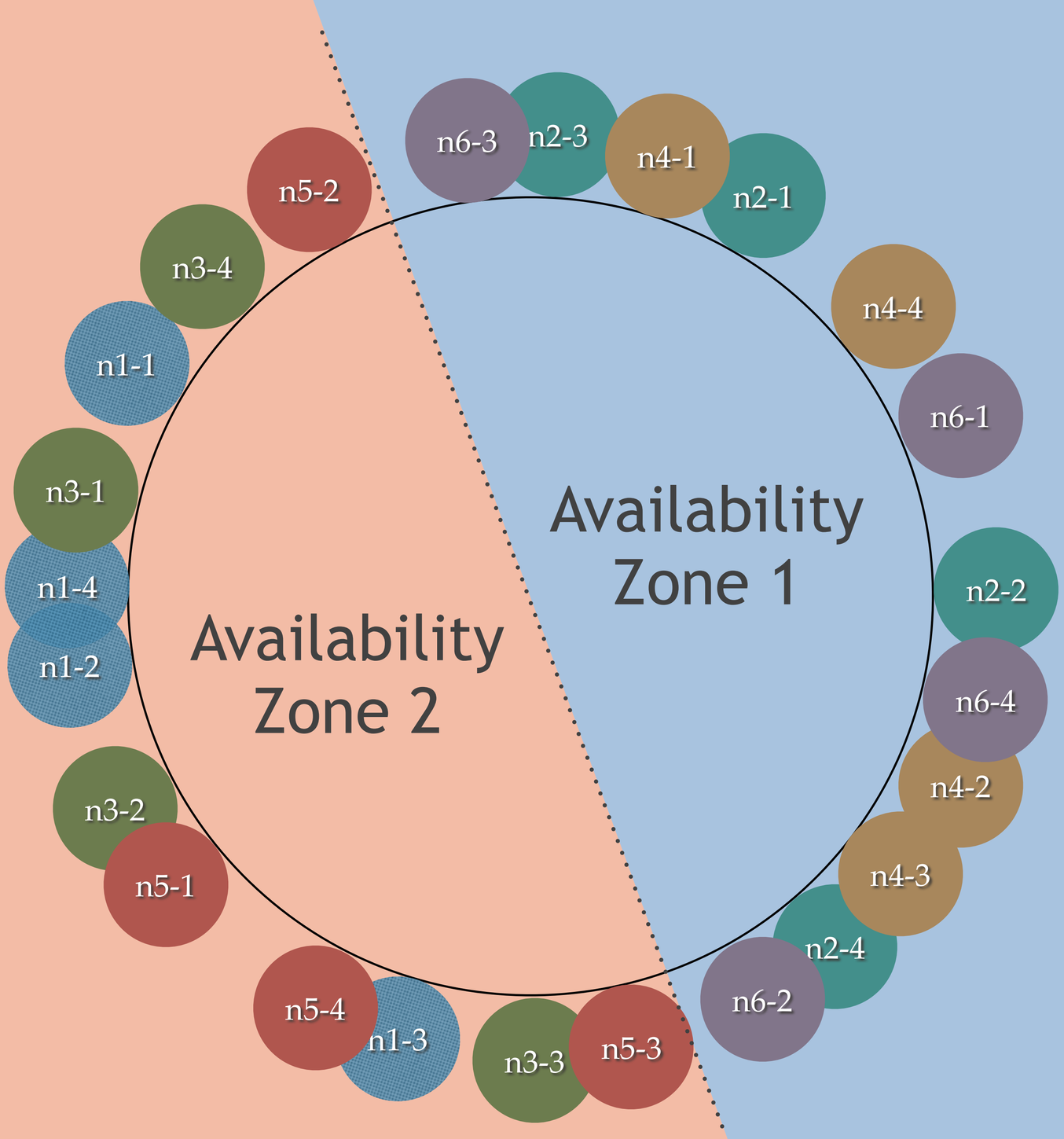
- Need: zero-downtime
- Know: Agreement is hard.
- Know: Consensus is expensive.
- CAP theorem tradeoffs suck.
- CRDT (Commutative Replicated Data Type)

