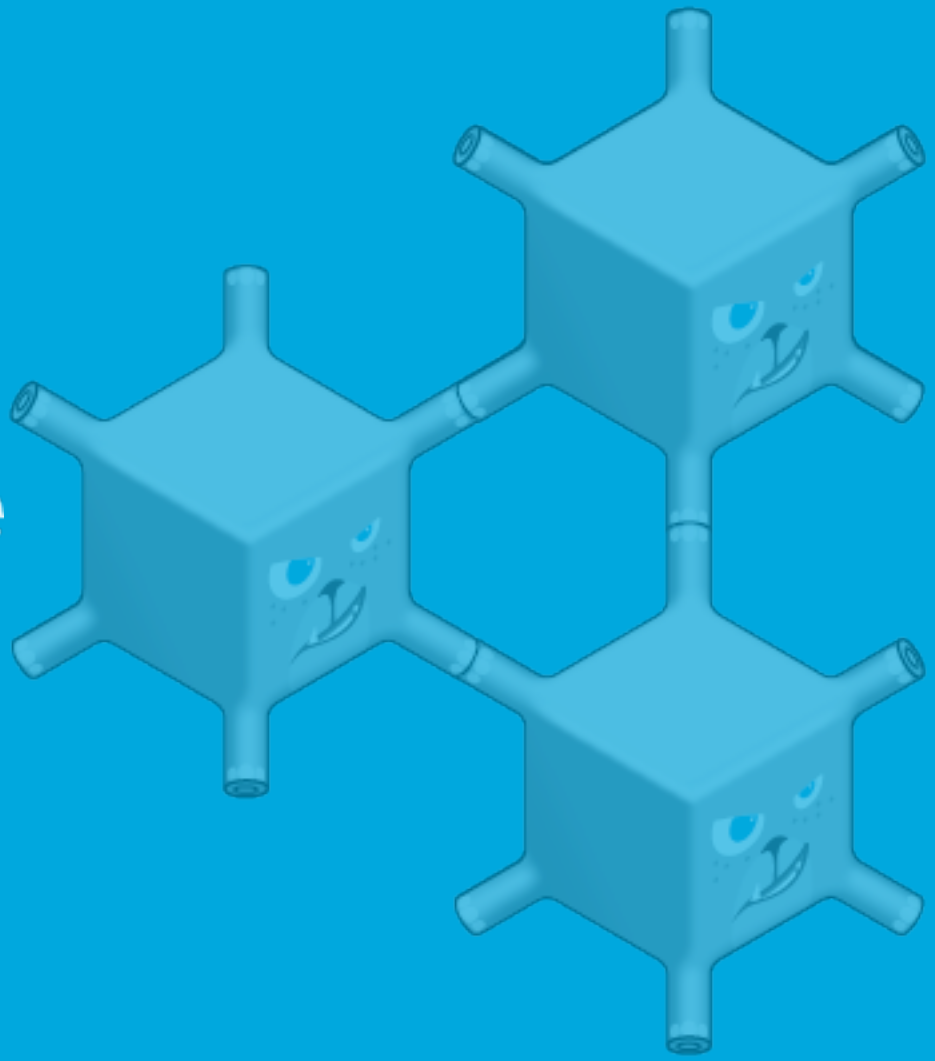
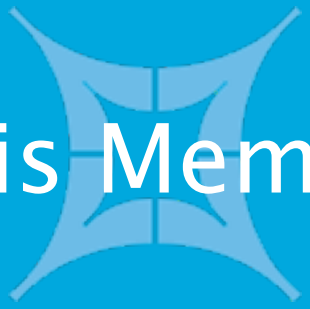
membase

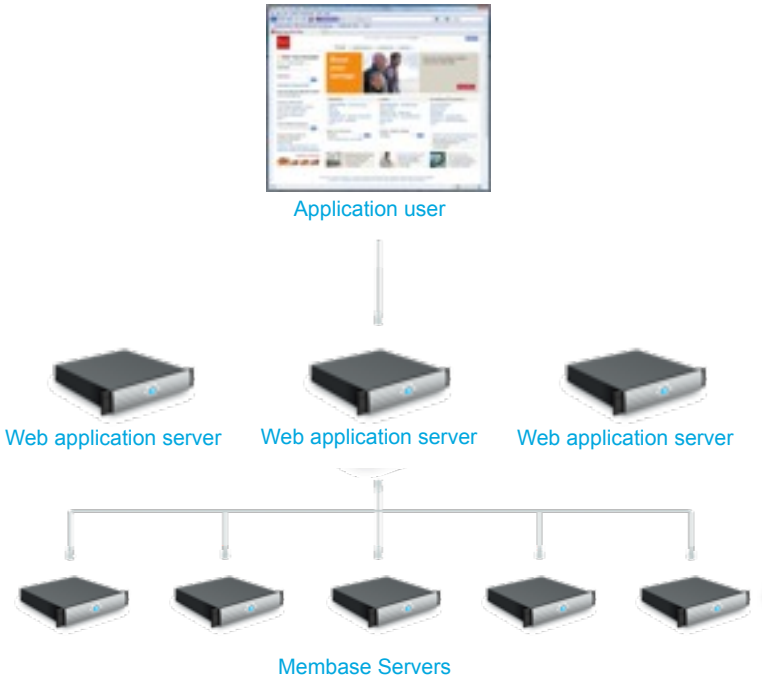


Erlang Factory Lite L.A.

What is Membase?