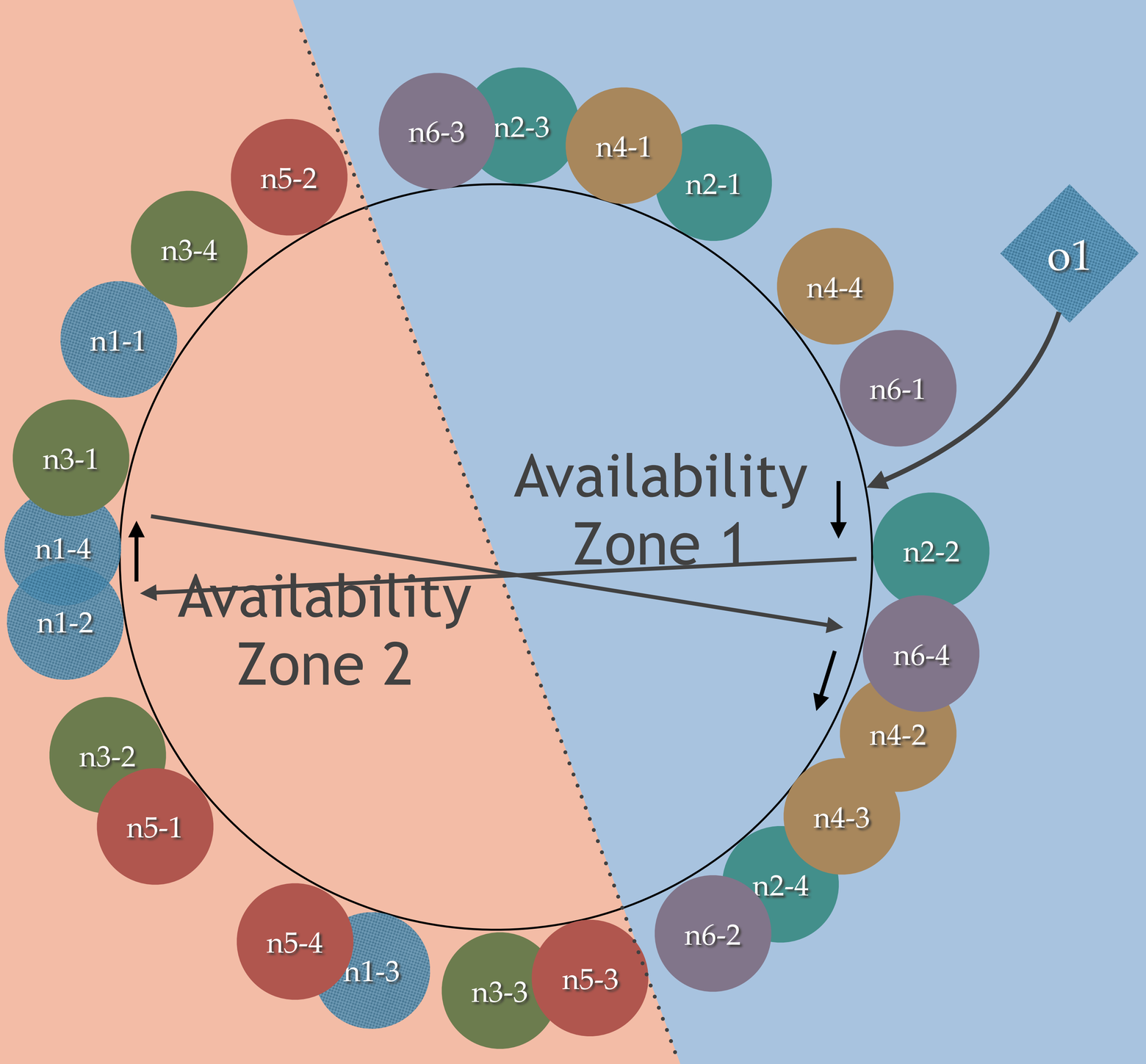

n6-3
n2-3
n5-2
n2-1
n3-2
n4-1
n4-4
n1-1
n5-4
n3-1
n1-3
n1-4
n1-2
n6-4
n3-4
n4-2
n4-3
n5-1
n2-4
n6-1
n6-2
n2-2
n3-3
n5-3

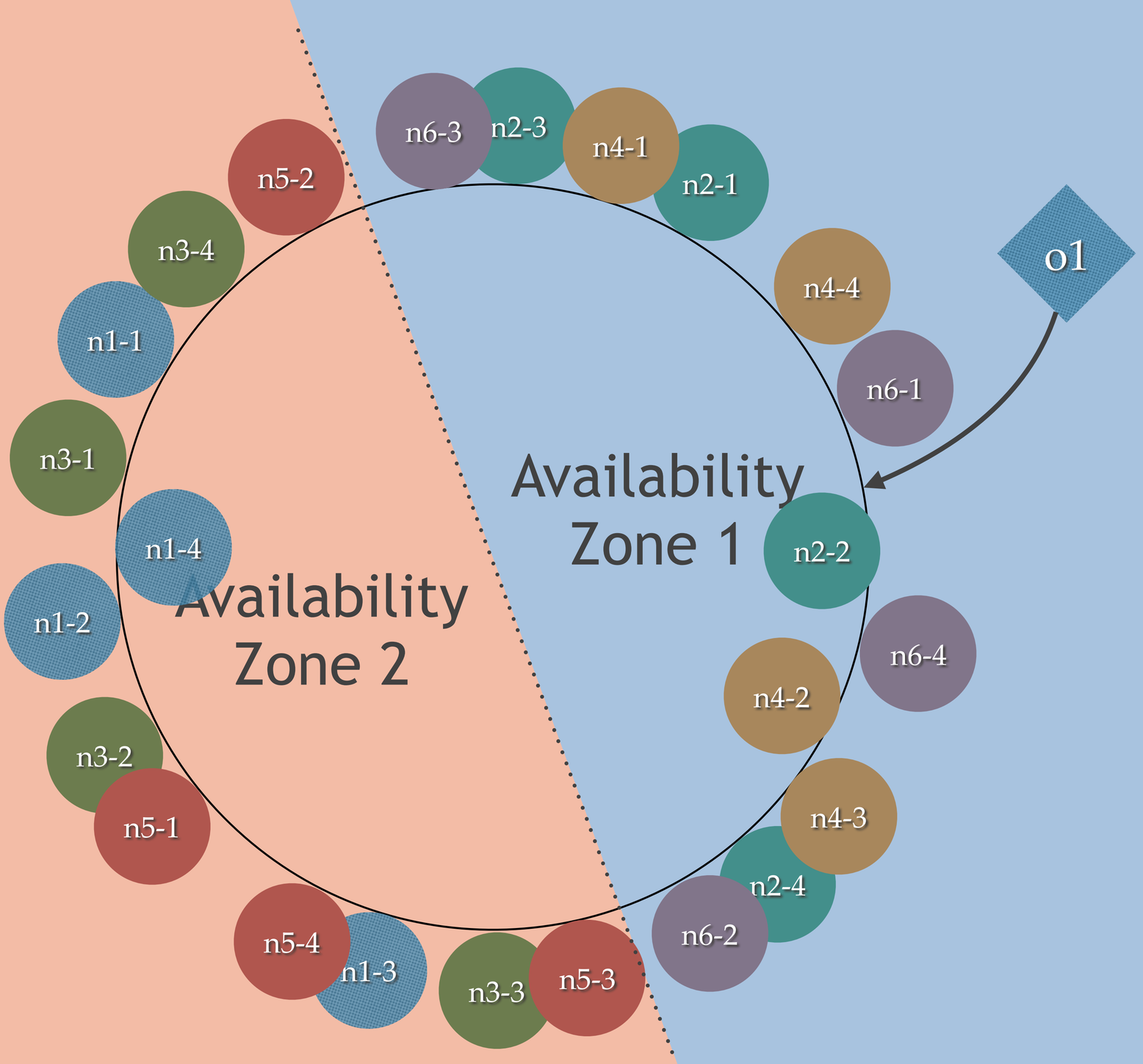

o1
n2-1
n3-2
n4-4
n5-4
n1-3
n6-4
n3-4
n6-2
n4-3
n5-3
n3-3
n2-2
n6-1
n2-4
n5-1
n4-2
n1-2
n1-4
n3-1
n1-1
n4-1
n5-2
n6-3
n2-3