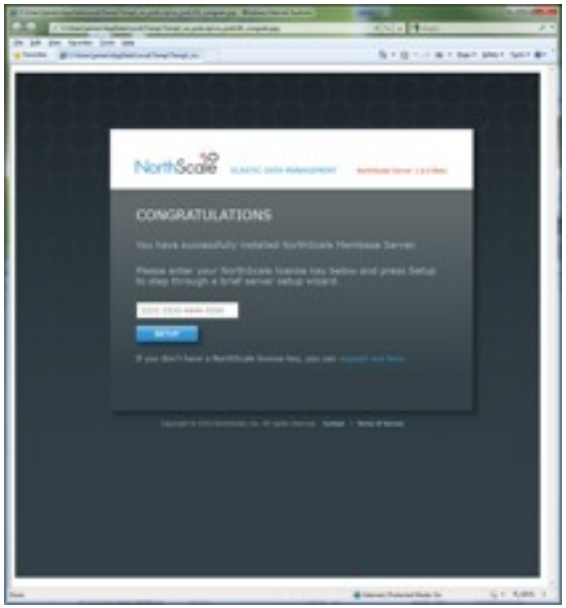
Membase is a distributed database



In the data center

On the administrator console

Membase is Simple, Fast, Elastic



- Five minutes or less to a working cluster
  - Downloads for Linux and Windows
  - Start with a single node
  - One button press joins nodes to a cluster
- Easy to develop against
  - Just SET and GET – no schema required
  - Drop it in. 10,000+ existing applications already “speak membase” (via memcached)
  - Practically every language and application framework is supported, out of the box
- Easy to manage
  - One-click failover and cluster rebalancing
  - Graphical and programmatic interfaces
  - Configurable alerting

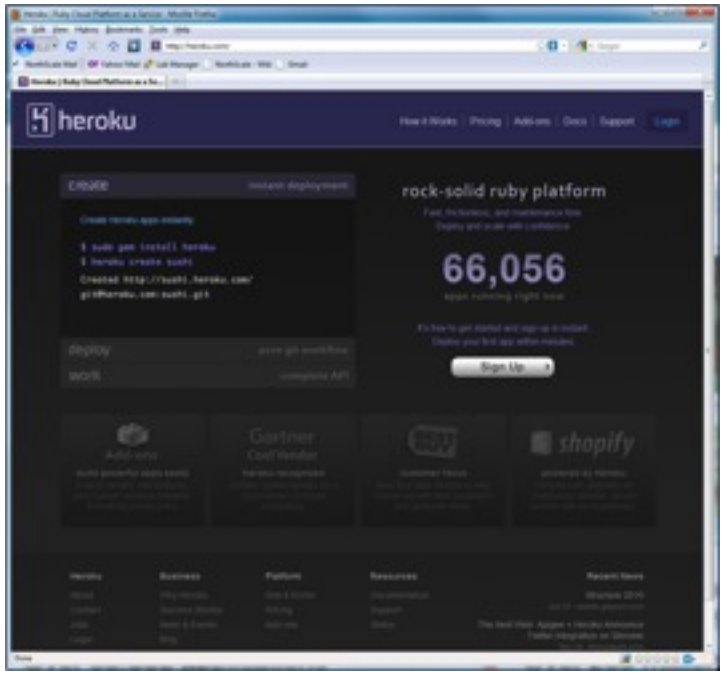
Membase is Simple, Fast, Elastic

- Predictable
  - "Never keep an application waiting"
  - Quasi-deterministic latency and throughput
- Low latency
  - Built-in Memcached technology
- High throughput
  - Multi-threaded
  - Low lock contention
  - Asynchronous wherever possible
  - Automatic write de-duplication

Membase is Simple, Fast, Elastic

- Zero-downtime elasticity
  - Spread I/O and data across commodity servers (or VMs)
  - Consistent performance with linear cost
  - Dynamic rebalancing of a live cluster
- All nodes are created equal
  - No special case nodes
  - Any node can replace any other node, online
  - Clone to grow
- Extensible
  - Filtered TAP interface provides hook points for external systems (e.g. full-text search, backup, warehouse)
  - Data bucket – engine API for specialized container types