n6-3
n2-3
n5-2
n2-1
n3-2
o1
n4-1
n4-4
n1-1
n5-4
n3-1
n1-3
n1-4
n1-2
n6-4
n3-4
n4-2
n4-3
n6-2
n5-1
n2-4
n6-1
n2-2
n3-3
n5-3


n6-3
n2-3
n5-2
n2-1
n3-2
n4-1
n4-4
n1-1
n5-4
n3-1
n1-3
n1-4
n1-2
n6-4
n3-4
n4-2
n4-3
n5-1
n2-4
n6-1
n6-2
n2-2
n3-3
n5-3


Availability Zone 1
Availability Zone 2
n6-3
n2-3
n4-1
n2-1
n5-2
n3-4
n4-4
n1-1
n6-1
n3-1
n2-2
n1-4
n6-4
n1-2
n4-2
n3-2
n4-3
n5-1
n2-4
n6-2
n5-4
n1-3
n3-3
n5-3


n6-3
n2-3
n4-1
n2-1
n5-2
n3-4
n4-4
o1
n1-1
n6-1
n3-1
Availability
Zone 1
n2-2
n1-4
n6-4
n1-2
Availability
Zone 2
n4-2
n3-2
n4-3
n5-1
n2-4
n6-2
n5-4
n1-3
n3-3
n5-3


n6-3
n2-3
n4-1
n2-1
n5-2
n3-4
o1
n4-4
n1-1
n6-1
n3-1
Availability
Zone 1
n2-2
n1-4
n6-4
n1-2
Availability
Zone 2
n4-2
n3-2
n4-3
n5-1
n2-4
n6-2
n5-4
n1-3
n3-3
n5-3

A look at adaptive algorithms in

Replication

How do you choose the right unit of work for tasks?

What does it sound like when a system

Backfires

Batch it faster than single ops

- less latency impact
- less transactional overhead

What with QoS enforcement & circuit breakers?

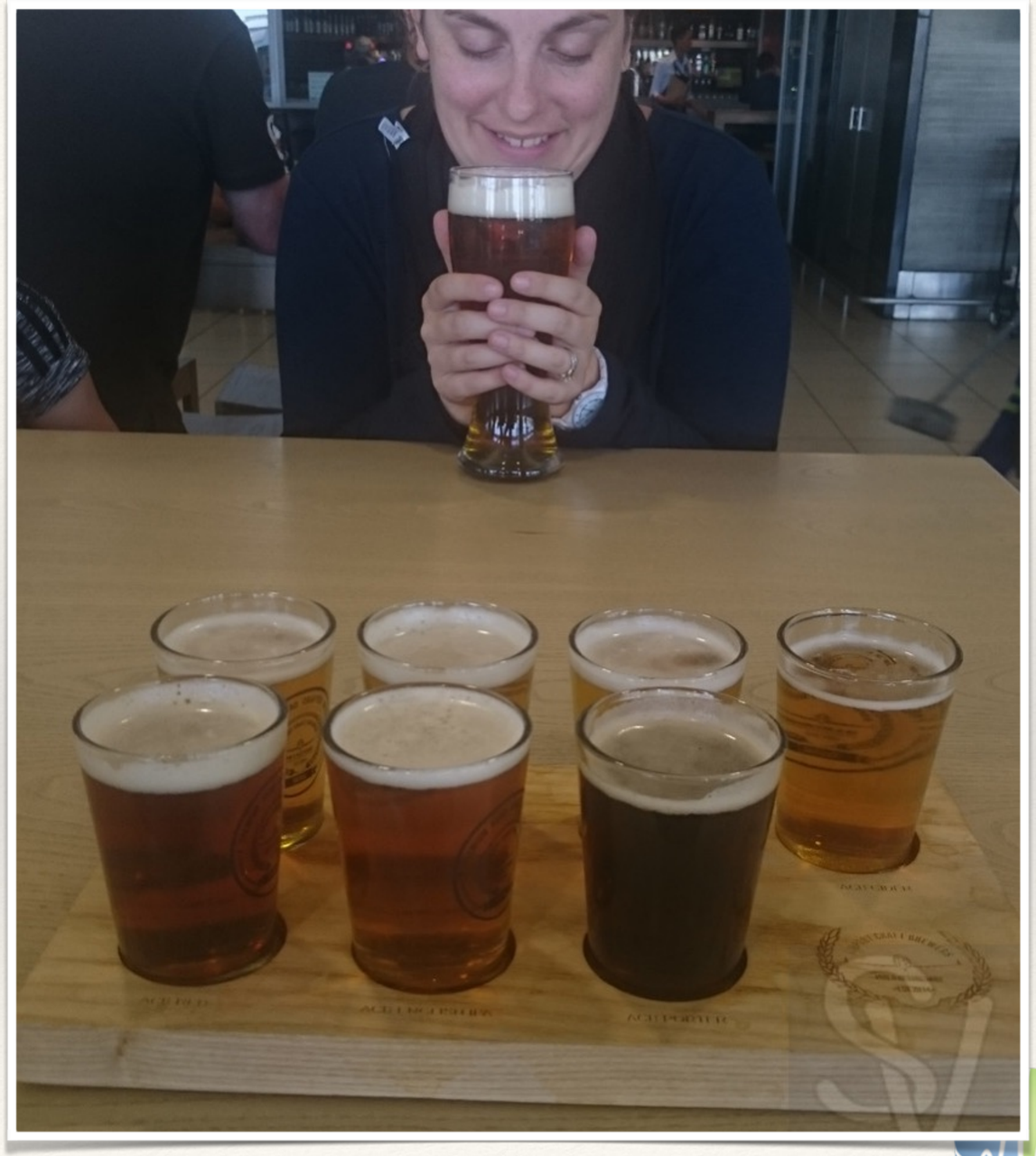
Flogging

TCP (and everything else) can teach us something.

This provides us

Opportunities

What if we had relative homogeny of systems and workloads?



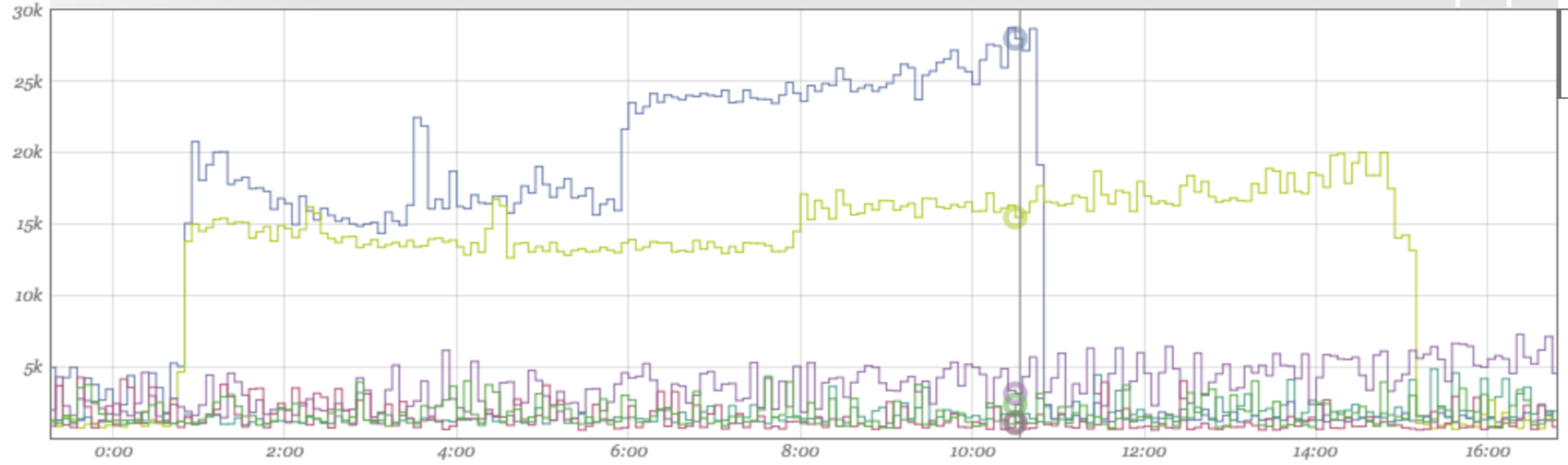
Some problems get easier

Simplified Outlier Detection

If there is an implicit assumption that machines behave similarly,

then it becomes much easier to determine when they fail to do so.