Deployments Leading Membase

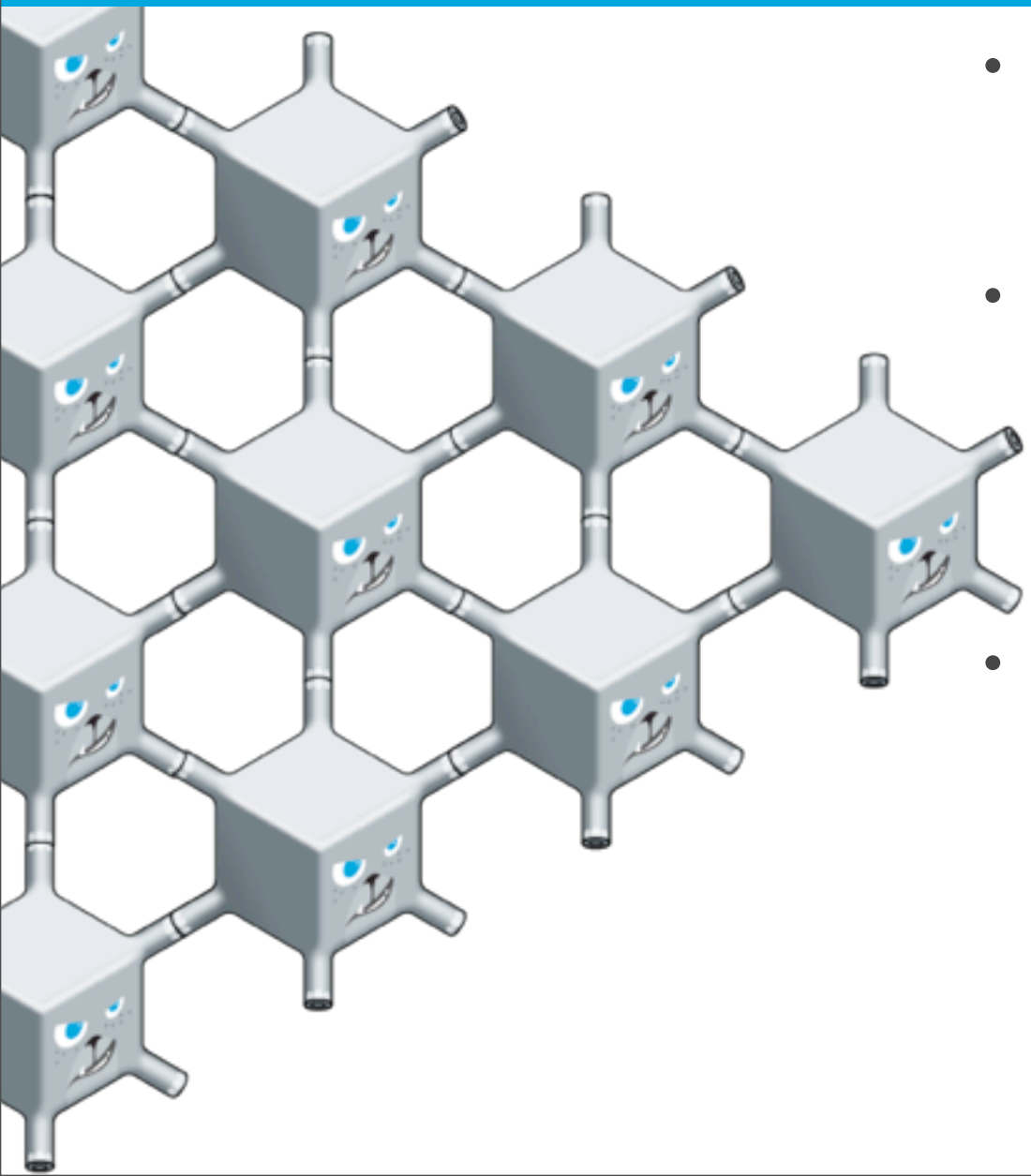


- Leading cloud service (PAAS) provider
- Over 65,000 hosted applications
- **Membase Server** serving over 1,200 Heroku customers (as of June 10, 2010)

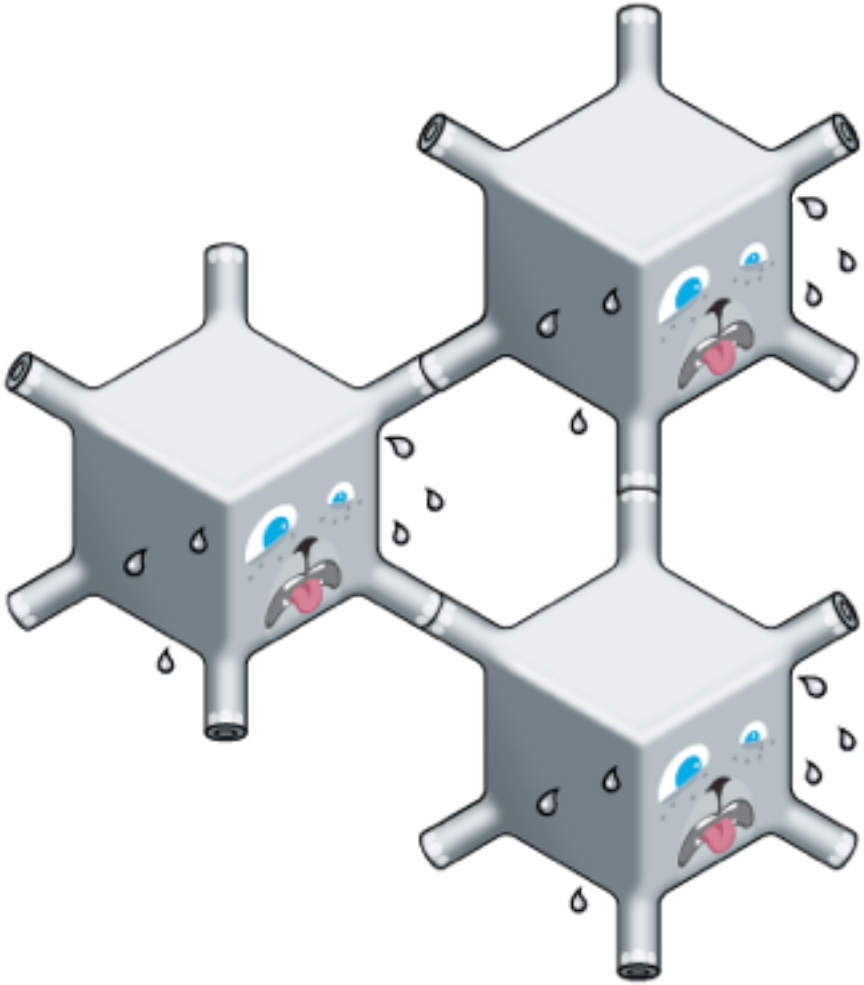
- Social game leader – FarmVille, Mafia Wars, Café World
- Over 230 million monthly users
- **Membase Server** is the 500,000 ops-per-second database behind FarmVille and Café World