Original Graph



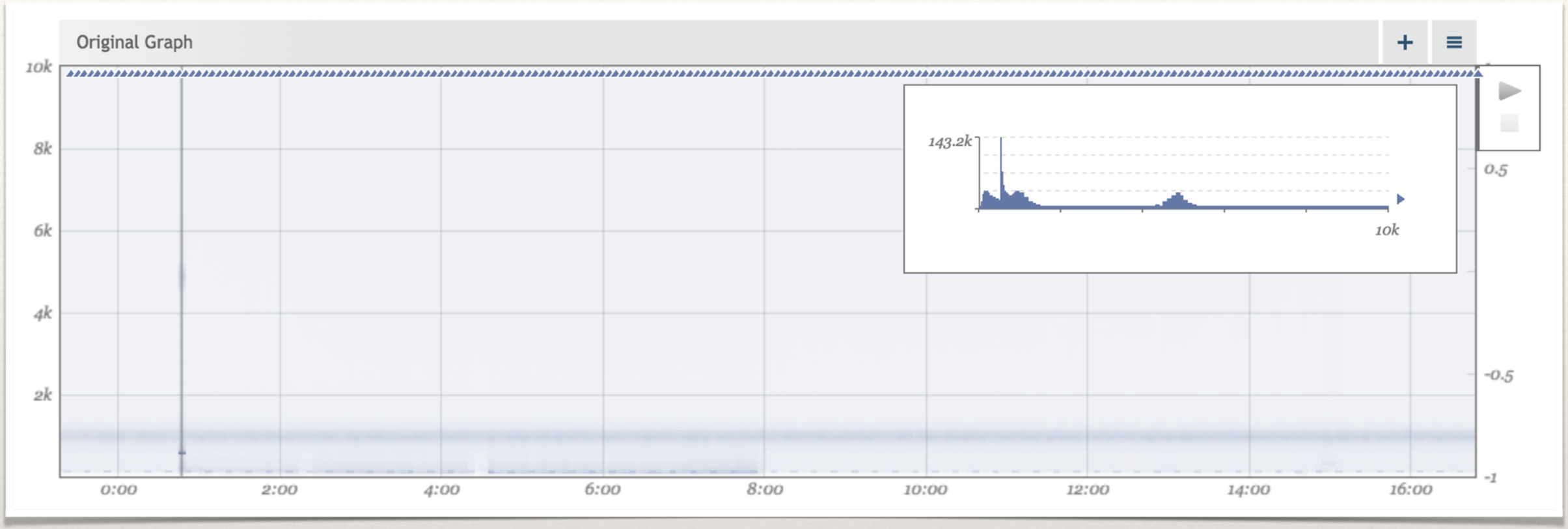
	Jul 02 2014, 10:30 (5M)
snowth5 latency IQR	1.1150449523579k
snowth6 latency IQR	28.010637338557k
snowth7 latency IQR	15.4763937582k
snowth8 latency IQR	3.1138964814012002k
snowth9 latency IQR	1.0534715994518k
snowth10 latency IQR	2.3435219867851003k

New things become possible

Predicting Future Conditions

With higher volume data,

statistical models offer higher confidence.



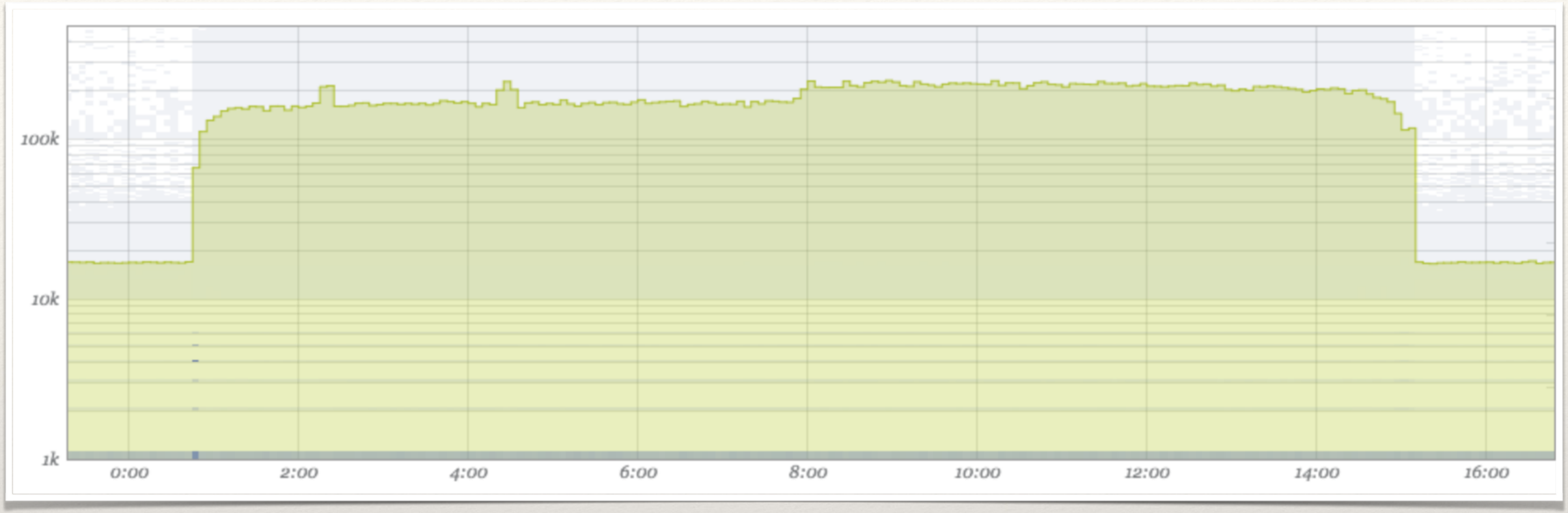
We have better tools now that high-volume data isn't intimidating:

Better insight

That hairline contains >9MM samples.

Histogram shown.

4 modes… WTF?



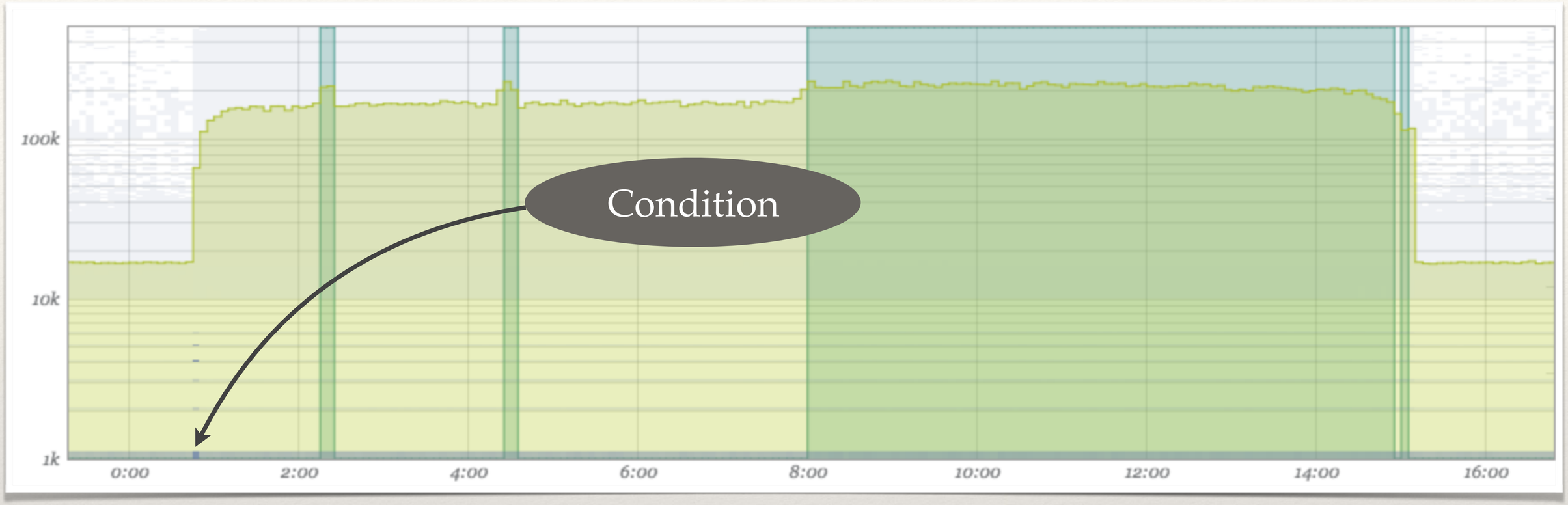
It takes good understanding of statistics to ask the right questions.