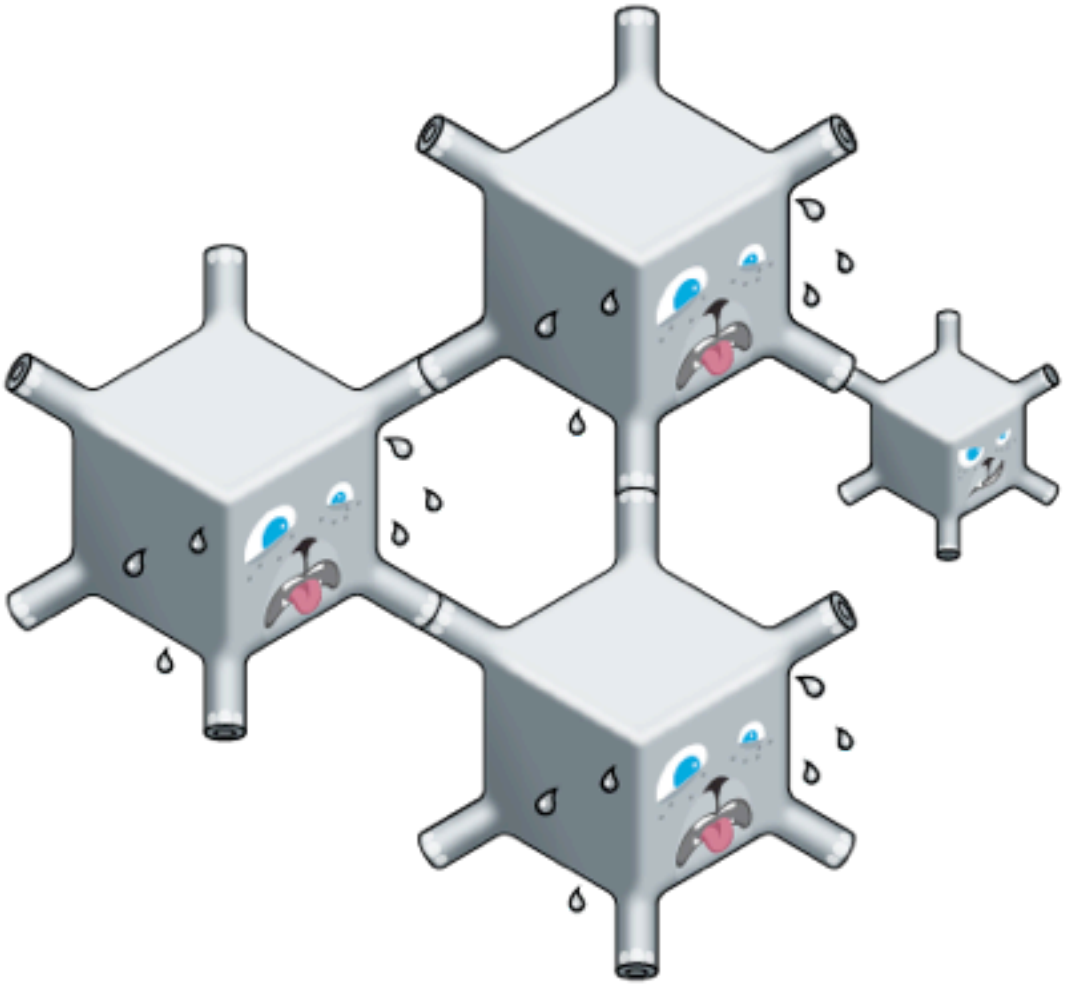
Membase Architecture

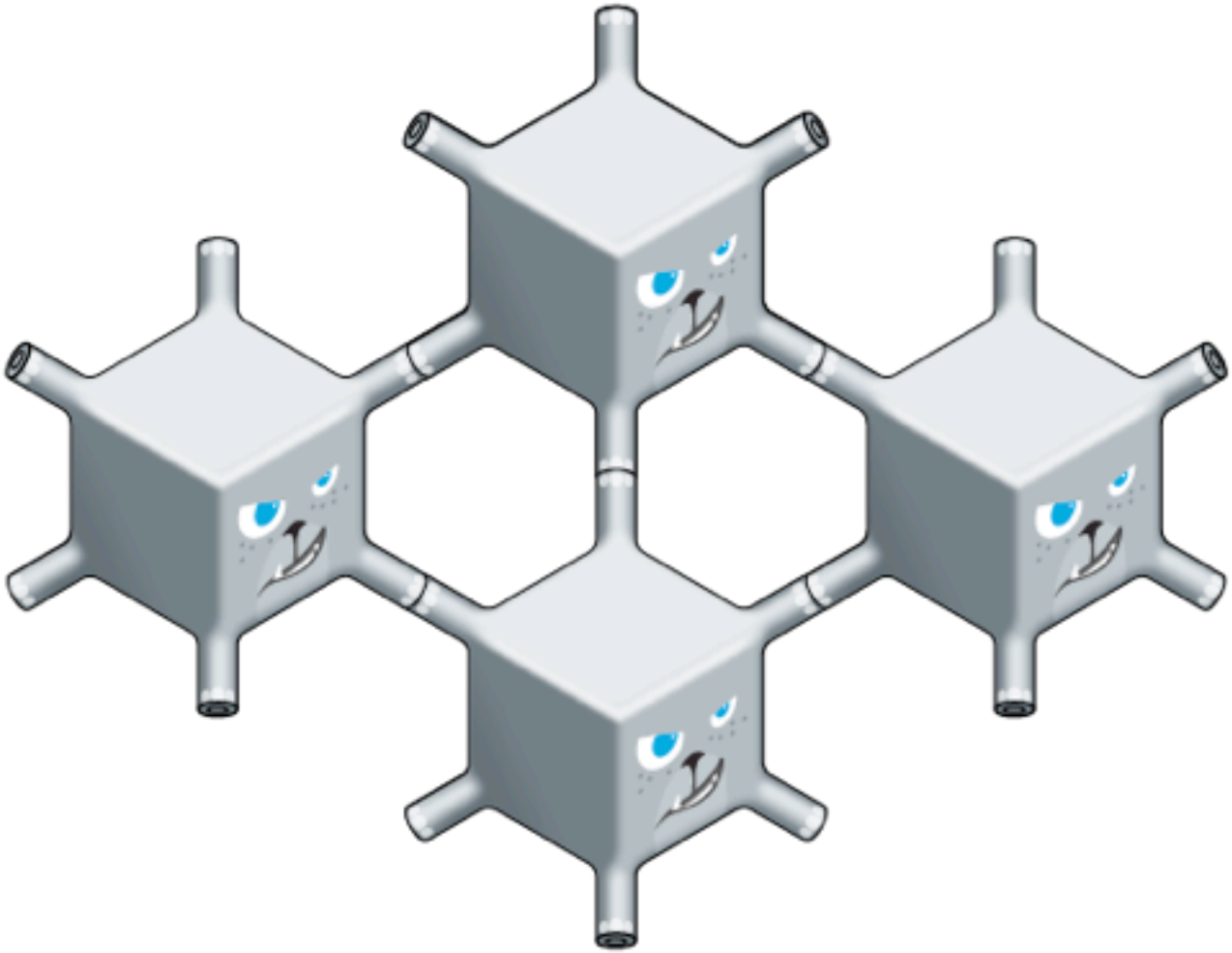
Clustering



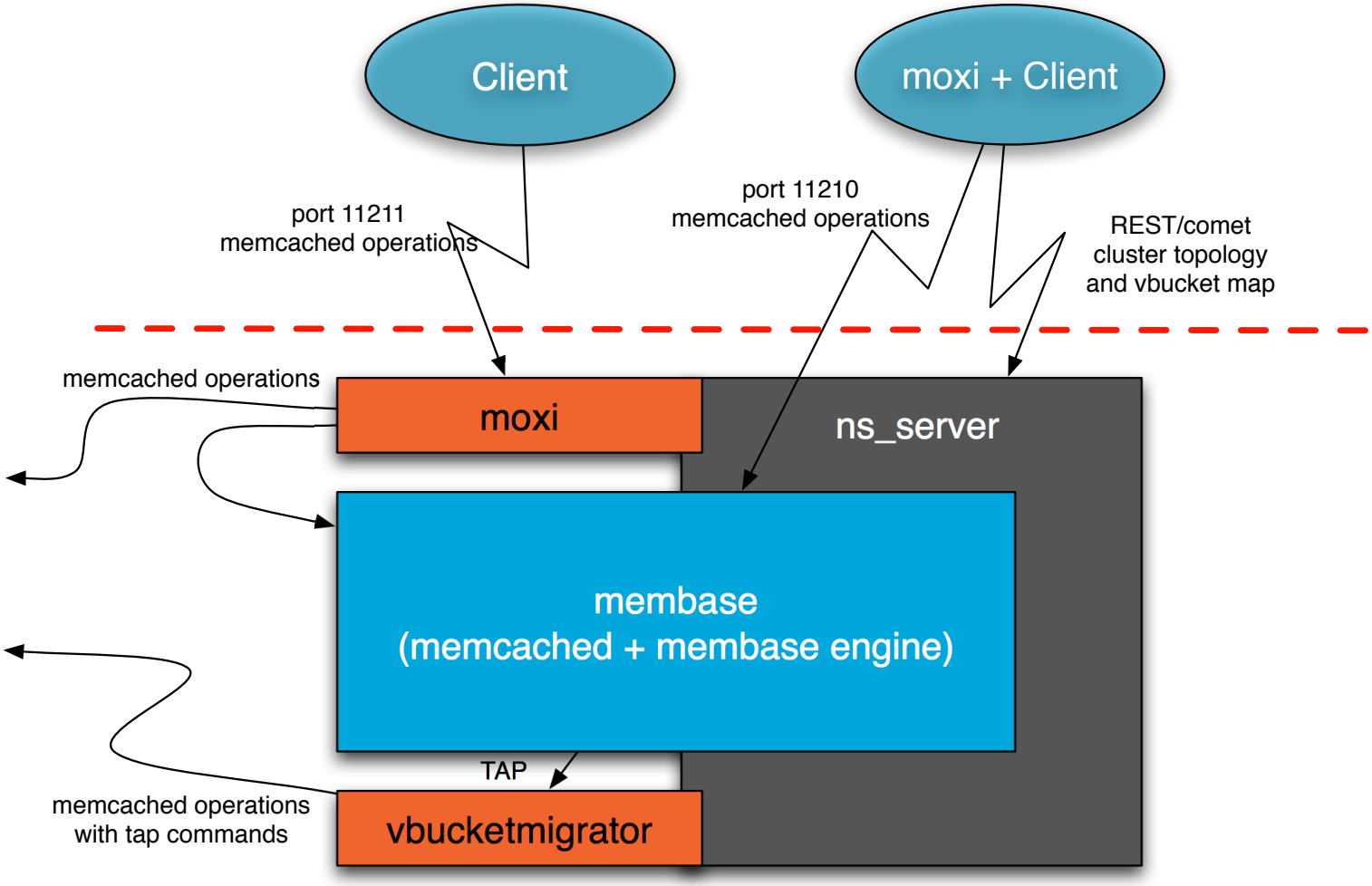
- Underlying cluster functionality based on erlang OTP
- Have a custom, vector clock based way of storing and propagating...
  - Cluster topology
  - vBucket mapping
- Collect statistics from many nodes of the cluster
  - Identify hot keys, resource utilization