Misleading yourself

This is a q(0.99) — 99th percentile.

It obviously goes off the rails around 1am.

No.

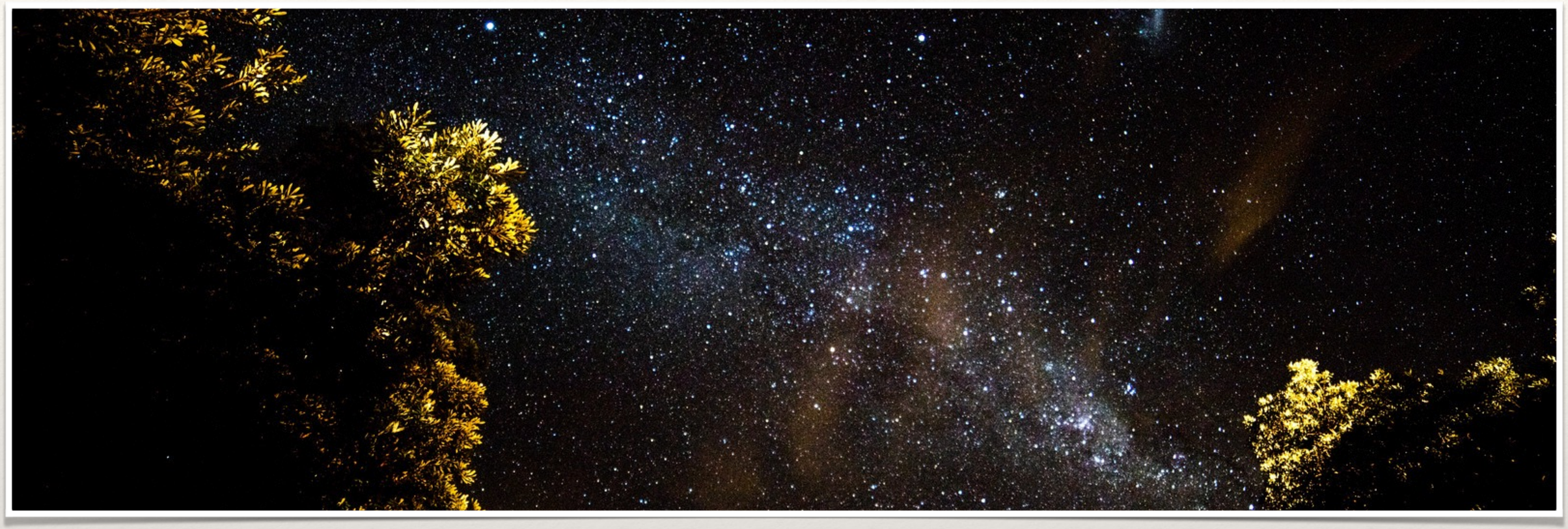


It takes good understanding of statistics to ask the right questions.

Measuring what matters

Instead of measuring
"how slow transaction are"

we measure
"how many transactions are too slow"



We have a new tool in the tool chest:

Intentionally Failing Nodes

When nodes are cattle, not pets…

Expect more from you systems.

Thank You

You can observe better,

know more,

don’t settle.