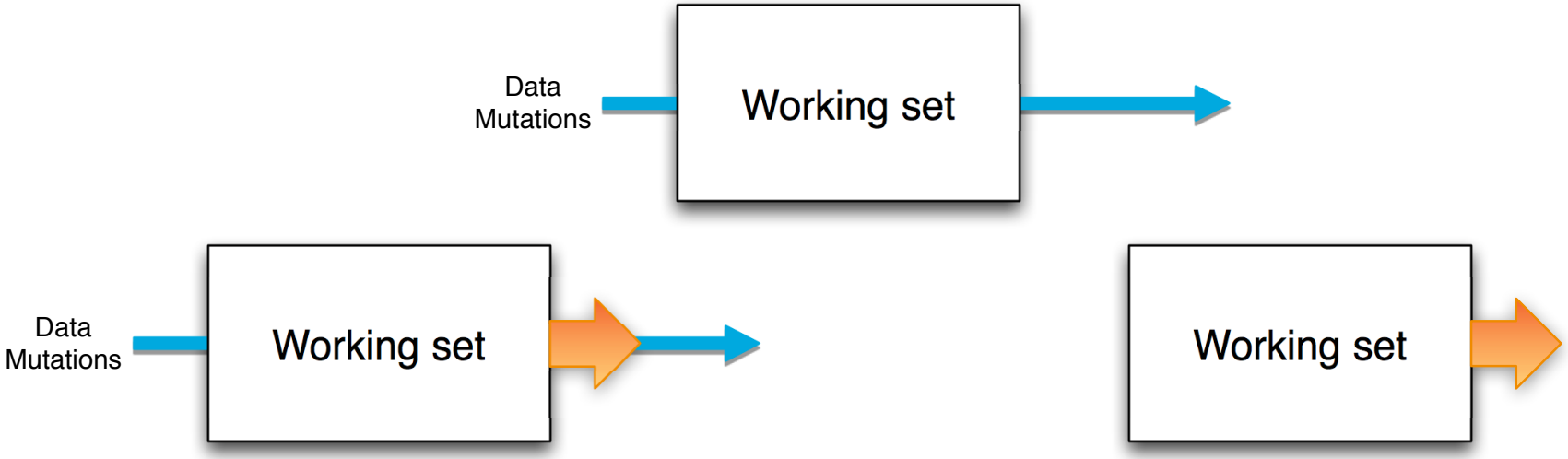


Clients, nodes and other nodes

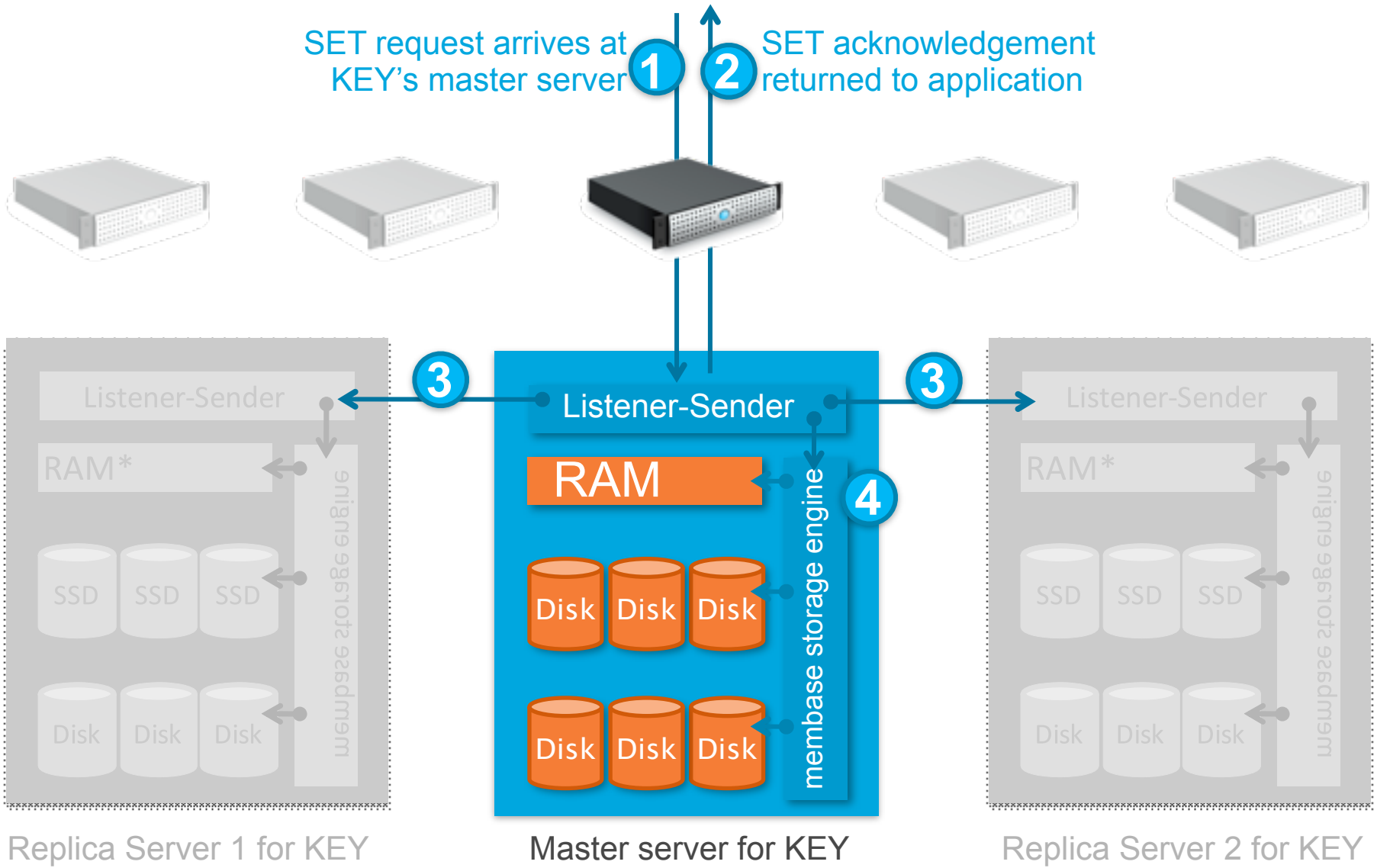


TAP

- A generic, scalable method of streaming mutations from a given server
  - As data operations arrive, they can be sent to arbitrary TAP receivers
- Leverages the existing memcached engine interface, and the non-blocking IO interfaces to send data
- Three modes of operation



Membase data flow – under the hood



Data buckets are secure membase “slices”

Application user

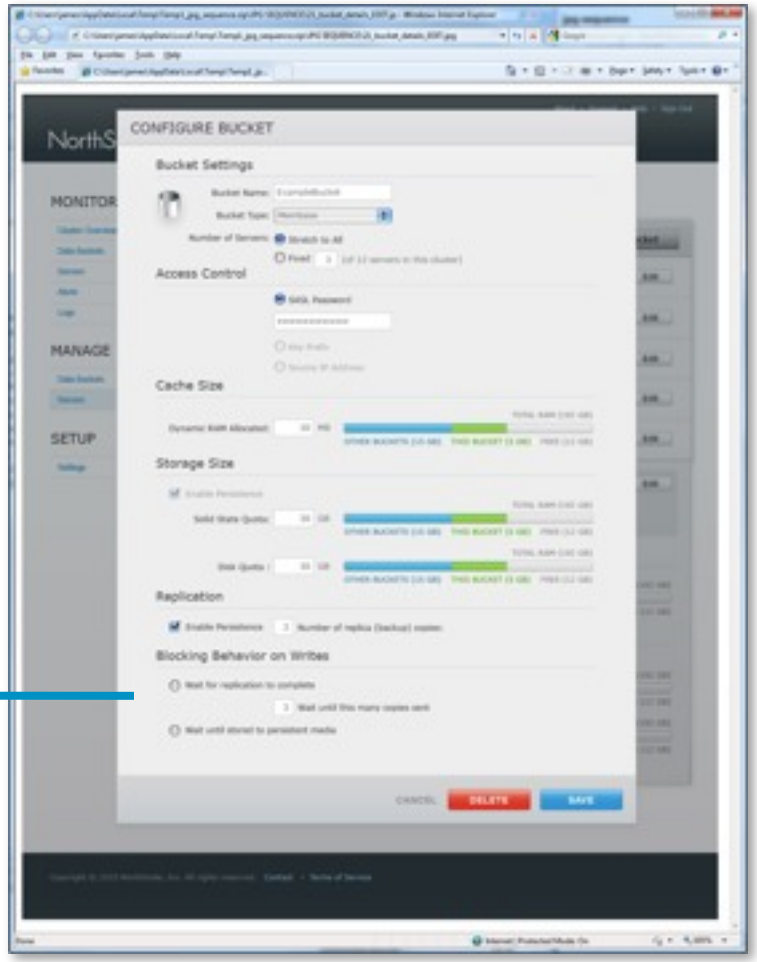
Web application server

Bucket 1

Bucket 2

Aggregate Cluster Memory and Disk Capacity

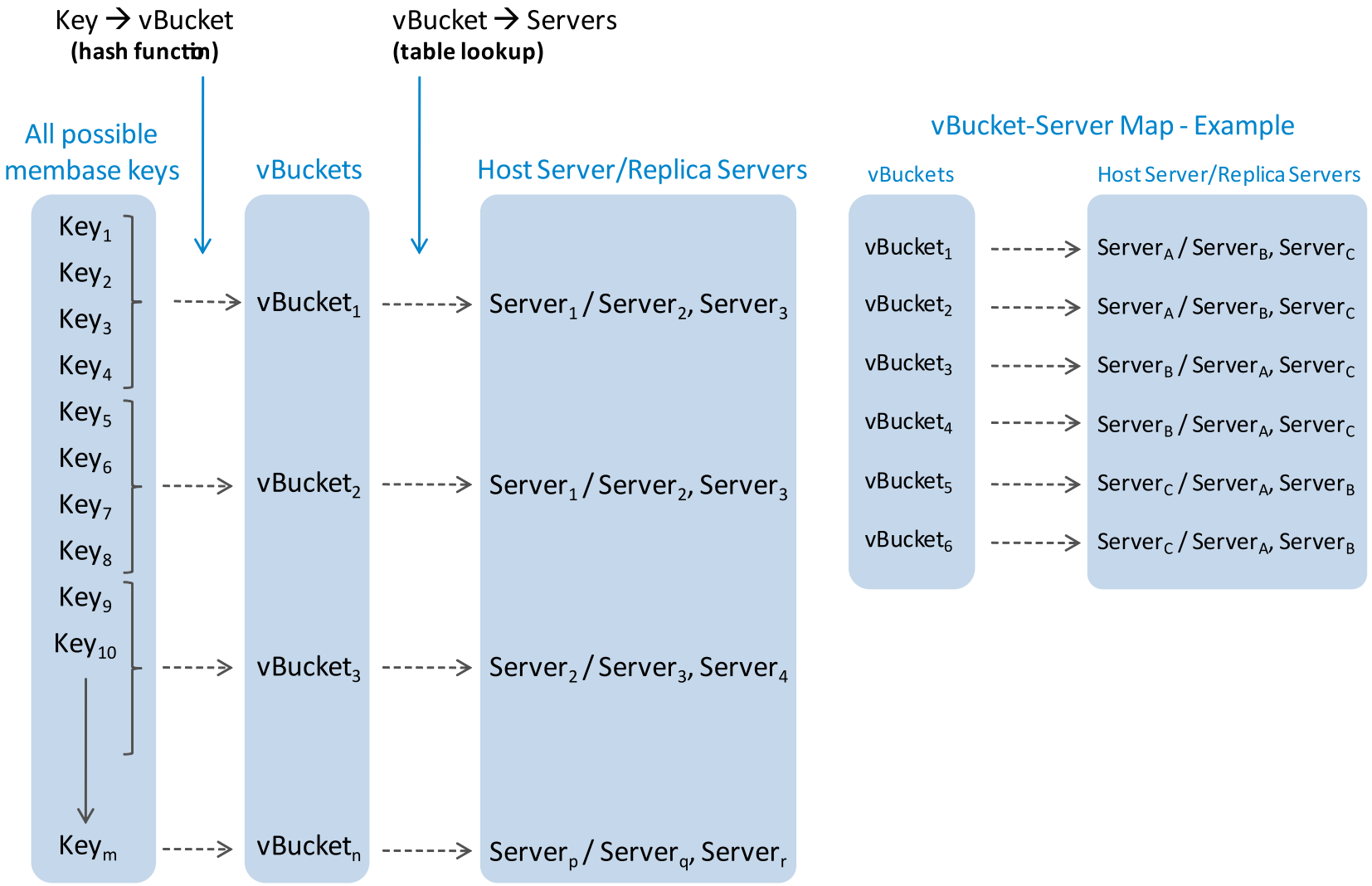
Membase data servers



In the data center

On the administrator console

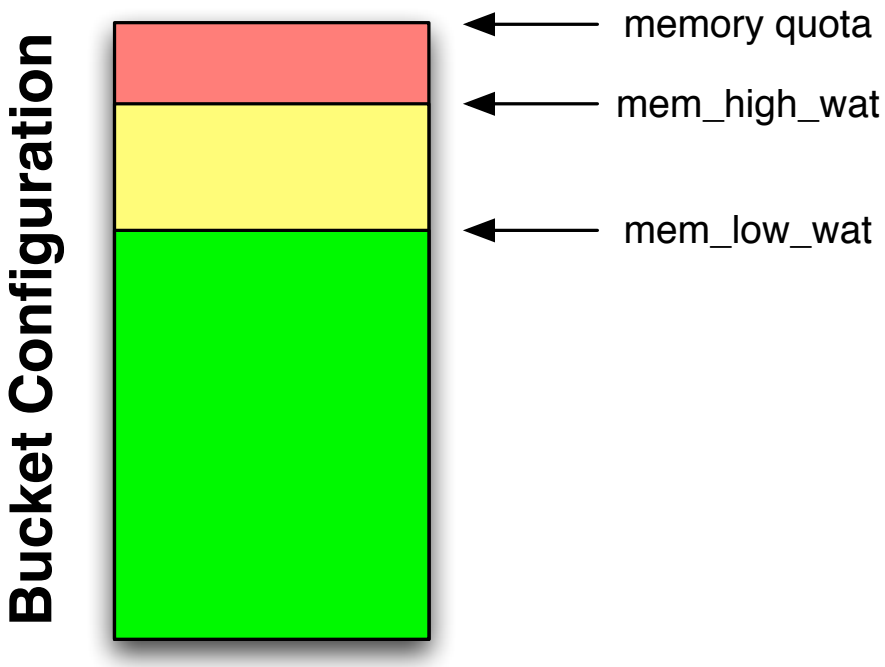
vBucket mapping



Disk > Memory

Dataset may have many items infrequently accessed. However, memcached has different behavior (LRU) than wanted with membase.

Still, traditional (most) RDBMS implementations are not 100% correct for us either. The speed of a miss is very, very important.



Erlang Experiences

Membase Erlang “Control Plane”

- Built atop distributed Erlang
  - Using os_mon for gathering cluster information
  - Using Mnesia to store historic statistics
- Our own Supervisors and hierarchy
  - Minorly modified C processes
  - Monitor OS processes as Erlang processes
  - Supervisor cushion
    - Slow down fast startup failures while keeping normal exit/crash fast
- Custom ‘heartbeat’
  - Determine failure and gather system resource basics
- Mochiweb for REST interface
  - Represent all cluster state and management

Erlang and Membase Tricks

- IP/Interface problems
  - Respond where asked
- Integrated erlwsh, behind HTTP auth
- Update state from anywhere
  - Vector clocks for config
- The “global singleton”
  - Blame Matt for name
  - Some processes in one place

Erlang and Membase: Lessons Learned

- Networks are more fluid
  - Developer laptops
  - Cloud compute environments
- Anyone need some I/O?
  - Look for the +A
    - "+A size: Sets the number of threads in async thread pool, valid range is 0-1024. Default is 0."
- os_mon
  - Virtual is still virtual
  - Disk info not quite what we needed
- List processing overuse
- SASL Logs for non-Erlang initiated

Membase Demo

Q&A


membase